Xiao *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:4
DOI 10.1186/s13634-015-0300-4

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation

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

Abstract

This paper investigates deep neural networks (DNN) based on nonlinear feature mapping and statistical linear feature adaptation approaches for reducing reverberation in speech signals. In the nonlinear feature mapping approach, DNN is trained from parallel clean/distorted speech corpus to map reverberant and noisy speech coefficients (such as log magnitude spectrum) to the underlying clean speech coefficients. The constraint imposed by dynamic features (i.e., the time derivatives of the speech coefficients) are used to enhance the smoothness of predicted coefficient trajectories in two ways. One is to obtain the enhanced speech coefficients with a least square estimation from the coefficients and dynamic features predicted by DNN. The other is to incorporate the constraint of dynamic features directly into the DNN training process using a sequential cost function.

In the linear feature adaptation approach, a sparse linear transform, called cross transform, is used to transform multiple frames of speech coefficients to a new feature space. The transform is estimated to maximize the likelihood of the transformed coefficients given a model of clean speech coefficients. Unlike the DNN approach, no parallel corpus is used and no assumption on distortion types is made.

The two approaches are evaluated on the REVERB Challenge 2014 tasks. Both speech enhancement and automatic speech recognition (ASR) results show that the DNN-based mappings significantly reduce the reverberation in speech and improve both speech quality and ASR performance. For the speech enhancement task, the proposed dynamic feature constraint help to improve cepstral distance, frequency-weighted segmental signal-to-noise ratio (SNR), and log likelihood ratio metrics while moderately degrades the speech-to-reverberation modulation energy ratio. In addition, the cross transform feature adaptation improves the ASR performance significantly for clean-condition trained acoustic models.

Keywords: Beamforming, Deep neural networks, Dynamic features, Feature adaptation, Robust speech recognition, Reverberation challenge, Speech enhancement

*Correspondence: xiaoxiong@ntu.edu.sg
[1]Temasek Lab@NTU, Nanyang Technological University, 50 Nanyang Drive, 637553 Singapore, Singapore
Full list of author information is available at the end of the article

1 Introduction

Automatic speech recognition (ASR) systems and hands-free speech acquisition systems have achieved satisfactory performance for close-talk microphones. However, the performance of these systems is still poor for far-talk microphones where the distance between the speaker and the microphone is large. This is because the speech signals recorded by the far-talk microphones are significantly corrupted by background noise and room reverberation. Hence, improving the robustness of the ASR and other speech-processing systems for far-talk speech is an important task for the deployment of such systems in more realistic environments.

It is well known that reverberation is produced by multipath propagation of an acoustic signal from its source to the microphone in enclosed spaces. The distortion of reverberation to an acoustic signal can be modeled by the acoustic impulse response (AIR), which may last for hundreds of milliseconds. Reverberation cancelation has been tried by many researchers using deconvolution techniques that estimate and apply the inverse of the AIR to the reverberant microphone outputs. Several significant contributions have been made in these areas [1–7]. However, the available techniques are sensitive to estimation error of the AIR, which is difficult to estimate in realistic environments. In addition, as the talker may change location in a room, the geometry between the talker and the microphone receiver will change accordingly, and consequently, the acoustic impulse response is time-varying. The additive background noise further increases the difficulty for accurately estimating the AIR.

Besides the studies on reverberation cancellation, other studies focus on reverberation suppression. Both spatial processing using multiple microphones and spectral enhancement belong to the reverberation suppression methods. Microphone array processing techniques such as beamforming provide spatial filtering to suppress specular reflections so that the speech signal from the desired direction of arrival (DOA) can be enhanced. The most direct and straightforward technique is the delay and sum (DS) beamfomer [8]. Several variants of the DS beamformer with adaptations also exist [8–12]. Generally, adaptive beamformers have been found to be efficient in applications to suppress localized additive noise sources. The DS beamformer can partially reduce reverberation coming from directions other than the look-direction. As the reverberant speech signal consists of highly dependent distortions that are delayed versions of the signal itself, the adaptive beamformers may attenuate the direct-path speech signal while reducing reverberation. Very recently, a conjunction of DS beamformer and minimum variance distortionless response (MVDR) beamformer has shown favorable performance for speech enhancement in the room environment [13].

To further reduce the reverberation from the look-direction, spectral-subtraction techniques may be applied. Lebart et al. has shown the effectiveness of spectral subtraction for dereverberation [14, 15]. The authors assume a statistical model of the room impulse response comprising Gaussian noise modulated by a decaying exponential function. The power spectral density of the later impulse response is identified and removed by spectral subtraction.

Although the above-mentioned approaches to dereverberation can be effective for both speech enhancement and ASR [16], blind identification of other features is often required. The reverberation conditions have high impact on the identification performance, thus affecting the dereverberation performance.

In this work, we investigated a different approach for speech dereverberation that is based on learning from data. If a parallel corpus of clean speech and the corresponding reverberant speech is available, it is possible to learn a mapping that maps reverberant speech coefficients to clean speech coefficients. Supposing that the parallel corpus is representative of the real test environment, and the mapping function is accurate, the learning-based approach is a viable way to deal with reverberation. In fact, the concept of the learning-based approach is not new. For example, in noise-robust ASR, a popular feature compensation method is called SPLICE [17, 18], which learns a set of linear transforms to map noisy feature vectors to clean feature vectors. In voice conversion [19], it is a common practice to learn a set of linear transforms to map a source speaker's speech coefficients, such as the log-magnitude spectrum, to a target speaker's speech coefficients. A major limitation of these methods is that linear transforms are too limited to accurately predict clean speech coefficients, especially when the relationship between the clean and distorted speech coefficients are highly nonlinear.

Recently, there are studies on using neural networks (NN) to learn the nonlinear mapping between clean and distorted speech coefficients. Neural networks 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 [20]. A NN with more than one hidden layer is usually called a deep NN, or DNN. Recently, DNN has become popular after a pretraining step, called restrictive Boltzmann machine (RBM) pretraining [21, 22], was introduced to initialize the network parameters to some reasonable values such that backpropagation can then be used to train the network efficiently on task-dependent objective functions. The advantage of a DNN over one-hidden-layer NN is that the deep structure of the DNN allows much more efficient representation of many nonlinear transformations/functions [22]. In the past several years, DNN and

other neural networks have been applied to many speech processing tasks. In [23], DNN was used for acoustic modelling in ASR systems and now it has become the de-facto standard acoustic model. In [24], another DNN architecture, called the deep recurrent neural network (DRNN), is used to estimate clean speech features (MFCC) from noisy features. In [25], a special case of recurrent neural networks, called long short-term memory (LSTM), is used to map reverberant features to clean features. In [26], a DNN is used to predict the speech mask, which is then used for enhancing speech for robust ASR. It is also found that adapting the mask-estimating DNN using a linear input transform further improves the ASR performance. While the previous two studies focus on predicting low-dimensional feature vector for ASR, in [27], deep neural networks (DNN) are used to directly estimate the high-dimension log-magnitude spectrum for speech denoising. This method was later applied as a preprocessor for a robust ASR task [28]. In this study, we use DNN to estimate both the low-dimensional speech feature vectors for the ASR task and a high-dimensional log-magnitude spectrum vector for the speech enhancement task. Although in [25], the authors show that LSTM outperforms DNN significantly for the ASR task, the DNN used in that study is very limited in terms of network size.

In this study, we extend our previous work on DNN-based speech dereverberation [29]. DNN is trained from parallel data of clean and distorted speech coefficients to map distorted speech to clean speech. To improve the performance of DNN mapping for speech enhancement which requires a smooth and natural coefficient trajectory, we proposed two methods that make use of the information of dynamic features. The motivation is that the predicted speech coefficient trajectory should have dynamics (represented by dynamic features) that are similar to real clean speech coefficient trajectories. Specifically, we propose least-square (LS) estimation of log-magnitude spectrum from the static, delta, and acceleration log-magnitude spectrum predicted by DNN. We also propose a new sequential cost function for DNN training that takes into account the mean-squared error of the dynamic spectrum.

Although DNN and other neural-network-based speech coefficients mapping approach have the potential to produce an accurate clean speech estimate, they rely on a *representative parallel* speech corpus for training the neural networks. To address this limitation, we also propose a feature adaptation method that only requires clean speech data during training. At the test phase, a linear transform is estimated to map the reverberant features such that the mapped features fit the clean speech model better. This is in spirit similar to the popular feature space maximum likelihood linear regression (fMLLR) [30] that uses the maximum likelihood criterion to estimate the linear transform. Our proposed method, called cross transform [31], can be seen as an extension of fMLLR in that the cross transform exploits both temporal information (between frames) and spectral information (within frame) in feature mapping, while fMLLR only uses spectral information and short-term temporal information (up to 0.1 s) through delta and acceleration features.

The rest of the paper is organized as follows. Section 2 describes the system designs of the speech enhancement system and the ASR system. Section 3 reviews briefly the beamforming methods used in our system to process multi-channel speech and the conventional spectral subtraction-based dereverberation method. Sections 4 and 5 introduce the DNN-based speech coefficients mapping approach and the cross transform feature adaptation methods in detail, respectively. In Section 6, experimental results and analysis are presented. Conclusions and future research directions are discussed in Section 7.

2 System modules

Our study is focused on reducing reverberation and noise in speech signals for speech enhancement and speech features for ASR. There are three stages of processing, i.e., (1) the beamforming, (2) log-magnitude spectrum enhancement for the speech enhancement task, and (3) speech features (such as MFCC) enhancement for the ASR task. The three processing stages and their combinations are illustrated in Fig. 1.

For beamforming, we investigate two popular methods, i.e., DS beamforming and MVDR beamforming. For spectrogram enhancement, we use the DNN-based speech coefficient mapping method with dynamic feature constraint. We also study the spectral subtraction method for late reverberation reduction. For speech feature enhancement, we also use the proposed cross transform feature adaptation. When multiple channels are available, a speech enhancement system is the cascade of Stages 1 and 2, while a feature enhancement system is the cascade of Stages 1 and 3. For single-channel processing, Stage 1 will be skipped. In the next three sections, we will describe the three stages in more detail.

3 Classic approaches

3.1 Problem formulation

Considering a multichannel speech acquisition system in an enclosed environment, the mth microphone receives the reverberant speech signal $x_m(n)$ with the room impulse response $h_m(n)$ as follows

$$x_m(n) = h_m(n) \otimes s(n) + v_m(n), m = 1, 2, \ldots, M \quad (1)$$

where $s(n)$ is the source signal and $v_m(n)$ is the additive observation noise; the symbol $\otimes$ denotes the linear convolution; M is the number of microphones in the system. The observed signal, $x_m(n)$, at microphone m can

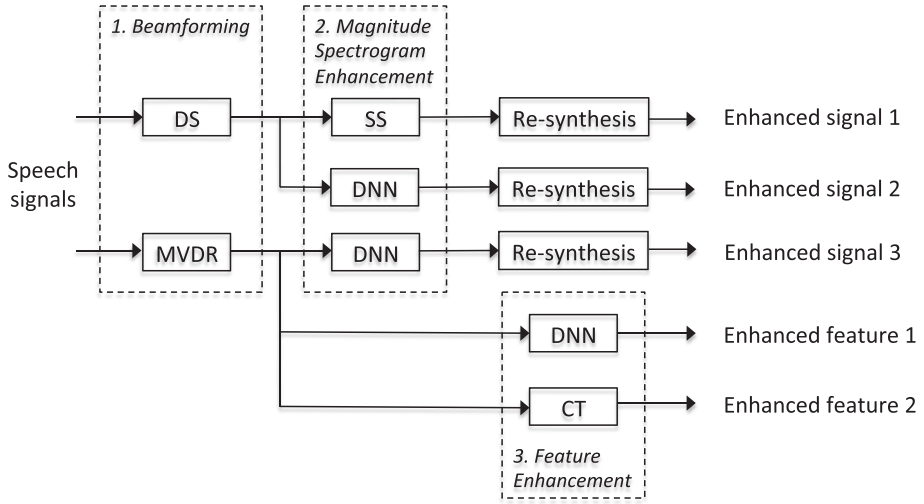

Fig. 1 Illustration of three processing stages: beamforming log-magnitude spectrum enhancement, feature enhancement. DS and MVDR are the delay and sum and minimum variance distortionless response beamforming, respectively. SS refers to spectral subtraction, while DNN stands for deep neural network-based speech coefficient mapping. CT is the cross transform feature adaptation. Three enhanced signals are produced for the speech enhancement task, and two enhanced features are produced for the ASR task

be described as the superposition of the direct-path signal and a finite set of reflections of the direct-path signal. The early part of $h_m(n)$ is referred to as early reverberation which causes spectral changes and leads to a perceptual effect referred to as coloration [32], while the late part (which is referred to as later reverberation) changes the waveform's temporal envelope as exponentially decaying tails are added at sound offsets.

The aim of speech dereverberation is to find a system, with input $x_m(n)$ and output $\hat{s}(n)$, which is an estimate of $s(n)$. The criteria in the estimation, $s(n)$, is application-dependent. It may be related to perceptual quality or ASR performance. The room impulse response $h_m(n)$ could be either time-varying or short-term fixed, which is also application-dependent.

3.2 The DS beamforming

The DS beamformer has a very simple form, and it was developed based on the fact that times of arrival (TOA) of the incoming signal $s(n)$ at different microphones are different. Adding all of the microphone signals $x_m(n)$ together with appropriate amounts of time delay reinforces the incoming signal at the steered direction with respect to noise and reverberant signals arriving from different directions. To obtain more reliable estimation of the time difference of arrivals (TDOAs), cross-correlation methods such as the GCC-PHAT are usually applied [33]. The DS beamformer has a signal-independent formulation, and the output of the DS beamformer improves the signal-to-noise ratio (SNR) [8]. The advantage of the DS beamformer is that it is simple and computationally efficient. There is usually very low distortion in the output of the DS beamformer.

3.3 The MVDR beamforming

The MVDR beamformer has a slightly more complicated form than the DS beamformer. It involves the estimation of the noise covariance matrix in the estimation of the optimal weight vector. The optimal weight vector minimizes the total output signal power at each frequency bin while constraining the filtering response of the signal from the look direction to unity [34]. The effect of the MVDR beamformer is to attenuate the signals that are uncorrelated with the target signal and which are coming from directions different from the target. The MVDR beamformer is usually more effective than the DS beamformer when the interference signals are uncorrelated with the target signal, but it introduces more distortion to the target signal in the dereverberation as shown in our experimental results.

3.4 Spectral subtraction

The DS and MVDR beamformers function as spatial filters and effectively improve the reverberant speech in general. However, there is very low attenuation for the reflected signals and noise coming from a similar direction as the desired signal. A post-processing stage is usually adopted to further attenuate the residual reverberation and noise after the beamforming. There are many single-channel enhancement techniques that can be applied, and the spectral subtraction technique [14] has been popular for reliable performance. It uses the

exponential model of the room impulse response and the quasi-stationary characteristics of speech signals. In the formulation of amplitude spectral subtraction, both the a priori SNR and the a posteriori SNR are estimated based on a smoothing exponentially decaying form. In addition, the reverberation time is required in the formulation, and we applied the improved maximum-likelihood (ML) estimator [35] in our experiments.

4 DNN-based speech coefficients mapping with a dynamic feature constraint

Neural networks are universal function approximators that can model complicated nonlinear relationships between system inputs and outputs. The use of deep architecture neural networks, such as the DNN used in this paper, is efficient in representing complex relationships [22]. Generally speaking, given a set of input/output vector pairs, we can train a DNN to learn a mapping that accurately predicts the output given the input. In this section, we describe how to use a DNN to learn the mapping from reverberant and noisy speech coefficients to clean speech coefficients and the details of dynamic feature constraint.

4.1 DNN-based mapping

The task of DNN-based speech mapping is illustrated in Fig. 2. At the bottom of the figure, there is a sequence of input vectors, which in our study are speech coefficient vectors computed from reverberant and noisy speech. The type of speech coefficients is task-dependent and will be discussed in the next section. At the top of the figure, there is a sequence of desired output vectors that the DNN is trying to predict. In our study, the desired output vectors are the speech coefficient vectors computed from clean speech. The input vector sequence and the desired output vector sequence should be aligned at the frame level, and we call this data parallel data (also called stereo data in SPLICE [17]). The frame-level alignment requirement can be easily satisfied in speech dereverberation tasks. For example, if the reverberant and noisy speech is generated from clean speech by convolving with a known RIR and adding recorded noise samples, the generated distorted speech will be aligned perfectly with the clean speech. Another way to generate the parallel data is to record speech signals using close-talk and far-talk microphones simultaneously; then, the close-talk recording can serve as the DNN's desired output and, the far-talk recording(s) will be used as input to the DNN. Hence, it is relatively simple to generate a large amount of parallel training data.

A DNN consists of several hidden layers, each including a linear transform and a nonlinearity function, also called an activation function. The output of a hidden layer can be computed as

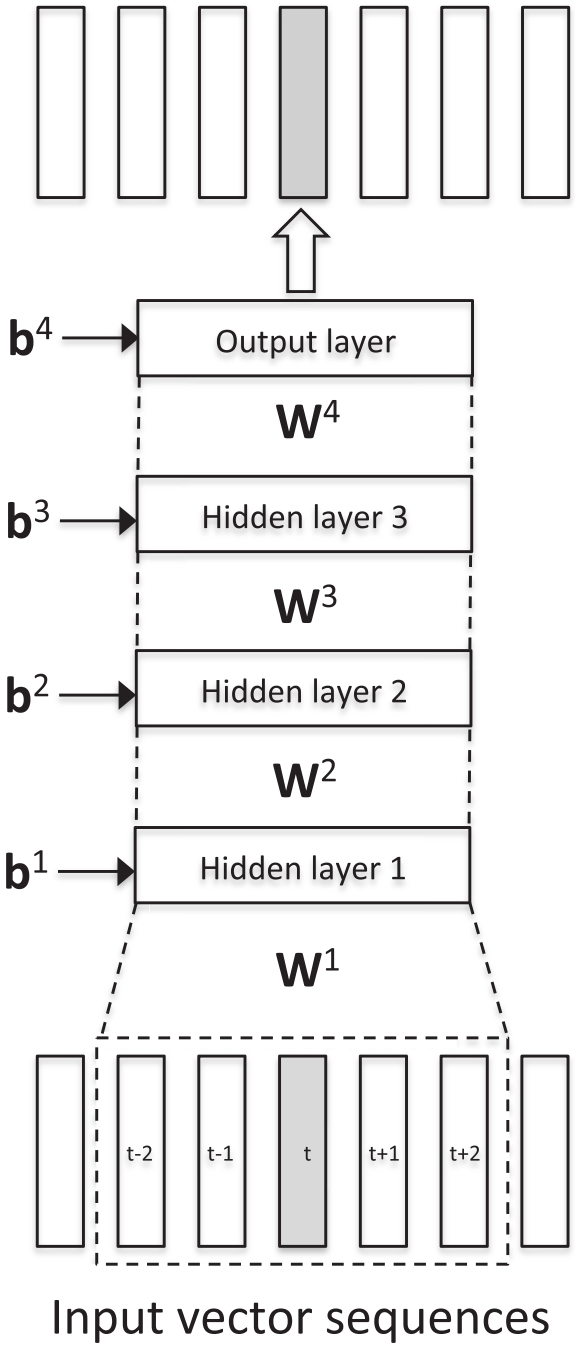

Fig. 2 The architecture of the DNN for speech coefficient mapping

$$\mathbf{a}^L = f(\mathbf{z}^L) \tag{2}$$

$$\mathbf{z}^L = \mathbf{W}^L \mathbf{a}^{L-1} + \mathbf{b}^L \tag{3}$$

where $\mathbf{z}^L$ and $\mathbf{a}^L$ are the input and output of the L^{th} hidden layer, and $f()$ is the activation function that operates

on the input matrix element by element. $\mathbf{W}^L$ is the vector weight connecting the nodes in layer $L-1$ to the nodes in layer L, and $\mathbf{b}^L$ is the bias of the nodes at layer L. In summary, the output of a hidden layer is obtained by first computing an affine transform of the previous layer's output and then applying the activation function. One popular choice for the activation function is the sigmoid function $f_{sig}(x) = 1/(1+e^{-x})$ which is used in this study.

In Fig. 2, there are three hidden layers in the illustrative DNN. The input vectors is the first affine transformed by $\mathbf{W}^1$ and $\mathbf{b}^1$, and then passed through the activation function to obtain the output of the hidden layer 1. The hidden layer 1 output is again affine transformed by $\mathbf{W}^2$ and $\mathbf{b}^2$ and passed through the activation function to get the output of hidden layer 2. This process continues with hidden layer 3. Finally, the output of hidden layer 3 is affine transformed by $\mathbf{W}^4$ and $\mathbf{b}^4$ of the output layer to generate the final output of the DNN. As the speech coefficient mapping task is a regression task and the output is real-valued rather than class labels as in classification problems, the output layer has a linear activation function $f_{lin}(x) = x$.

As a speech coefficient vector is computed from a single frame which is typically quite short (e.g., 25 ms), it is important to use a sequence of speech coefficient vectors as the input of the DNN. There are two reasons why the speech vectors from neighbouring frames are important to predict the clean speech vector of the current frame. First, without context information, it is difficult to determine the underlying phone identity of the current frame using only a short frame of 25 ms. One frame is not enough even for accurate phone classification, so it would not be sufficient for predicting the underlying clean vector. Second, due to the effect of reverberation, the current frame is affected by previous frames which may be dozens of frames away. The T60 time considered in normal meeting room can be as high as 1s, which is equivalent to 100 speech frames if the frame shift is 10 ms. This may suggest that there is a relationship between two frames that are as far away as 100 frames. Therefore, it is important to feed a local patch of speech vectors to DNN. Figure 2 shows the example of using five consecutive frames as DNN input. The optimal number of input vectors will be determined experimentally.

4.2 DNN training

Although DNN has a long pipeline of processing, it is obvious that the DNN's output is a highly nonlinear but continuously differentiable function of the input. Hence, the parameters of DNN, such as the weights and bias, can be trained from parallel data by using a gradient descent method. As the speech coefficient mapping is a regression task, it is natural to choose the cost function as the mean-squared error (MSE) between the DNN's output and the desired output, i.e., the speech coefficients of the underlying clean speech for current frame. The back-propagation (BP) algorithm is used to implement the gradient descent to train the DNN parameters. Specifically, we used the stochastic gradient descent (SGD) method to update the DNN after the model sees a few hundred of training frames. The SGD algorithm is known to work well for large-scale DNN training problems.

One of the major difficulties in training a DNN is the initialisation of the parameters. Usually, random initialisation will result in slow training, and often, the training will become trapped in a local minimum of the parameter space. To address this issue, RBM pretraining [21] is used to initialise the DNN's parameters before the MSE training. The RBM pretraining is an unsupervised learning and does not require any labeling of the training data. It tries to learn the intrinsic structure in the input, which is the reverberant and noisy speech in our study. The RBM pretraining initialises the DNN parameters to reasonable values and makes the MSE training (also called fine tuning) much easier. In our study, we have observed that the RBM training is critical to the successful training of a DNN with more than one hidden layer.

Another issue with DNN training is to choose the proper parallel training data. To achieve good test performance, it is required that the training environments should be sufficiently similar to the potential test environment. In many cases, it is not clear what the test environment is, so it may be a good approach to make the training data as diverse as possible. For example, the training of parallel data should cover different room RIRs, different noise types, different signal-to-noise ratios (SNR), and different speakers. The mapping problem is basically a many-to-one mapping, i.e., we want to map many versions of distorted speech coefficients that are different due to reverberation and noise characteristics to the same underlaying clean speech coefficients. In other words, if a clean speech signal is corrupted in 100 different ways, we want the DNN to map the 100 different speech coefficients to the same coefficients of the original clean speech signal. Such many-to-one mapping is difficult to achieve by conventional linear mapping methods such as fMLLR or piecewise linear mapping methods such as SPLICE due to their limited flexibility. A DNN has much more modelling power than linear transforms due to its nonlinear activation function and deep layered structure, allowing efficient representation of complex mapping [22]. In this study, we use the multi-condition training scheme defined for the ASR task to train the DNN for both speech enhancement and ASR feature compensation. As the multi-condition training data is generated from the clean-condition training data by adding reverberation and noise effects, we can obtain the perfect frame level alignment between the two corpus. There are about

30 h of parallel training data (15 h clean and 15 h distorted speech) for DNN training. The multi-condition data are distorted by multiple types of RIRs and noises; hence, it is expected to cover the test environments in the REVERB challenge evaluation data reasonably well.

4.3 Speech coefficients generation and preprocessing

The type of coefficients used as input and desired output of DNN mapping is task-dependent. For example, in speech enhancement, we use log-magnitude spectrum as the speech representation; it has a dimension of 257 when using a frame-length of 25 ms and a FFT length of 512. The log-magnitude spectrum is required to recover high-quality speech waveform for human listening. On the other hand, to enhance speech features for ASR, we only need to use the 39-dimensional MFCC as speech coefficient vectors, as finer spectral detail is not useful for differentiating speech classes.

It is usually beneficial to preprocess the DNN's input to achieve easier model training and better performance. We apply two stages of feature normalisation to the DNN inputs. In the first stage, for the speech enhancement task, we conducted experiments to understand whether utterance-wise mean normalisation should be applied to log-magnitude spectrogram to reduce channel effects on the data. For the ASR feature compensation task, utterance-wise cepstral mean and variance normalisation (CMVN) is applied to the MFCC features to reduce channel and reverberation effects. The use of CMVN is motivated by the observation that on the baseline ASR system, CMVN processed features produced better results than raw MFCC or CMN processed features. In the second stage, global mean and variance normalisation (MVN) is applied to normalise the distribution of the DNN inputs to zero mean and unit variance on a corpus level. The bias and scale used for the global MVN is also used to normalise any test data so that the training data and test data are processed in the same way.

4.4 Incorporation of dynamic feature constraint

A limitation of the DNN-based speech coefficient mapping scheme in Fig. 2 is that each frame is predicted independently, and we cannot guarantee that the predicted frame sequence is smooth and sounds natural [1]. Hence, we investigated two DNN speech-enhancement schemes that improve the smoothness of the enhanced spectrogram. As the proposed methods can also be used to features other than the log-magnitude spectrum, we will use the term "feature" to represent log-magnitude spectrum.

4.4.1 Least-squares estimation of static features

We train the DNN to predict both the clean features and their time derivatives from distorted features. Similar to time-derivative features in speech recognition, we used both delta features (first-order time derivative) and acceleration features (second-order time derivative). The delta and acceleration feature vectors are concatenated to the original feature vector (called the static feature) to form the new target vector for DNN learning. Hence, the target vector's dimension becomes three times of the original one. During the enhancement phase, the static, delta, and acceleration features are all predicted. The final enhanced features can be found by solving a least-squares (LS) problem described as follows.

Given a sequence of static feature vectors, we can approximate their time derivatives as follows [36]

$$\mathbf{y}_D(t) = \frac{\sum_{l=1}^{L} l \times (\mathbf{y}_S(t+l) - \mathbf{y}_S(t-l))}{\sum_{l=1}^{L} 2l^2} \tag{4}$$

where $\mathbf{y}_S(t)$ is the static feature vector at frame t, $\mathbf{y}_D(t)$ is the delta feature vector at frame t, L is the order of computing the derivatives and set to two in this study. The acceleration features can be obtained by applying Eq. (4) to the delta features. The delta and acceleration features are called dynamic features. They are the band-pass filtered versions of the static features that carry the temporal information of the static features. The DNN is trained to predict the static, delta, and acceleration features at the same time.

Let $\mathbf{Y} = [\mathbf{Y}_S, \mathbf{Y}_D, \mathbf{Y}_A]$ denote the output of the DNN which is a $\mathcal{T} \times 3\mathcal{M}$ matrix, where $\mathcal{T}$ is the number of frames in the current utterance and $\mathcal{M}$ is the dimension of the feature vectors. The subscripts S, D, and A represent the static, delta, and acceleration features, respectively. Since the DNN is not aware of the physical relationship between the three components of the target vector, the static, delta, and acceleration components of $\mathbf{Y}$ do not obey the physical relationship defined in (4). For speech enhancement purposes, we do not need the delta and acceleration features, but we can use them to improve the estimation of the static features.

Let $\mathbf{X}_S$ be the static-feature matrix we want to estimate from $\mathbf{Y}$. Let $\mathbf{X}_D$ and $\mathbf{X}_A$ be the delta and acceleration feature matrices of $\mathbf{X}_S$ computed according to (4). Hence, $\mathbf{X}_D$ and $\mathbf{X}_A$ are completely determined by $\mathbf{X}_S$ linearly as follows

$$\mathbf{X}_D = \mathbf{D}\mathbf{X}_S \tag{5}$$

$$\mathbf{X}_A = \mathbf{A}\mathbf{X}_S \tag{6}$$

where $\mathbf{D}$ is the $\mathcal{T} \times \mathcal{T}$ matrix that performs the linear filtering on $\mathbf{X}_S$ to generate the delta features. Similarly, $\mathbf{A}$ is the linear transform for generating the acceleration features from the static features. The parameters of $\mathbf{D}$ and $\mathbf{A}$ are determined by $\mathcal{T}$ and Eq. (4). The formulation of the delta and acceleration features as the linear transform of the static features is motivated by the work in [19].

We estimate $\mathbf{X}_S$ by solving the following least-squares problem

$$\hat{\mathbf{X}}_S = \arg\min_{\mathbf{X}_S} \mathcal{L}(\mathbf{X}_S) \quad (7)$$

$$\mathcal{L}(\mathbf{X}_S) = ||\mathbf{X}_S - \mathbf{Y}_S||_F^2 + w_D||\mathbf{X}_D - \mathbf{Y}_D||_F^2 + w_A||\mathbf{X}_A - \mathbf{Y}_A||_F^2 \quad (8)$$

where $||\mathbf{X}||_F^2$ is the Frobinus norm of $\mathbf{X}$, and w_D and w_A are the weights of cost contributed by the delta and acceleration features. Usually, we want to use a value larger than 1 for these two weights as the dynamic ranges of the delta and acceleration features are much smaller than those of static features. By investigating the variances of the log magnitude spectrum features, we found that the average variance of static features is about 20 and 114 times as large as that of the delta and acceleration features, respectively. Therefore, w_D and w_A are empirically set to 20 and 114, respectively, to make the static, delta, and acceleration features contribute similiarly to the cost function. From Eq. (7), we are searching for static features that obey the physical relationship between static, delta, and acceleration log spectrograms, and at the same time fit the DNN's output. Compared to the DNN's output $\mathbf{Y}_S$ which is estimated independently for each frame, the $\mathbf{X}_S$ will have better temporal continuity as it is estimated from the whole utterance.

The closed-form solution of the least-squares problem in Eq. (7) is

$$\hat{\mathbf{X}}_S = \mathbf{R}^{-1}\mathbf{P} \quad (9)$$

$$= \mathbf{R}^{-1}\mathbf{Y}_S + w_D\mathbf{R}^{-1}\mathbf{Y}_D + w_A\mathbf{R}^{-1}\mathbf{Y}_A \quad (10)$$

$$\mathbf{R} = \mathbf{I} + w_D\mathbf{D}^T\mathbf{D} + w_A\mathbf{A}^T\mathbf{A} \quad (11)$$

$$\mathbf{P} = \mathbf{Y}_S + w_D\mathbf{Y}_D + w_A\mathbf{Y}_A \quad (12)$$

where $\mathbf{I}$ is a $\mathcal{T} \times \mathcal{T}$ identity matrix. The LS solution can be interpreted as a weighted sum of transformed static, delta, and acceleration features as shown in Eq. (10). Hence, it is clear that enhanced delta and acceleration features will help improve the static features.

The LS solution requires the inversion of the $\mathcal{T} \times \mathcal{T}$ matrix $\mathbf{R}$, and the computational cost of the matrix inversion can be high when $\mathcal{T}$ is large. In practice, we can reduce the computational cost by breaking a long sentence into several segments during silence frames, and find the LS solution for each segment independently. Another way to reduce the computational cost is to store the inversion $\mathbf{R}^{-1}$ for every possible length $\mathcal{T}$.

4.4.2 Sequential training of DNN

The LS solution presented in the previous section requires expensive matrix inversion. In this section, we present another method that directly optimizes the DNN using a sequential cost function similar to Eq. (8) in which no LS post-processing is necessary.

The sequential cost function is illustrated in Fig. 3. Note that to save space, only the last two layers of the DNN is drawn. After the static features are predicted by the DNN, the delta and acceleration features are obtained from the static feature vector sequence by applying the $\mathbf{D}$ and $\mathbf{A}$ matrices. Due to the need to compute the dynamic spectra, we cannot randomize the training frames and need to train the DNN sequentially. One simple way is to use each utterance as a minibatch. The predicted static, delta, and acceleration spectra are then used to compute the cost function

$$\mathcal{L}_{\text{DNN}}(\mathbf{X}_S) = ||\mathbf{X}_S - \mathbf{Z}_S||_F^2 + w_D||\mathbf{D}\mathbf{X}_S - \mathbf{Z}_D||_F^2 + w_A||\mathbf{A}\mathbf{X}_S - \mathbf{Z}_A||_F^2 \quad (13)$$

where $\mathbf{Z}_S$, $\mathbf{Z}_D$, and $\mathbf{Z}_A$ are the static, delta, and acceleration spectra of the clean target speech signal. By optimizing the DNN using the sequential cost function (13), the

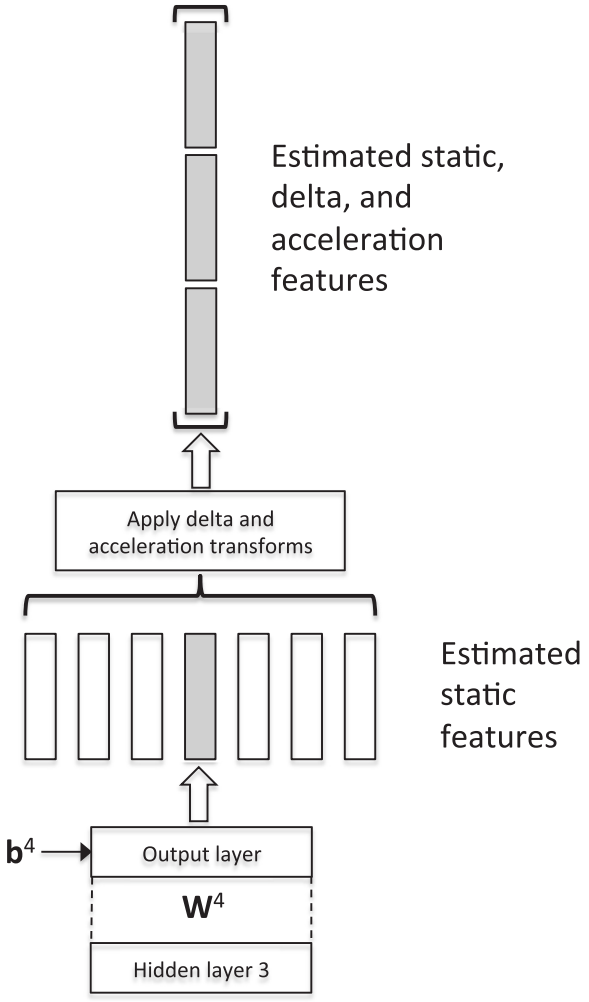

Fig. 3 The architecture of the DNN coefficient mapping using a sequential training cost function

predicted static spectra will tend to have delta and acceleration features that are close to the clean (natural) one. The use of a sequential cost function can be seen as a special case of multi-task learning where the error gradient of the delta and acceleration spectra prediction are used to help train the static feature prediction. One advantage of this approach over the LS solution discussed in the previous section is that there is no need to perform matrix inversion. In practice, we use the sequential training to fine-tune the DNN trained to predict static and/or dynamic features.

5 Cross transform feature adaptation

In the DNN-based speech coefficient mapping, parallel data of clean and distorted speech is required to train the mapping network. In this section, we study the scenario where only the clean speech signal is available at the system-building phase for ASR. This is also known as clean-condition training of an acoustic model. As a result, there is a significant mismatch between the clean-trained model and the distorted test speech features, and the speech recognition performance will be significantly affected.

There are two major approaches to reduce the mismatch between the clean model and distorted test features. One is through the model adaptation approaches, which adapt the clean model towards the distorted speech features. Some examples of the model adaptation method include the (constrained) maximum likelihood linear regression (CMLLR/MLLR) [30], maximum a posteriori (MAP) [37], and vector Taylor series (VTS) [38–40] based adaptation methods. Another approach to reduce train/test mismatch is to adapt the distorted features towards the clean model. Examples include the feature space maximum-likelihood linear regression (fMLLR, equivalent to global CMLLR), SPLICE feature compensation [17], etc. In this paper, for the clean-condition training scheme of the ASR task, we apply a new feature adaptation method called cross transform that we proposed in [31]. For each test utterance, speaker, or condition, a linear transform is estimated to adapt the test features towards the clean acoustic model. For completeness, we briefly describe the cross transform in the following section.

5.1 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. 4. Linear transformation uses all dimensions of the current frame to predict new features that fit the acoustic model under the maximum-likelihood (ML) criterion [30, 41]. On the other hand, temporal filtering uses the context information in neighboring frames to estimate features that fit the acoustic model [42–45]. 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.

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

$$\mathbf{y}_t = \sum_{\tau=-L}^{L} \mathbf{B}_\tau \mathbf{x}_{t+\tau} + \mathbf{c} = \mathbf{W}\tilde{\mathbf{x}}_t, \tag{14}$$

where $\mathbf{x}_t$ and $\mathbf{y}_t$ are D-dimensional input and output feature vectors, respectively. $\mathbf{B}_\tau$, $\tau = -L, \ldots, L$ are the transformation matrices, and $\mathbf{W} = [\mathbf{B}_{-L}, \ldots, \mathbf{B}_L, \mathbf{c}]$ and $\tilde{\mathbf{x}}_t = \left[\mathbf{x}_{t-L}^T, \ldots, \mathbf{x}_{t+L}^T, 1\right]^T$ are the concatenated transformation matrices and inputs, respectively. Although the transform in (14) is possible in theory, it is hard to apply 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 reliably estimate 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. 5. The simplified transform is simply the combination of the linear transform and temporal filter illustrated in Fig. 4. As the shape of the transform often looks like a cross, we call it the cross transform.

Similar to maximum normalized likelihood linear filtering in [45], the parameters of the cross transform can be estimated by minimizing an approximated KL divergence between the distribution of the 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 $\Lambda = \{c_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j | j = 1, \ldots, J\}$ and $\boldsymbol{\Sigma}_j$ being diagonal. The optimal $\mathbf{W}$ is found by minimizing the following approximated KL divergence

$$\begin{aligned} f(\mathbf{W}) = & -\frac{\lambda}{2}\log\det\left(\mathbf{W}\boldsymbol{\Sigma}_{\tilde{\mathbf{x}}}\mathbf{W}^T\right) + \frac{\beta}{2T}||\mathbf{W} - \mathbf{W}_0||_2 \\ & -\frac{1}{T}\sum_{t=1}^{T}\log\sum_{j=1}^{J} c_j \mathcal{N}(\mathbf{W}\tilde{\mathbf{x}}_t; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \end{aligned} \tag{15}$$

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. $\mathbf{W}_0$ is the initial value of the weight matrix

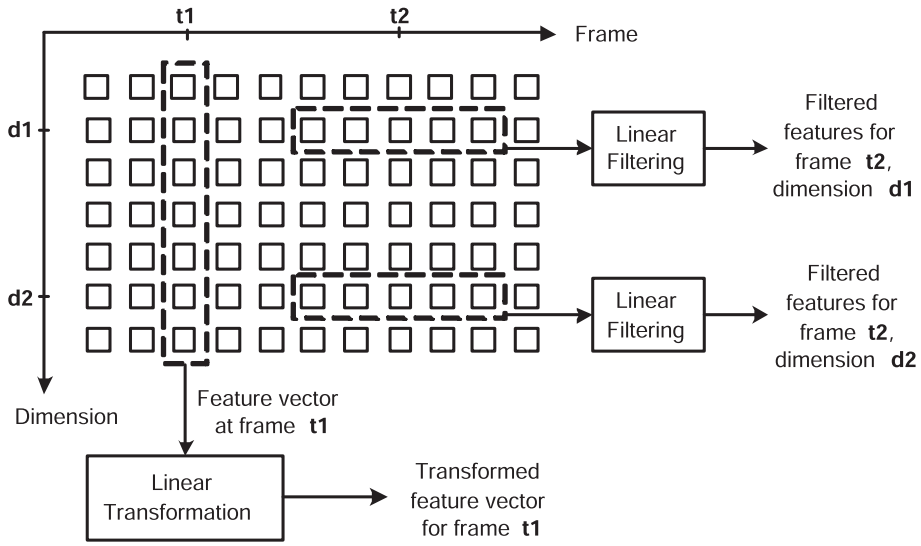

Fig. 4 Temporal filtering of feature trajectories versus linear transformation of feature vectors

$\mathbf{W}$. When constructing $\mathbf{W}_0$, the sub-matrix $\mathbf{B}_\tau$ contains all 0's except that $\mathbf{B}_0$ is an identity matrix. In this way, the application of $\mathbf{W}_0$ will result in the same output features as the input features. 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, the covariance matrix of the transformed features is encouraged to have a large log-determinant to prevent the variance of the transformation features from shrinking. The cost function can be iteratively minimized via an EM algorithm. For the detailed solution of the cross transform, readers are referred to [31].

5.2 Comparison between cross transform and DNN-based feature compensation

A major difference between the feature adaptation methods (like fMLLR and cross transform) and the DNN feature compensation is how the free parameters are

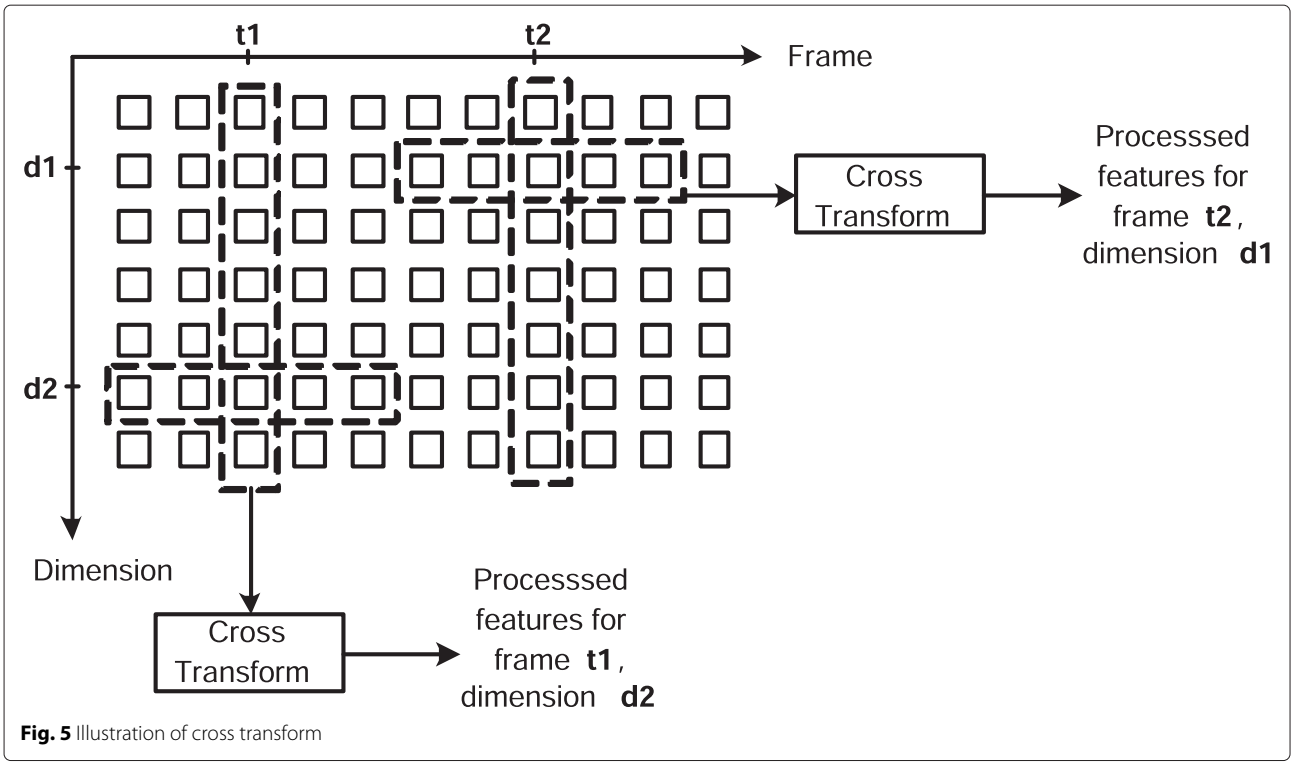

Fig. 5 Illustration of cross transform

estimated. The parameters of the feature adaptation are learned from the test data itself and hence dynamically updated during test, while the DNN parameters are trained from parallel training data and fixed once trained. When the test environment is known during system building and it is possible to collect data with similar characteristics for training, using DNN-based feature compensation is a suitable approach. On the other hand, feature-adaptation methods such as the cross transform does not require the knowledge of the test environment and availability of parallel training data, making them universally applicable in all test environments. However, the linear transforms used in fMLLR and the cross transform are less powerful than DNN.

6 Experiments

The techniques are evaluated on the REVERB Challenge 2014 tasks [46]. In this section, we first describe the task and evaluation metrics, followed by the implementation parameter settings for the techniques. Then, we present the evaluation results on both speech enhancement and ASR tasks.

6.1 Task, data, and evaluation metrics

The REVERB Challenge 2014 [46] is a benchmark task to evaluate dereverberation techniques for both human listening and ASR. The distortion considered is mainly reverberation, with a moderate amount of additive background noise. There are two types of evaluation data. One type is simulated reverberant and noisy speech, generated by adding noise and reverberation to clean utterances from the WSJCAM0 [47] corpus, which is the British version of the WSJ0 corpus [48]. The other type is a real meeting recording from the MC-WSJ-AV [49] corpus, which is the re-recorded version of WSJCAM0 in a meeting-room environment. Both types of data are provided for system development (dev set) and evaluation (eval set). In this experiment, we report results on the eval set only.

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 [47] corpus, while the multi-condition training data is artificially generated by corrupting the clean-condition training data in a similar way as the generation of simulated test data.

Speech enhancement methods are evaluated by several metrics, such as cepstral distance (CD) [50], log-likelihood ratio (LLR) [50], frequency-weighted segmental SNR [50], speech-to-reverberation modulation energy ratio (SRMR) [51], and optional PESQ [52]. For real room 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 do not have a PESQ license, we will not report PESQ results in this paper. For more details of the corpus and tasks, please visit the REVERB Challenge website (http://reverb2014.dereverberation.com).

6.2 Settings of methods

6.2.1 DS beamformer

In the implementation of the DS beamformer, we segmented the speech utterances into segments of 64 ms long with 75 % overlap. We then applied pre-emphasis and a Hanning window of 64 ms long to the segments. A 1024-point short-time Fourier transform (STFT) was used to obtain the frequency-domain signals. The frequency components of the array channels were aligned according to the TODAs computed by the GCC-PHAT method [33] where the first channel in each record was selected as the reference channel. Since the maximum delay between any microphone pairs in the array is 0.59 ms, the TDOAs in samples were calculated from the cross-correlation peaks that appeared within 0.59 ms. The speed of sound was set to 340 m/s in our implementation. The aligned channels were summed with a normalization to obtain the frequency components of the DS output. Then, we applied the inverse STFT and the overlap-save method to obtain the time-domain signal of 16 ms long. The final output of the DS beamformer was obtained after the de-emphasis. The output of the DS beamformer served as the input signals for the spectral subtraction module and the DNN module.

6.2.2 MVDR beamformer

In the implementation of the MVDR beamformer, we used the same window size and overlap as in the DS beamformer. The frequency-domain components of all the array channels and the TDOAs were obtained similarly as in the DS beamformer. To estimate the noise covariance matrix, we applied an energy-based voice activity detector to detect the absence of speech signals. When the speech signals are absent, the noise covariance matrix can be updated.

6.2.3 Spectral subtraction

The spectral subtraction algorithm [14] was implemented separately from the beamformers. Only the outputs of the beamformers were passed to the spectral subtraction algorithm. In the spectral subtraction algorithm, we obtained the frequency-domain signals using the STFT of 256 points with pre-emphasis and a Hanning window of 16 ms long and 75 % overlap. The reverberation time used in the algorithm was estimated from the beamformer

output by the ML method [35]. Noting that this estimator takes the advantage of a long input signal, we repeatedly refined the reverberation time estimation by using the up-to-date output signals of the beamformers for each utterance.

6.2.4 DNN-based speech coefficient mapping

The DNN-based speech coefficients mapping is applied to both log-magnitude spectrogram enhancement for speech enhancement task, and also MFCC enhancement for ASR task. For speech enhancement, the speech signal is transformed into frequency domain using a 25-ms Hamming window and 10-ms frame shifts. The logarithm of the magnitude spectrum has a dimensionality of 257 and is used as the input features of the DNN, and each input is a concatenation of multiple frames. The enhanced log-magnitude spectrum produced by the DNN is combined with the original distorted phase spectrum to re-synthesize the enhanced speech signal.

For ASR, a concatenation of the MFCC feature vectors is used as the DNN input. Each MFCC feature vector has 39 dimensions, including the first 13 cepstral coefficents (c0–c12) and their first and second derivatives. The MFCC features are normalized by utterance-based CMVN and then enhanced by DNN-based speech coefficient mapping. The DNN takes 15 frames of MFCC as input, hence the input layer dimension is $15 \times 39 = 585$. We empirically choose to use three hidden layers with 2048 hidden nodes in each layer. The output layer is 39-dimensional, including both the static and dynamic features. The feature compensation DNN is pretrained using RBM training and refined using the MSE criterion. The enhanced MFCC features are used for both ASR training and testing.

When there are multiple input channels, DS or MVDR beamformers are applied to produce the enhanced single-channel speech signal first, and then DNN-based speech coefficient mappings are applied on the single-channel signal.

6.2.5 Cross transform feature adaptation

The cross transform uses a context size of frames (i.e., $L = 16$) to capture long-term temporal information to effectively suppress reverberation. The clean reference GMM model is trained from the clean-condition training data with about 4400 Gaussian components. The hyper-parameters β and λ are set to 1 and 0.9, respectively.

6.3 Speech enhancement

We evaluate the DNN approach for speech enhancement and compare it to the approach of DS beamformer plus spectral subtraction (SS). We report the objective measures in this section.

6.3.1 Effect of input context size and hidden layer size

Table 1 shows the performance of speech enhancement obtained by three DNNs of different input context sizes and hidden layer sizes. A single microphone input is used and the performance of SS is also shown for comparison. By comparing DNN1 and DNN2, it is observed that using longer input context and larger hidden layer size improves the performance metrics in most cases except for SRMR of real test data. The poorer results on real rooms may be because that the DNNs are trained from simulated room data and hence do not generalise well on real room due to mismatch in distortion characteristics. From DNN2 to DNN3, the network input is further increased from 15 to 19 frames. However, the results of DNN3 is similar to or even slightly worse than that of DNN2. This may suggest that 15 frames are enough for speech dereverberation, or the DNN3 model is overfitted to the training data as it contains more parameters. Therefore, we decided to use 15 frames as the input context size and three hidden

Table 1 Performance of speech enhancement methods using single-channel input on simulated and real room recordings. The results are averaged over near and far test cases

Processing				Simulated rooms							Real
				CD		SRMR	LLR		SNR		SRMR
Name	Architecture	CMN	-	mean	med.	mean	mean	med.	mean	med.	mean
None	-	No		3.97	3.69	3.68	0.58	0.51	3.62	5.39	3.18
SS	-	No		3.82	3.51	4.0	0.56	0.51	4.75	6.89	3.94
Effect of input context size and hidden layer size											
DNN1	11x257-2048-2048-2048-257	Yes		2.64	2.41	5.78	0.52	0.48	7.19	8.09	4.54
DNN2	15x257-3072-3072-3072-257	Yes		2.53	2.31	5.80	0.51	0.47	7.54	8.31	4.36
DNN3	19x257-3072-3072-3072-257	Yes		2.50	2.28	5.77	0.50	0.47	7.55	8.35	4.36

"mean" represents the mean value of the scores of utterances, while "med." represents the median of the scores. "CMN" indicates whether utterance-wise mean normalization is applied during both DNN training and testing. A DNN architecture of "11x257-2048-2048-2048-257" means that 11 frames of 257D log-magnitude spectrum are used as input, followed by three hidden layers each with 2048 nodes, and the output layer predicts the 257D log-magnitude spectrum of clean speech

layers, each with 3072 sigmoid nodes, for the following experiments.

6.3.2 Effect of beamforming

The performance comparisons between MVDR and DS beamforming are shown in the first two rows of Table 2. It is observed that DS produces better results than MVDR except for SRMR. This could be because DS produces less distortions than MVDR in reverberant environments. It is also observed that when SS is applied after DS, significantly better results are obtained than DS alone except for LLR.

Both MVDR and DS beamforing are used as the preprocessing unit of the DNN-based speech enhancement (DNN4). It is observed that DS+DNN4 generally produces better results than MVDR+DNN4. This shows that the DNN-based speech enhancement should work with a beamformer with low distortions. Therefore, we will use DS as the preprocessor for DNN-based speech enhancement in the following experiments.

6.3.3 Effect of CMN preprocessing on different evaluation metrics

One important question for DNN-based speech dereverberation is whether the CMN should be applied for each training and testing utterance. The motivation of using CMN is that it may reduce some variations in the training data and make the mapping from the distorted speech spectrum to the clean speech spectrum easier. We show the performance of speech enhancement with and without CMN preprocessing in Table 2. By comparing DS+DNN4 and DS+DNN5, the CMN has a significant but mixed effect on the evaluation metrics. Specifically, the SNR and LLR are much better without CMN, while CD is better with CMN applied. In addition, applying CMN causes SRMR degradation for simulated data but the reverse is true for real data.

To understand the different trends, it is necessary to look into the detailed implementation of the evaluation metrics. For CD computation, as CMN is also applied to the reference (clean) and enhanced cepstral coefficients before the calculation of distance, applying CMN to both input and target of DNN training leads to lower CD than without CMN. On the other hand, CMN is not applied when computing LLR and SNR, so applying CMN on the target log-magnitude spectrum during DNN training effectively causes a mismatch between the enhanced spectrum and the reference spectrum. This mismatch is similar to filtering the speech with a channel. Therefore, DS+DNN4 performs even worse than the

Table 2 Performance of speech enhancement methods using eight channel inputs on simulated and real room recordings. The results are averaged over near and far test cases

Processing				Simulated rooms							Real
				CD (dB)		SRMR	LLR		SNR (dB)		SRMR
Name	Target /train	Estimation	CMN	mean	med.	mean	mean	med.	mean	med.	mean
MVDR	-	-	No	3.64	3.28	4.85	0.48	0.43	5.31	7.76	4.12
DS	-	-	No	3.11	2.76	4.34	0.410	0.360	6.60	9.24	3.84
DS+SS	-	-	No	3.00	2.67	4.56	0.410	0.360	7.16	9.68	4.62
Effect of beamforming											
MVDR+DNN4	257/random	-	Vis/Tgt	2.28	2.07	5.88	0.470	0.440	8.44	8.88	4.51
DS+DNN4	257/random	-	Vis/Tgt	*2.01*	*1.85*	*5.92*	0.467	0.440	8.56	8.88	4.40
Effect of CMN											
DS+DNN5	257/random	-	No	2.15	1.96	4.80	0.205	0.155	12.07	12.55	*4.95*
DS+DNN6	257/random	-	Vis	2.18	2.00	5.27	0.235	0.198	11.11	11.49	4.24
DS+DNN4a	257/random	-	Vis/Tgt2	2.02	1.86	5.16	0.278	0.237	10.93	11.55	3.76
Effect of dynamic features											
DS+DNN7	3x257/random	Use static	No	2.15	1.96	4.88	0.205	0.158	12.04	12.59	*4.95*
DS+DNN7LS	3x257/random	LS (9)	No	2.07	1.90	4.83	0.195	0.150	12.42	12.99	4.78
DS+DNN8	257-3x257/seq.	Use static	No	2.04	1.86	4.61	*0.193*	*0.147*	*12.64*	*13.22*	4.62
Effect of clean phase											
DS+DNN8c	257-3x257/seq.	Use static	No	1.84	1.66	4.90	0.165	0.123	13.37	13.88	-

"mean" represents the mean value of the scores of utterances, while "med." represents the median of the scores. "random" means that the frames of each minibatch of DNN training are randomly selected from the whole training corpus, while "seq." refers to the case in which each utterance is used as a minibatch. An output size of "3x257" means that the static, delta, and acceleration spectra are all predicted. The target size of "257-3x257" refers to the sequential training in Section 4.4.2. All DNNs use 15 frames of input and 3072 nodes per hidden layer. DNN8c is the same as DNN8 except that clean phase is used. Best results of each metric (not including DNN8c) are shown in italics

DS baseline for LLR and the median value of SNR. To reduce the mismatch, we add back the average clean log-magnitude spectrum (a single vector of 257 dimensions) to all enhanced log-magnitude spectra (see DS+DNN4a). The results show that this leads to significant improvement of LLR and SNR, but degardation for SRMR. We also tried another setting, i.e., only apply CMN to the input to reduce channel mismatch (DS+DNN6). The results are even worse than not applying CMN at all except for SRMR.

In summary, we found that not applying CMN leads to significantly better LLR and SNR, moderate degradation to CD, significant degradation to SRMR on simulated data but improvement to SRMR on real data. Hence, we will not apply CMN in the following experiments.

6.3.4 Effect of dynamic feature estimation

We now investigate the effect of predicting dynamic log-magnitude spectra and using them to improve the static spectra. The results are shown in the rows DNN7, DNN7LS, and DNN8 of Table 2. The DNN7 is trained to predict both the static and the dynamic log-magnitude spectra. During testing, only the predicted static spectra are used for resynthesizing waveform. By comparing DNN7 and DNN5, we observe no significant difference in performance due to the additional task of predicting the dynamic spectrum in DNN7. This shows that the DNN is able to predict the static and dynamic spectrum simultaneously without degrading the performance of static spectrum prediction. The DNN7LS uses the same trained DNN as DNN7, except that the static spectra are obtained by using the LS estimation of Eq. (9). The results show that DNN7LS outperforms DNN7 in terms of CD, LLR, and SNR, but degrades SRMR a bit, especially for real data. DNN8 is trained by using the sequential training cost function in Section 4.4.2. Compared to DNN7LS, DNN8 further improves CD, LLR, and SNR, but also further degardes the SRMR. An important advantage of DNN8 is that it does not need to perform the computationally expensive LS estimation during test time. The detailed results of DS+DNN8 are listed in Table 3.

We also show the mean-squared errors (MSE) between the clean log-magnitude spectra and its DNN-predicted versions to get a better understanding of how the dynamic spectra help to improve the prediction of the static spectrum. The MSE averaged over the six simulated test conditions of eval set are shown in Table 4. Here, the log-magnitude spectra of the enhanced speech is taken from the DNN's output layer directly to avoid the effect of imperfect phase. For DS+DNN7LS, the static log-magnitude spectrum are first obtained by finding the LS solution of (9), then the static and acceleration spectra are obtained by applying Eq. (4). By comparing DS+DNN7 and DS+DNN7LS, we can conclude that the LS solution not only reduces the MSE of the static spectra but also the delta and acceleration spectra. This shows that the LS solution is an effective way to exploit the physical relationship between the static and dynamic spectra to produce a better prediction of the clean speech's spectrum. The sequential training produces similar results as the LS solution. This is achieved with just passing the input through the DNN without the LS post-processing which may be computationally expensive.

Table 3 Detailed results obtained by DS+DNN8 on the six test conditions

Cases	Simulated rooms				Real
	CD (dB)	SRMR	LLR	SNR (dB)	SRMR
far1	1.64/1.52	4.85	0.12/0.09	14.33/14.52	4.81
far1	2.76/2.49	4.83	0.31/0.24	10.18/11.36	-
far1	2.44/2.21	4.16	0.28/0.22	10.30/11.13	-
near1	1.44/1.33	4.72	0.09/0.07	15.15/15.34	4.43
near1	2.00/1.82	4.60	0.17/0.12	13.04/13.64	-
near1	1.97/1.80	4.49	0.19/0.14	12.82/13.30	-
avg.	2.04/1.86	4.61	0.193/0.147	12.64/13.22	4.62

For CD, LLR, and SNR, the numbers before and after "/" represents the mean and median of the scores, respectively

6.3.5 Subjective listening

We performed informal listening of the enhanced speech. Perceptually, it is found that the SS and DS+SS methods produces moderate reduction of reverberation and also improve speech qualities. The DNN-based system significantly removes the reverberation from the speech. However, it also degrades the speech quality, especially when the distortion is strong. To find out which part of the DNN system causes the quality degradation, we use the DNN8 predicted magnitude spectrum together with the phase spectrum of the clean speech to re-synthesize the enhanced waveform. The results are presented as DNN8c in Table 2. It can be seen that the use of clean phase produces a significant improvement to all objective measures. An informal listening test also confirmed that the waveform reconstructed from DNN enhanced magnitude and clean phase is much better than that from DNN-enhanced magnitude and original reverberant and noisy

Table 4 Mean-squared error (MSE) between the log-magnitude spectra of reference (clean) and DNN enhancement

Processing	Static	Delta	Acceleration
DS	226.21	10.76	2.06
DS+DNN5	91.81	6.53	1.31
DS+DNN7	91.34	6.28	1.22
DS+DNN7LS	*87.70*	*5.99*	*1.18*
DS+DNN8	88.77	*5.99*	1.20

The enhanced spectra are taken from the DNN's output layer without going through the resynthesis to avoid the effect of imperfect phase. The MSE is averaged over all the six simulated test conditions of eval set

phase, especially when the reverberation is strong, such as in far rooms. The analysis shows that it is necessary to improve the phase in addition to the magnitude.

6.4 Speech recognition

6.4.1 Clean-condition training

We used the HTK-based ASR system from the REVERB Challenge 2014 organizer for evaluation on clean-condition training. Detailed results are shown in Table 5. From the table, we have several observations. First, cross transform (CT) reduces the WER significantly for both the single-channel scenario and multiple-channel scenario where it is used after MVDR beamforming. In addition, the improvement is obtained for both simulated and real data. This verified the effectiveness of the cross transform for real reverberant recordings. Second, the cross transform performs similarly to the 256-class CMLLR model adaptation (MA), despite that many more free parameters are being used in CMLLR than in cross transform. Third, if cross transform feature adaptation and the CMLLR model adaptation are applied in sequence, the WER is further reduced. This shows that although both cross transform and CMLLR are linear transforms of the features, they are complementary to each other because they use different information. Specifically, cross transform uses both the spectral and temporal information (up to 0.33 s) while CMLLR mainly uses spectral information and limited temporal information (up to 0.1 s). Another reason that cross transform and CMLLR are complementary may be due to that one is applied in utterance mode and the other is in batch mode. In utterance mode, the transform is able to adapt to each speaker and utterance-specific distortion characteristics, while in batch mode the transform is adapted to the average speaker and distortion. In summary, the results on cross transform show that long-term temporal information is important in improving the speech recognition performance of reverberant speech.

6.4.2 Multi-condition training

For the multi condition training scheme, we developed an ASR system based on the Kaldi toolkit [53] and DNN acoustic model. This is because when multi-condition training is used, a DNN acoustic model provides much better results than the GMM acoustic model. The DNN-based acoustic model uses 9 frames of 39D MFCC features as input, and we found that using longer context did not lead into better results. There are seven hidden layers in the model, each layer with 2048 sigmoid hidden nodes. The output layer of the DNN contains about 3500 classes, i.e., the number of tied triphone states. We use seven hidden layers and 2048 nodes. 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 [53]. Note that the DNN acoustic model is different from the DNN-based speech coefficient mapping described in Section 4. In the following experiments, the MFCC features are first enhanced by the DNN-based

Table 5 ASR performance (WER) using clean condition training data on the evaluation data

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*
Two 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*
Eight 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*

CT stands for cross transform, while MA refers to the 256-class based CMLLR model adaptation

speech coefficient mapping, and then fed to the DNN acoustic model for speech recognition.

The performance of the ASR system using multi-condition training is shown in Table 6. For comparison, we also show two results obtained from the clean-condition trained DNN acoustic model. From the results, we have several observations. First, by comparing Rows 1 and 3, the DNN acoustic model itself is not robust against reverberation distortion. The robustness is coming from the multi-condition training. Second, when the multi-condition training data is used to train the DNN-based feature mapping, and the acoustic model is still clean-trained (Row 2, MAP = Yes), the performance is close to when the multi-condition training data are directly used to train the DNN acoustic model (Row 3). In addition, if both DNNs are trained from multi-condition training data, we obtained the best results for the single-channel scenario (Row 4). This shows that it is useful to use two DNNs for reverberant speech recognition, one for feature compensation, and one for acoustic modelling. This observation is true for the 2-channel and 8-channel scenarios also. There may be two reasons for the usefulness of DNN feature compensation. One is that DNN feature compensation uses more information than the 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 the DNN acoustic model automatically discover useful features for speech recognition.

7 Conclusions

In this paper, we studied two methods for speech dereverberation for both speech enhancement and ASR tasks. In the DNN-based speech coefficient mapping, parallel training data of reverberant speech (observation) and clean speech (desired output) are used to train the DNN to predict clean speech. This mapping approach is applied to both speech enhancement and ASR feature enhancement tasks. For speech enhancement, we also proposed a LS postprocessing and a sequential cost function to incorporate the constraint of dynamic features to improve the smoothness of the enhanced magnitude spectrum. Results in both tasks show the effectiveness of the DNN-based mapping. The proposed LS postprocessing and sequential cost function improves the CD, SNR, and LLR evaluation metrics but cause slight degradation for SRMR. We also noticed that the DNN mapping causes distortion to enhanced speech waveforms especially when the reverberation is strong. This is partially due to the reverberant phase spectrogram being used with the DNN-enhanced magnitude spectrogram to re-synthesise the speech waveforms. It will be interesting to study the possibility of predicting the clean phase spectrogram together with the clean magnitude spectrogram.

We also studied the cross transform feature adaptation method that does not rely on parallel training data and can be applied to any unknown test environment for ASR. The cross transform only relies on the information of the clean reference model (a GMM in our study) and the current test utterance or speaker, and hence is not restricted by the training/testing environment mismatch. To effectively remove reverberation effects from the speech features, the cross transform uses long-term temporal information as input (up to 0.33 s), which is much longer than traditional feature/model adaptation methods such as fMLLR/CMLLR. Experimental results

Table 6 WER on the evaluation data using DNN based acoustic models. "MAP" refers to the DNN based MFCC feature mapping

MAP	Simulated rooms						Real		Avg.
	Room1A		Room2A		Room3A		Room1		
	near	far	near	far	near	far	near	far	
Single channel + Clean condition training									
No	10.6	19.3	23.2	69.3	30.2	74.6	68.0	66.2	45.2
Yes	9.3	10.6	12.7	21.4	16.5	25.1	40.2	39.0	*21.9*
Single channel + Multi condition training									
No	8.7	9.4	10.5	16.5	13.4	20.0	35.4	34.3	18.5
Yes	8.9	8.8	8.8	13.9	11.4	15.5	32.2	32.7	*16.5*
MVDR(2ch) + Multi condition training									
No	8.6	9.6	9.1	14.9	11.6	18.3	33.3	30.7	17.0
Yes	8.5	8.6	7.9	12.4	10.1	14.8	29.1	29.1	*15.1*
MVDR(8ch) + Multi condition training									
No	7.8	8.3	8.3	10.8	9.8	13.3	24.8	25.1	13.5
Yes	7.5	8.2	7.4	9.7	8.9	11.3	22.7	24.4	*12.5*

confirmed the effectiveness of cross transform in improving the ASR performance in real reverberation test scenarios. In addition, the cross transform is complementary to CMLLR model adaptation as they use different information sources.

In the future, we will continue to pursue the two research directions we studied in this paper. For DNN-based speech coefficient mapping, one important question is how quickly the prediction accuracy degrades as the test condition deviates from the training conditions. Another question is how much diverse training data can the DNN-based mapping learn well. Both questions are related to whether such a pretrained mapping machine is a practical solution to real-world speech enhancement and ASR tasks where the test environment is usually unpredictable. For the cross transform, we may extend it in several ways. One way is to use a nonlinear transform such as an MLP to replace the linear transform to increase the flexibility of the method. This is motivated by the fact that the distortion in the cepstral domain is usually highly nonlinear. Another way is to use a more powerful clean reference model to learn the intrinsic structure of the speech features. It is the clean reference model that provides guidance to the transform estimation; hence, a better reference model could lead to a more accurate translation estimation.

Endnote

[1]Even if recurrent neural network is used to predict clean speech coefficients, there is no explicit constraint on the temporal structure of the predicted coefficient trajectories.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

XX designed the DNN based speech enhancement systems and performed experiments. He also contributed in the design of the cross transform and coordinated the works done in this paper. SZ implemented the beamforming modules and performed related experiments with the help of XZ. DHHN designed and performed experiments on cross transform. DLJ, ESC and HL helped to revise the manuscript and approved it for publication. All authors read and approved the final manuscript.

Author details

[1]Temasek Lab@NTU, Nanyang Technological University, 50 Nanyang Drive, 637553 Singapore, Singapore. [2]Advanced Digital Sciences Center, 1 Fusionopolis Way, 138632 Singapore, Singapore. [3]School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, 639798 Singapore, Singapore. [4]Department of Human Language Technology, Institute for Infocomm Research, 1 Fusionopolis Way, 138632 Singapore, Singapore.

Received: 22 February 2015 Accepted: 28 December 2015
Published online: 13 January 2016

References

1. TH Li, Estimation and blind deconvolution of autoregressive systems with nonstationary binary inputs. J. Time Ser. Anal. **14**(6), 575–588 (1993)
2. R Chen, TH Li, Blind restoration of linearly degraded discrete signals by gibbs sampling. IEEE Trans. Signal Process. **43**, 2410–2413 (1995)
3. O Cappe, A Doucet, M Lavielle, E Moulines, Simulation-based methods for blind maximum-likelihood filter deconvolution. IEEE Trans. Signal Process. **73**(1), 3–25 (1999)
4. S Gannot, M Moonen, Subspace methods for multimicrophone speech dereverberation. EURASIP J. Appl. Signal Process. **2003**(11), 1074–1090 (2003)
5. M Triki, DTM Slock, in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Delay and predict equalization for blind speech dereverberation, vol. 5, (Toulouse, France, 2006), pp. 97–100
6. M Delcroix, T Hikichi, M Miyoshi, Precise dereverberation using multichannel linear prediction. IEEE Trans. Audio, Speech, Lang. Process. **15**(2), 430–440 (2006)
7. S Subramaniam, A Petropulu, C Wendt, Cepstrum-based deconvolution for speech dereverberation. IEEE Trans. Speech Audio Process. **4**(5), 392–396 (1996)
8. BDV Veen, KM Buckley, Beamforming: A versatile approach to spatial filtering. IEEE ASSP Mag. **5**(2), 4–24 (1988)
9. J Allen, D Berkley, Multimicrophone signal processing technique to remove room reverberation from speech signals. J. Acoust. Soc. Am. **62**, 912–915 (1977)
10. R Zelinski, in *Int. Conf. on Acoust. Speech and Sig. Proc.* A microphone array with adaptive post-filtering for noise reduction in reverberant rooms, (New York, USA, 1988), pp. 2578–2581
11. S Fischer, Beamforming microphone arrays for speech acquisition in noisy environments. Speech Commun. **20**, 215–227 (1996)
12. E Habets, J Benesty, I Cohen, S Gannot, J Dmochowski, New insights into MVDR beamformer in room acoustics. IEEE Trans. Audio, Speech Lang. Process. **18**(1), 158–170 (2010)
13. E Habets, J Benesty, A two stage beamforming approach for noise reduction and dereverberation. IEEE Trans. Audio, Speech Lang. Process. **21**(5), 945–958 (2013)
14. K Lebart, JM Boucher, PN Denbigh, A new method based on spectral subtraction for speech dereverberation. ACUSTICA. **87**(3), 359–366 (2001)
15. FS Pacheco, R Seara, in *Proc. of the Fifth International Telecommunications Symposium (ITS2006)*. Spectral subtraction for reverberation reduction applied to automatic speech recognition, vol. 4, (Fortaleza-CE, Brazil, 2006), pp. 581–584
16. T Yoshioka, MJ Gales, Environmentally robust asr front-end for deep neural network acoustic models. Comput. Speech Lang. **31**(1), 65–86 (2015)
17. L Deng, A Acero, M Plumpe, XD Huang, in *Proc. ICSLP '00*. Large-vocabulary speech recognition under adverse acoustic environment, (Beijing, China, 2000), pp. 806–809
18. X Xiao, J Li, ES Chng, H Li, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Feature compensation using linear combination of speaker and environment dependent correction vectors, (Florence, Italy, 2014), pp. 1720–1724
19. T Toda, AW Black, K Tokuda, Voice conversion based on maximum-likelihood estimation of spectral parameters trajectory. IEEE Trans. Audio, Speech, Lang. Process. **15**(8), 2222–2235 (2007)
20. EA Wan, AT Nelson, in *Handbook of neural networks for speech processing*, ed. by S Katagiri. Networks for speech enhancement (Artech House, Boston, 1998)
21. GE Hinton, S Osindero, Y Teh, A fast learning algorithm for deep belief nets. Neural Comput. **8**(7), 1527–1554 (2006)
22. Y Bengio. Foundations and Trends® in Machine Learning. Learning deep architectures for AI, vol. 2, (2009), pp. 1–127
23. GE Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, T Sainath, B Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. Signal Process. Mag. IEEE. **29**(6), 82–97 (2012)
24. AL Maas, QV Le, TM O'Neil, O Vinyals, P Nguyen, AY Ng, in *Interspeech 2012*. Recurrent neural networks for noise reduction in robust asr (Citeseer, Portland, Oregon, 2012)
25. F Weninger, J Geiger, M Wöllmer, B Schuller, G Rigoll, Feature enhancement by deep lstm networks for asr in reverberant multisource environments. Comput. Speech Lang. **28**(4), 888–902 (2014)
26. B Li, KC Sim, A spectral masking approach to noise-robust speech recognition using deep neural networks. IEEE/ACM Trans. Audio, Speech Lang. Process. (TASLP). **22**(8), 1296–1305 (2014)
27. Y Xu, J Du, L-R Dai, C-H Lee, A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio, Speech, Lang. Process. **23**(1), 7–19 (2015)

28. J Du, Q Wang, T Gao, Y Xu, L Dai, C-H Lee, in *Interspeech 2014*. Robust speech recognition with speech enhanced deep neural networks, (Singapore, 2014)
29. X Xiao, S Zhao, DHH Nguyen, X Zhong, DL Jones, ES Chng, H Li, in *Proceeding of REVERB challenge workshop*. The NTU-ADSC systems for reverberation challenge, (Florence, Italy, 2014)
30. MJF Gales, Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang. **12**, 75–98 (1998)
31. DHH Nguyen, X Xiao, ES Chng, H Li, in *ICASSP 2014*. Generalization of temporal filter and linear transformation for robust speech recognition, (Florence, Italy, 2014)
32. H Kuttruff, *Room acoustics*, 4th edn. (Taylor & Francis, 270 Madison Avenue, New York, NY, 2000)
33. CH Knapp, GC Carter, The generalized correlation method for estimation of time delay. IEEE Trans. Acoust., Speech Signal Process. **24**(4), 320–327 (1976)
34. OLF III, An algorithm for linearly constrained adaptive array process. IEEE Proc. **60**(8), 926–935 (1972)
35. HW Löllmann, E Yilmaz, M Jeub, P Vary, in *International Workshop on Acoustic Echo and Noise Control (IWAENC)*. An improved algorithm for blind reverberation time estimation, (Tel Aviv, Israel, 2010)
36. S Furui, Speaker independent isolated word recognizer using dynamic features of speech spectrum. IEEE Trans. Acoustics, Speech Signal Process. **34**(1), 52–59 (1986)
37. JL Gauvain, CH Lee, Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Trans. Speech Audio Process. **2**(2), 291–298 (1994)
38. PJ Moreno, *Speech recognition in noisy environments. PhD thesis*. (ECE, Carnegie Mellon University, 1996)
39. A Acero, L Deng, T Kristjansson, J Zhang, in *Proc. ICSLP '00*. HMM adaptation using vector Taylor series for noisy speech recognition, (Beijing, China, 2000), pp. 869–872
40. J Li, L Deng, D Yu, Y Gong, A Acero, A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions. Comput. Speech Lang. **23**(3), 389–405 (2009)
41. Y Li, H Erdogan, Y Gao, E Marcheret, in *Proc. ICSLP '02*. Incremental on-line feature space MLLR adaptation for telephony speech recognition, (Denver, USA, 2002), pp. 1417–1420
42. H Hermansky, N Morgan, RASTA processing of speech. IEEE Trans. Speech Audio Process. **2**(4), 578–589 (1994)
43. C-P Chen, JA Bilmes, MVA processing of speech features. IEEE Trans. Audio, Speech, Lang. Process. **15**(1), 257–270 (2007)
44. X Xiao, ES Chng, H Li, Normalization of the speech modulation spectra for robust speech recognition. IEEE Trans. Audio, Speech, Lang. Process. **16**(8), 1662–1674 (2008)
45. X Xiao, ES Chng, H Li, in *Proc. ICASSP '13*. Temporal filter design by minimum KL divergence criterion for robust speech recognition (Vancouver, Canada, 2013)
46. K Kinoshita, M Delcroix, T Yoshioka, T Nakatani, E Habets, R Haeb-Umbach, V Leutnant, A Sehr, W Kellermann, R Maas, S Gannot, B Raj, in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA-13)*. The REVERB challenge: a common evaluation framework for dereverberation and recognition of reverberant speech, (New Paltz, NY, 2013)
47. T Robinson, J Fransen, D Pye, J Foote, S Renals, in *Proc. ICASSP '95*. WSJCAM0: a British English speech corpus for large vocabulary continuous speech recognition, (Detroit, MI, 1995), pp. 81–84
48. DB Paul, JM Baker, in *Proceedings of the Workshop on Speech and Natural Language (HLT-91)*. The design for the wall street journal-based csr corpus, (Stroudsburg, PA, 1992), pp. 357–362
49. M Lincoln, I McCowan, J Vepa, HK Maganti, in *Proc. ASRU '05*. The multi-channel wall street journal audio visual corpus (MC-WSJ-AV): specification and initial experiments, (Cancun, Mexico, 2005), pp. 357–362
50. Y Hu, P Loizou, Evaluation of objective quality measures for speech enhancement. IEEE Trans. Audio, Speech, Lang. Process. **16**(1), 229–238 (2008)
51. TH Falk, C Zheng, W-Y Chan, A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech. IEEE Trans. Audio, Speech, Lang. Process. **18**(7), 1766–1774 (2010)
52. A Rix, M Hollier, A Hekstra, JG 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. **50**(10), 755–764 (2002)
53. D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlicek, Y Qian, P Schwarz, J Silovsky, G Stemmer, K Vesely, in *Proc. ASRU '11*. The kaldi speech recognition toolkit, (Waikoloa, HI, 2011)