Xiong *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:70
DOI 10.1186/s13634-015-0256-4

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Front-end technologies for robust ASR in reverberant environments—spectral enhancement-based dereverberation and auditory modulation filterbank features

Feifei Xiong[1,3*], Bernd T. Meyer[2,3], Niko Moritz[1,3], Robert Rehr[2,3], Jörn Anemüller[2,3], Timo Gerkmann[2,3], Simon Doclo[1,2,3] and Stefan Goetze[1,3]

Abstract

This paper presents extended techniques aiming at the improvement of automatic speech recognition (ASR) in single-channel scenarios in the context of the REVERB (REverberant Voice Enhancement and Recognition Benchmark) challenge. The focus is laid on the development and analysis of ASR front-end technologies covering speech enhancement and feature extraction. Speech enhancement is performed using a joint noise reduction and dereverberation system in the spectral domain based on estimates of the noise and late reverberation power spectral densities (PSDs). To obtain reliable estimates of the PSDs—even in acoustic conditions with positive direct-to-reverberation energy ratios (DRRs)—we adopt the statistical model of the room impulse response explicitly incorporating DRRs, as well in combination with a novel proposed joint estimator for the reverberation time T_{60} and the DRR. The feature extraction approach is inspired by processing strategies of the auditory system, where an amplitude modulation filterbank is applied to extract the temporal modulation information. These techniques were shown to improve the REVERB baseline in our previous work. Here, we investigate if similar improvements are obtained when using a state-of-the-art ASR framework, and to what extent the results depend on the specific architecture of the back-end. Apart from conventional Gaussian mixture model (GMM)-hidden Markov model (HMM) back-ends, we consider subspace GMM (SGMM)-HMMs as well as deep neural networks in a hybrid system. The speech enhancement algorithm is found to be helpful in almost all conditions, with the exception of deep learning systems in matched training-test conditions. The auditory feature type improves the baseline for all system architectures. The relative word error rate reduction achieved by combining our front-end techniques with current back-ends is 52.7% on average with the REVERB evaluation test set compared to our original REVERB result.

Keywords: Automatic speech recognition; Dereverberation; Auditory modulation filterbank; Deep neural network; REVERB challenge

*Correspondence: feifei.xiong@idmt.fraunhofer.de
[1] Fraunhofer Institute for Digital Media Technology IDMT, Project Group Hearing, Speech and Audio Technology (HSA), Oldenburg, Germany
[3] University of Oldenburg, Cluster of Excellence Hearing4All, Oldenburg, Germany
Full list of author information is available at the end of the article

1 Introduction

Improving the performance of automatic speech recognition (ASR) systems in reverberant environments is still a major challenge in the signal enhancement and machine learning communities [1, 2]. Strategies that aim to alleviate the influence of the reverberation effect range from dereverberation techniques in audio processing [3–5] over robust feature extraction methods [6] to reverberant signal modeling in ASR [7]. In order to provide a common evaluation framework for developing and testing of algorithms in the fields of dereverberation as well as reverberation-robust ASR, the REverberant Voice Enhancement and Recognition Benchmark (REVERB) challenge [8] has been launched and REVERB contributions showed significant improvements for speech enhancement (cf., e.g., [9]) and ASR (cf., e.g., [10, 11]).

Our previous contribution to the REVERB challenge [12] proposed a combined system including speech enhancement, robust feature extraction, acoustic model adaptation, posterior decoding, and word hypothesis fusion of multiple ASR systems for the REVERB single-channel (*1ch*) ASR task. Compared to the REVERB challenge baseline results of the final evaluation test set, an absolute improvement of average word error rate (WER) of 12.43 % in the *utterance-based batch processing* mode and of 9.42 % in the *full batch processing* mode were achieved in [12]. For the single-channel scenario, the submitted system in [12] showed the best performance amongst all the submitted results which solely used the ASR back-end system based on the hidden Markov model toolkit (HTK) [13] that was provided as baseline system by the REVERB challenge. It should of course be noted, that by far better results were obtained with more advanced ASR back-end technologies, e.g., feature transformation/adaptation from the Kaldi toolkit [14] in [10], or deep neural networks (DNNs) in [11]. In general, a gap of 50 % relative difference w.r.t. WERs exists between the results using the provided baseline ASR back-end recognizer of the REVERB challenge and those using more advanced back-end recognizers. For instance, we achieved an average WER of 42.12 % in [12] with the real recording data in the *utterance-based batch processing* mode, while the best challenge result under this processing mode was 20.30 % by [11]. Such a significant boost also motivates the extensions of our work in this contribution by combining our front-end technologies with state-of-the-art ASR back-end strategies such as using DNNs to generate bottleneck (BN) features [15], and subspace Gaussian mixture models (SGMMs) [16], as well as DNN-based acoustic modeling [17]. Our proposed front-end is composed of two components. One is the speech enhancement system aiming at suppressing the interference signal components, i.e., the noise and late reverberation which significantly degrade ASR performance [3, 18]. The other component is the extraction of robust features [6] in adverse environments, which are based on findings in the auditory processing of mammals.

We pre-process the noisy and reverberant signal using a single-channel speech enhancement scheme before the recognition takes place. It has been shown that the side-effects brought by speech enhancement such as musical noise and speech distortions are also detrimental to ASR systems [19, 20]. Therefore, a clean speech estimator is required which keeps the speech distortions at a low level and introduces a minimal amount of artifacts like musical noise. The minimum mean square error (MMSE) estimator of the clean speech amplitudes proposed in [21] is a parametric estimator that can be tuned in such a way that it gives a good compromise between musical noise, speech distortions, and speech enhancement. For this, reliable estimates of the corresponding desired speech and interference power spectral densities (PSDs) are required. Here, for the considered reverberant scenarios with stationary noises present in the REVERB challenge data, the minimum statistics (MS) approach [22] is employed to estimate the noise PSD. However, the temporal smearing caused by the reverberation leads to more minima to be affected by the reverberant speech energy. In order to be sure that the tracked minima only belong to the noise energy, the MS search window length is increased for the proposed system. After estimating the reverberant speech PSD and applying temporal cepstrum smoothing (TCS) [23] which is capable of reducing the effect of detrimental musical noise to ASR [19], the late reverberation PSD is obtained from the estimate of the reverberant speech PSD based on the approach of [24]. However, Polack's statistical model of the room impulse response (RIR) used in [24] only considers scenarios for which the speaker-microphone distance is larger than the critical distance, i.e., the direct-to-reverberation energy ratio (DRR) is smaller than 0 dB [25]. In order to also cover reverberant situations for which the speaker-microphone distance is smaller than the critical distance, i.e., for positive DRRs, the late reverberation PSD estimator by [26] is adopted that considers the direct sound separately. In contrast to [24] where only the reverberation time T_{60} is needed for Polack's RIR model, the method proposed by [26] additionally requires an estimate of the DRR. Thus, a novel estimator is proposed based on our previous work in [27, 28] using a multi-layer perceptron (MLP) to jointly estimate T_{60} and DRR. With these quantities, the enhanced speech PSD is estimated and the time domain enhanced speech signal is used for the feature extraction stage of the ASR systems.

Robust feature extraction in this work is achieved based on auditory processing. It is obtained based on an amplitude modulation filterbank (AMFB) [6, 29], which is inspired by the finding that the processing of amplitude fluctuations plays an important role for speech intelligibility. In [30], a periodotopic arrangement of neurons tuned to certain modulation frequencies in the inferior colliculus was observed to be almost orthogonal to the tonotopic arrangement of neurons tuned to certain acoustic frequencies. It has also been shown in [31] that the human auditory system decomposes an audio signal not only into its acoustic frequencies but also into its amplitude modulation frequency components. As well, it is known that a wider temporal context is essential for human understanding and automatic speech recognition [29, 32, 33]. The AMFB features are capable of capturing temporal dynamic information, while the conventional mel-frequency cepstral coefficients (MFCCs) [34] only extract information from a relatively limited temporal context. The AMFB features have been shown to be effective in reverberant acoustical environments earlier compared to MFCCs in [12], particularly under far-field conditions in larger reverberant rooms.

The remainder of this paper is organized as follows: Section 2 describes the speech enhancement algorithm. The auditory modulation filterbank features, i.e., AMFB features, as well as BN features are introduced in Section 3. The acoustic models for ASR used in this contribution are briefly described in Section 4. The experimental procedure and results of the proposed system are presented and analyzed in Section 5. Conclusions are given in Section 6.

2 Speech enhancement (SE)

Single-channel speech enhancement (SE) is used to dereverberate, as well as to de-noise the speech signal. Here, the MMSE estimator described in [21] is used to obtain the clean speech magnitude, which requires the PSDs of the speech signal and the interference. It has been shown, e.g., in [3, 18], that attenuating late reverberation is crucial for ASR while early reflections can be mitigated well by, e.g., cepstral mean subtraction (CMS) [35]. Hence, the interference signal is considered to contain late reverberation and the noise, while the desired speech signal is formed by the direct path and some early reflections.

2.1 Structure of the SE algorithm

The recorded microphone signal $y[k]$ consists of the reverberant speech signal $x[k]$ and the noise $n[k]$,

$$y[k] = x[k] + n[k] \ , \tag{1}$$

with k denoting the discrete time index. The reverberant speech signal $x[k]$ can be modeled as the convolution of the clean (anechoic) speech signal $s[k]$ and the RIR $h[k]$. The reverberant signal $x[k]$ can be split into the clean speech signal $s[k]$ and the residual reverberation $x_\mathrm{r}[k]$ as

$$\begin{aligned} x[k] &= s[k] * h[k] = \sum_{k'=0}^{\infty} s[k-k'] \cdot h[k'] \\ &= s[k] + \underbrace{\sum_{k'=1}^{\infty} s[k-k'] \cdot h[k']}_{=x_\mathrm{r}[k]}, \end{aligned} \tag{2}$$

where $*$ denotes the convolution. Furthermore, $x_\mathrm{r}[k]$ can be decomposed into early reflections $x_\mathrm{e}[k]$ and late reverberation $x_\mathrm{l}[k]$ separated at sample K_e (usually up to about 50 ms of $h[k]$ [25], i.e.,$K_\mathrm{e} = \lfloor f_s \cdot 50\mathrm{ms} \rfloor$ at a sampling frequency f_s) as

$$x_\mathrm{r}[k] = \underbrace{\sum_{k'=1}^{K_\mathrm{e}} s\left[k-k'\right] \cdot h\left[k'\right]}_{=x_\mathrm{e}[k]} + \underbrace{\sum_{k'=K_\mathrm{e}+1}^{\infty} s\left[k-k'\right] \cdot h\left[k'\right]}_{=x_\mathrm{l}[k]} . \tag{3}$$

With (2) and (3), (1) can be rewritten as,

$$y[k] = s[k] + x_\mathrm{r}[k] + n[k] = \underbrace{s[k] + x_\mathrm{e}[k]}_{=y_\mathrm{d}[k]} + \underbrace{x_\mathrm{l}[k] + n[k]}_{=y_\mathrm{i}[k]}, \tag{4}$$

with $y_\mathrm{d}[k]$ being the desired signal part which contains the direct-path signal $s[k]$ and early reflections $x_\mathrm{e}[k]$. The interference component $y_\mathrm{i}[k]$ consists of late reverberation $x_\mathrm{l}[k]$ and the noise $n[k]$. Note that the noise $n[k]$ considered here is mostly stationary and was recorded under the same reverberation conditions as the RIR measurement [8]. The goal is to find an estimate of the desired speech component $\hat{y}_\mathrm{d}[k]$ and, by this, to reduce the interference $y_\mathrm{i}[k]$.

The proposed speech enhancement algorithm depicted in Fig. 1 operates in the short-time Fourier transform (STFT) domain. Uppercase variables, e.g., $Y[m,\ell]$, $X_\mathrm{l}[m,\ell]$, and $Y_\mathrm{d}[m,\ell]$, denote the STFT representations of $y[k]$, $x_\mathrm{l}[k]$, and $y_\mathrm{d}[k]$, respectively, with m and ℓ representing the frequency bin and the temporal frame index. First, we need to estimate the background noise (cf. Section 2.2, please note that no reverberation is included in the noise PSD estimate) to obtain an estimate of the reverberant speech from TCS (cf. Section 2.3). This PSD $\hat{\lambda}_X[m,\ell]$ is then used to obtain an estimate of late reverberation based on the RIR model from [26] (cf. Section 2.4), which allows for estimating the clean speech PSD $\hat{\lambda}_{Y_\mathrm{d}}[m,\ell]$ by applying TCS again. Then, the MMSE estimator (cf. Section 2.5) is applied to obtain an estimate of the clean speech signal $\hat{Y}_\mathrm{d}[m,\ell]$, which is then transferred back to the time domain by

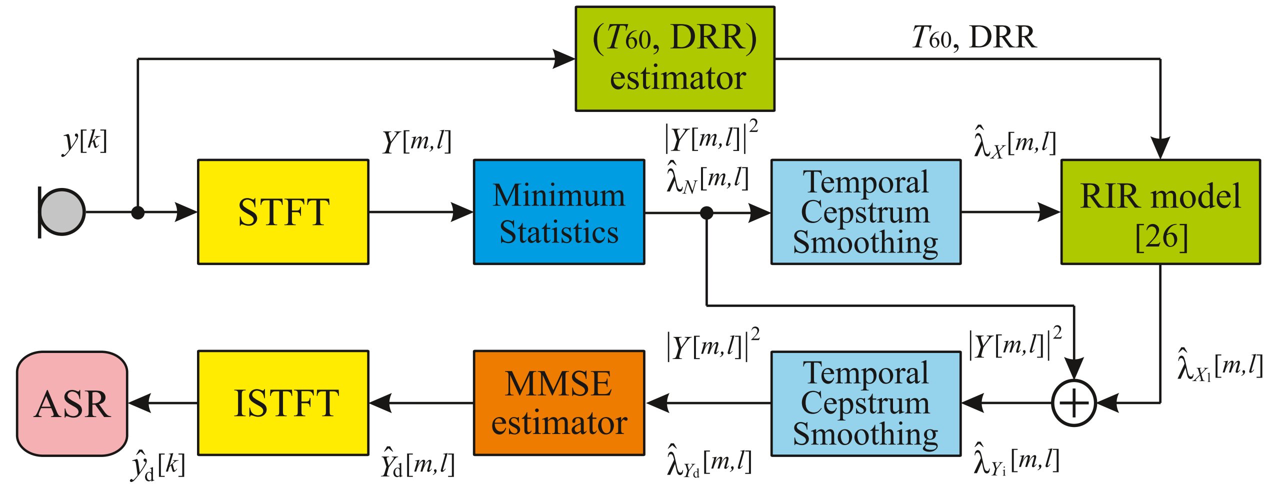

Fig. 1 Overview of the speech enhancement (SE) algorithm to suppress the noise and late reverberation

inverse STFT (ISTFT) and subsequently forwarded to the ASR system. Additionally, a joint (T_{60}, DRR) estimator using an MLP network (cf. Section 2.6) is proposed as input for the late reverberation PSD estimate $\hat{\lambda}_{X_l}[m,\ell]$.

2.2 Noise PSD estimation

The noise PSD $\lambda_N[m,\ell]$ is estimated using an adapted version of the well-known MS method [22]. MS offers accurate noise PSD estimation especially if the noise signal is stationary to a certain extent, i.e., varies slowly compared to the statistics of the desired speech component, which is true for the noise contained in the REVERB challenge data set. It is assumed that the minima in this PSD originate from time-frequency bins that do not contain speech. These minima are tracked using a search window spanning usually 1.5 s of the estimated input PSDs in [22]. However, to ensure that no reverberation (which introduces the decay tail to the speech pauses) leaks into the noise PSD estimate, we enlarged this search window to 3 s [36].

2.3 Speech PSD estimation

We employ temporal cepstrum smoothing (TCS) [23, 37] for estimating the reverberant and the clean speech PSDs. This approach smoothes the maximum likelihood (ML) estimate of the clean or reverberant speech PSD over time in the cepstral domain. Due to the compact representation of speech in the cepstral domain, speech-related and non-speech-related coefficients can be selectively smoothed. Compared to other approaches, e.g., the decision-directed approach [38], TCS is able to reduce musical noise artifacts, which is crucial for ASR systems [19].

First, the PSD of the reverberant speech $\lambda_X[m,\ell] = \mathrm{E}\left\{|X[m,\ell]|^2\right\}$ ($\mathrm{E}\{\cdot\}$ is the expectation operator) is obtained by the ML estimate [38],

$$\hat{\lambda}_X^{\text{ml}}[m,\ell] = \max\left(|Y[m,\ell]|^2 \hat{\lambda}_N[m,\ell], \, \xi_{\min}\cdot\hat{\lambda}_N[m,\ell]\right), \quad (5)$$

where $\xi_{\min}$ is the lower bound of the *a priori* signal-to-noise-ratio. Then, the cepstral representation of the above ML estimate is calculated as

$$\hat{\lambda}_X^{c,\text{ml}}[q,\ell] = \mathcal{F}^{-1}\left\{\ln\left(\hat{\lambda}_X^{\text{ml}}[m,\ell]\right)\right\}, \quad (6)$$

where the superscript ^{c} denotes the cepstral domain and q represents the cepstral or quefrency index. $\mathcal{F}^{-1}\{\cdot\}$ denotes the inverse discrete Fourier transform (IDFT). After that, smoothing is applied to (6), i.e.,

$$\hat{\lambda}_X^{c}[q,\ell] = \alpha^c[q,\ell]\cdot\hat{\lambda}_X^{c}[q,\ell-1] + (1-\alpha^c[q,\ell])\cdot\hat{\lambda}_X^{c,\text{ml}}[q,\ell], \quad (7)$$

where $\alpha^c[q,\ell]$ represents a quefrency-dependent smoothing coefficient, which should be chosen such that the coefficients relevant for speech production are maintained while the remaining coefficients are strongly smoothed [23]. Thus, in an SE framework, usually $\alpha^c[q,\ell]$ is chosen small for the speech spectral envelope represented by the low quefrencies and the fundamental period peak in the cepstrum [37]. In contrast to SE, preserving the fundamental frequency is not crucial for ASR systems [39], and $\alpha^c[q,\ell]$, thus, is chosen as,

$$\alpha^c[q,\ell] = \begin{cases} 0.0 \text{ for } q = 0,\ldots,\lceil f_s\cdot 0.5\text{ ms}\rceil - 1, \\ 0.5 \text{ for } q = \lceil f_s\cdot 0.5\text{ ms}\rceil,\ldots,\lceil f_s\cdot 1\text{ ms}\rceil - 1, \\ 0.9 \text{ for otherwise.} \end{cases} \quad (8)$$

Note that the application range of q is only given for the lower half of the cepstrum, which will be applied accordingly to the symmetric counterpart. Finally, the reverberant speech PSD estimate $\hat{\lambda}_X[m,\ell]$ is achieved after transforming (7) back to the frequency domain,

$$\hat{\lambda}_X[m,\ell] = b \cdot \exp\left(\mathcal{F}\left\{\hat{\lambda}_X^c[q,\ell]\right\}\right), \tag{9}$$

where $\mathcal{F}\{\cdot\}$ represents the DFT operator. The factor b is a function of the smoothing factor $\alpha^c[q,\ell]$ in (8) and compensates for the bias caused by the averaging in the cepstral domain [23]. For a detailed discussion of b the reader is referred to [23].

2.4 Late reverberation PSD estimation

In our workshop paper [12], Polack's statistical RIR model [3, 24] has been used to achieve reverberation suppression based on an estimate of the late reverberation PSD $\lambda_{X_l}[m,\ell] = \mathrm{E}\left\{|X_l[m,\ell]|^2\right\}$. Please note, that for simplicity, the frequency bin m will be omitted in the following descriptions within Section 2, since the spectral bins are assumed to be independent. By using the reverberant speech PSD estimate $\hat{\lambda}_X[\ell]$ obtained from (9), as well as the separation of the early and late part in (2)-(4), the late reverberation PSD estimate can be calculated by [24]

$$\hat{\lambda}_{X_l}[\ell] = \exp(-2\rho\tau_s L_e) \cdot \hat{\lambda}_X[\ell - L_e], \tag{10}$$

where the parameter L_e is a number of frames which corresponds to the duration of early part of the RIR (cf. K_e in samples in (3)). Consequently, $L_e \cdot \tau_s$ is the start time of late reverberation (which is fixed to 50 ms here), and τ_s is the STFT time shift (hop size in s). ρ is the decay rate related to the reverberation time T_{60}, i.e., $\rho = 3\ln(10)/T_{60}$. In [12], blind reverberation time T_{60} estimation was achieved by the method proposed in [40], which is based on spectral decay distributions of the observed speech signal and is shown to be robust against additive noise when a noise PSD estimator is appended.

Considering reverberant situations where the speaker-microphone distances are smaller than the critical distance, i.e., those with positive DRRs [25], the statistic reverberation model proposed in [26] is used here which separates the direct path from Polack's RIR model as used in [24], defined with the spectral variance $\lambda_H[\ell]$ of the RIR $h[k]$ in the STFT domain as

$$\lambda_H[\ell] = \begin{cases} \beta_d & \text{for } \ell = 0, \\ \beta_r \exp(-2\rho\tau_s\ell) & \text{for } \ell \geq 1, \end{cases} \tag{11}$$

where β_d and β_r denote the variances of the direct path and the residual reverberation part, respectively. Accordingly, the relationship to the DRR is given by [26]

$$\mathrm{DRR} = 10\log_{10}\left(\frac{1-\exp(-2\rho\tau_s)}{\exp(-2\rho\tau_s)} \cdot \frac{\beta_d}{\beta_r}\right). \tag{12}$$

Using (11), the reverberant speech PSD can be computed by [26]

$$\begin{aligned}\hat{\lambda}_{X_r}[\ell] = {} & (1-\kappa) \cdot \exp(-2\rho\tau_s)\hat{\lambda}_{X_r}[\ell-1] \\ & + \kappa \cdot \exp(-2\rho\tau_s)\hat{\lambda}_X[\ell-1],\end{aligned} \tag{13}$$

where $\kappa = \beta_r/\beta_d$ is calculated from the DRR in (12), constraint in the range of (0, 1]. Then, the late reverberation PSD from (10) is modified to

$$\hat{\lambda}_{X_l}[\ell] = \exp(-2\rho\tau_s(L_e-1)) \cdot \hat{\lambda}_{X_r}[\ell - L_e + 1]. \tag{14}$$

If κ equals 1, then (14) is equivalent to (10), which shows that this approach is the same as the approach described in [24] under this condition. It has been shown in [41] that (14) provides a more reliable late reverberation PSD estimate so that less speech distortions are achieved, which is a benefit for the ASR system. A disadvantage of this method is that it requires not only T_{60} but also the DRR. In other words, the reliable estimation of these two parameters plays a crucial role for the late reverberation PSD estimate, which here can be obtained by an MLP estimator described in Section 2.6.

2.5 MMSE estimator

After estimating the noise and late reverberation PSDs, the interference PSD can be obtained from (4) in a straightforward way as

$$\hat{\lambda}_{Y_i}[\ell] = \hat{\lambda}_{X_l}[\ell] + \hat{\lambda}_N[\ell], \tag{15}$$

assuming that the late reverberant signal $x_l[k]$ and the noise $n[k]$ are uncorrelated. $\hat{\lambda}_{Y_i}[\ell]$ will be used to estimate the PSD of the desired speech component $Y_d[\ell]$ by another TCS procedure as depicted in the lower branch of Fig. 1. To achieve this, as aforementioned in Section 2.3, the input noise PSD $\hat{\lambda}_N[\ell]$ in (5)-(9) is replaced by $\hat{\lambda}_{Y_i}[\ell]$. As we now use the PSD of the interference signal $\hat{\lambda}_{Y_i}[\ell]$, i.e., the noise and late reverberation, TCS will estimate the PSD of the clean speech signal and early reflections $\hat{\lambda}_{Y_d}[\ell]$.

In the final step, a parameterized MMSE spectral magnitude estimator [21] is used to determine the weighting function $G[\ell]$ to obtain the enhanced speech signal $\hat{Y}_d[\ell]$. A simplified, computationally less complex version [42] based on the confluent hypergeometric function [43] is used, which is defined as

$$\begin{aligned}G[\ell] = {} & \left(\frac{1}{1+\nu[\ell]}\right)^{p_0} \cdot G_0[\ell] \\ & + \left(\frac{\nu[\ell]}{1+\nu[\ell]}\right)^{p_\infty} \cdot \frac{\hat{\xi}[\ell]}{\mu + \hat{\xi}[\ell]}\end{aligned} \tag{16}$$

$$G_0[\ell] = \left(\frac{\Gamma(\mu+\gamma/2)}{\Gamma(\mu)}\right)^{1/\gamma} \cdot \left(\frac{\hat{\xi}[\ell]}{\mu+\hat{\xi}[\ell]} \cdot \frac{1}{\hat{\zeta}[\ell]}\right)^{1/2}, \tag{17}$$

$$\nu[\ell] = \frac{\hat{\xi}[\ell]}{\mu + \hat{\xi}[\ell]} \cdot \hat{\zeta}[\ell], \tag{18}$$

with $\Gamma(\cdot)$ being the complete gamma function. The estimates of the *a priori* and *a posteriori* desired-signal-to-interference-ratios are defined as $\hat{\xi}[\ell] = \hat{\lambda}_{Y_\text{d}}[\ell] / \hat{\lambda}_{Y_\text{i}}[\ell]$, and $\hat{\zeta}[\ell] = |Y[\ell]|^2 / \hat{\lambda}_{Y_\text{i}}[\ell]$, respectively. The constant parameters μ and γ can be tuned to yield several types of estimators. In [21], $\mu = 0.5$ and $\gamma = 0.5$ have been identified as a good compromise between the amount of musical noise and the clarity of speech and are, therefore, also applied here. For obtaining the correct approximation for the selected values of μ and γ, the exponents p_0 and p_∞ in (16) have to be set to 0.5 and 1.0, respectively [42]. Subsequently, the estimated desired signal $\hat{Y}_\text{d}[\ell]$ is calculated by

$$\hat{Y}_\text{d}[\ell] = \max(G[\ell], G_\text{min}) \cdot Y[\ell] \,, \tag{19}$$

with G_min being a lower bound for the weighting function $G[\ell]$ which alleviates speech distortions, however, also limits the possible amount of interference suppression. In conformance with [44], $G_\text{min} = -10$ dB is chosen as a good value to improve the ASR performance in reverberant environments. Then, as illustrated in Fig. 1, an ISTFT is conducted to reconstruct the output speech signal in the time domain $\hat{y}_\text{d}[k]$ used for the subsequent ASR experiments.

2.6 Estimation of room parameters (T_{60}, DRR)

A novel approach to jointly estimate T_{60} and DRR is proposed here based on our previous work [27, 28]. An overview of the estimation process is presented in Fig. 2: In a first step, reverberant signals are converted to spectro-temporal Gabor filterbank (GBFB) features [33, 45, 46] to capture information relevant for room parameter estimation. For details on GBFB selection, the reader is referred to [27]. A multi-layer perceptron (MLP) classifier, belonging to the class of feedforward artificial neural network models [47], is trained to map the input pattern to pairs of (T_{60}, DRR) values. Since the MLP generates one estimate per time step, we obtain an *utterance*-based estimate by simple temporal averaging and subsequent selection of the output neuron with the highest average activation (*winner-takes-all* approach). The MLP was implemented with the freely available QuickNet package [48] and has three layers. The output layer corresponds to the (T_{60}, DRR) pairs.

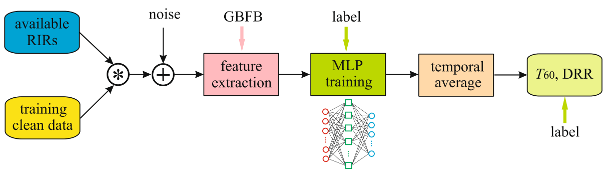

Fig. 2 Overview of the MLP setup for (T_{60}, DRR) estimation

These pairs were defined based on the RIRs provided by the training data of the REVERB challenge. Figure 3 shows the distribution of (T_{60}, DRR) values for the given RIRs. The bounding boxes in the figure denote the categorical boundaries for the classes. We defined 28 classes as a compromise between a large number of classes (with the potential of more accurate (T_{60}, DRR) classification, but only few training examples for each class) and few classes (with coarse classification, but many training examples).

The T_{60} values are obtained using Schroeder's method [49], which formulates a poly-fit in the range between −35 to −5 dB of the RIR accumulation energy. The DRR in dB is calculated as

$$\text{DRR} = 10 \log_{10} \frac{\sum_{k=0}^{K_\text{d}} |h[k]|^2}{\sum_{k=K_\text{d}+1}^{\infty} |h[k]|^2} \,, \tag{20}$$

where K_d represents the sample length of the direct sound arrival, which is usually measured as a short time period after the onset of the RIR. Here we take the maximum value of $h[k]$ as the onset of the RIR, and the following 0.5 ms range as the direct path samples, i.e., $K_\text{d} = \lfloor f_s \cdot 0.5 \text{ ms} \rfloor$.

3 Auditory modulation filterbank features

Baseline features of the REVERB challenge are MFCCs [34] plus delta (D) and double-delta (DD) coefficients combined with CMS. To improve robustness towards channel mismatch and quasi-stationary interference, we apply mean variance normalization (MVN) [35] instead of CMS to MFCC-D-DD and other feature types. This

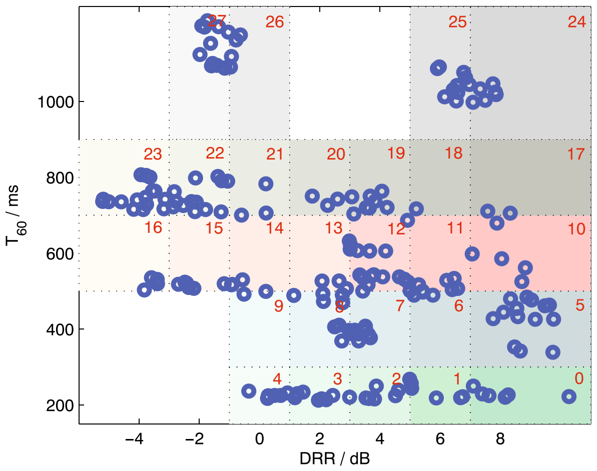

Fig. 3 Distribution of (T_{60}, DRR) values for the RIRs provided by the REVERB challenge (for ASR multi-condition training). Each *dot* represents one single-channel RIR with its (T_{60}, DRR). Twenty-eight classes are defined based on this distribution (*labels 0 to 27 inside the class boundary boxes*)

section introduces auditory modulation filterbank features which are based on an amplitude modulation filterbank (AMFB), as well as the bottleneck (BN) features concept which can be derived from different feature types.

3.1 Amplitude modulation filterbank (AMFB) features

The AMFB is employed for ASR feature extraction that analyzes temporal dynamics of a short-term spectro-temporal representation [6]. Please note that the AMFB employed in this study is based on an implementation proposed in [29] but without adjusting the distance between modulation filters.

$$a_m[q] = s_{\mathrm{carr}}[q] \cdot h_{\mathrm{env}}[q] \,, \tag{21}$$

$$s_{\mathrm{carr}}[q] = \exp\left(i\omega(q-q_0)\right), \tag{22}$$

$$h_{\mathrm{env}}[q] = 0.5 - 0.5 \cdot \cos\left(\frac{2\pi(q-q_0)}{W_q+1}\right). \tag{23}$$

The amplitude modulation filters a_m are complex exponential functions that are modulated by a Hann-envelope, as described in (21) by the Hadamard product of s_{carr} (cf. (22)) and h_{env} (cf. (23)), in which i is the imaginary unit and W_q is the Hann-envelope window length with the center index q_0 in the cepstral domain. Note that beyond the length W_q, the coefficients of Hann-envelope are set to zero in (23) as illustrated in the upper panel of Fig. 4.

The periodicity of the sinusoidal-carrier function is defined by the radian frequency ω. By varying ω and W_q, the AMFB can be tuned to cover different temporal amplitude modulation frequencies with different bandwidths. For the AMFB feature extraction, five amplitude modulation filters are selected, whose center frequencies and bandwidth settings are chosen according to the psycho-physically motivated amplitude modulation filterbank proposed in [32], which has a constant bandwidth of 5 Hz for amplitude modulation frequencies up to 10 Hz and a constant-Q relationship with value of 2 for higher modulation frequencies.

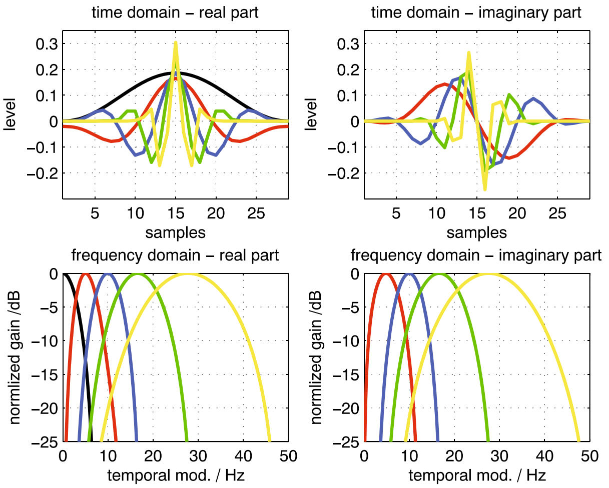

Fig. 4 The time domain filter coefficients (*upper panel*) and frequency domain normalized gain functions (*lower panel*) of the AMFB filters with real and imaginary parts; center temporal modulation frequencies are 0, 5, 10, 16.67, and 27.78 Hz

Figure 4 shows the AMFB coefficients in the time domain as well as their corresponding normalized gain functions in the frequency domain. Center modulation frequencies of the chosen five amplitude modulation filters are located at 0, 5, 10, 16.67, and 27.78 Hz, respectively, i.e., cover a much wider modulation frequency range compared to D and DD features that are located around 10 Hz.

3.2 Bottleneck (BN) features

BN features have been shown to be effective in improving the ASR performance [15, 50]. They are usually generated from a 4- or 5-layer MLP or DNN, in which one of the internal hidden layers has a small number of units compared to the sizes of other hidden layers or the output layer. As a result, such a small layer creates a constriction inside the network that forces the relative information into a low dimensional representation. Usually, the inputs to the hidden units of the BN layer will be used as features for the conventional HMM-based speech recognizer. Such BN features represent a nonlinear transformation of the original input features and represent the underlying speech quite well after the DNN is trained to show a good classification accuracy.

Another advantage of BN features is the dimension reduction functionality. For practical reasons, it is not feasible to pass very high dimensional features to conventional GMM-HMM systems. Other dimensionality reduction techniques such as principal component analysis or linear discriminant analysis (LDA) have to face the problem that the feature information in highly dimensional vectors might not be separable linearly. BN processing is particularly useful for our auditory modulation filterbank features which are characterized by more than 100 dimensions.

4 Advanced acoustic modeling

Stochastic processing with HMMs predominated acoustic modeling for ASR for nearly four decades [51], trained from data by, e.g., using the expectation maximization algorithm and incorporating GMMs [52] that efficiently represent the relationship between HMM states and the acoustic input. More recent approaches like SGMM [16] or DNN systems [17], however, lead to higher recognition performance especially in acoustically adverse conditions.

4.1 Subspace Gaussian mixture models (SGMMs)

The conventional GMM-HMM framework requires training of separate GMMs in each HMM state. SGMMs [16] allow HMM states to share a common structure, in which the means and mixture weights can vary within a subspace of the full parameter space. A global mapping from a vector space to the space of the GMM parameters is used, and the shared GMM is usually referred to as a universal background model (UBM). It has been shown [16] that SGMMs behave more compact and perform better than the conventional GMMs approach, as well, without the loss of compatibility to most standard techniques such as feature-space adaptation using maximum likelihood linear regression (fMLLR) [53], discriminative training like boosted maximum mutual information (bMMI) [54], or the minimum Bayes risk (MBR) approach [55].

4.2 Deep neural networks (DNNs)

DNNs have gained much attention during the last years because of the achieved dramatic improvements in acoustic modeling, especially with the recent breakthrough regarding a proper training of DNNs [17], which initializes the DNN weights to a suitable starting point rather than using a random initialization. This allows for using back-propagation training [56] and by this, a convergence to a better local optimum. The DNN-HMM approach has proven to be more effective in large vocabulary speech recognition tasks compared to the conventional GMM computation in HMMs (cf., e.g., [17, 57]), in which a DNN was trained to predict context-dependent posterior probabilities for the HMM states. Furthermore, the deep structure of DNNs allows for a more efficient representation of many nonlinear transformations due to many layers of simple nonlinear processing. This allows DNNs to learn more invariant and discriminative features [58]. Usually, such invariance improves the ASR robustness against the mismatch between the training and test data.

5 Experiments and results

Experiments and results shown in the following are all carried out according to the instructions of the REVERB challenge [8].

5.1 Database

The database provided by the REVERB challenge consists of simulated data (SimData) and real recordings (RealData) for different room sizes and speaker-microphone distances. Based on the WSJCAM0 corpus [59], SimData is artificially generated by convolving clean WSJCAM0 signals with measured RIRs, as well as by adding additional measured noises with a desired-signal-to-noise-ratio (DSNR) of approx. 20 dB. Six different acoustic conditions are simulated in the SimData, considering 3 different room sizes (Room1, 2, 3) with 2 different speaker-microphone distances (Near, Far). Furthermore, utterances from the MC-WSJ-AV corpus [60] were recorded in a room (Room1) with 2 different speaker-microphone distances (Near, Far) to generate the dataset RealData. The sampling frequency of the REVERB challenge data is 16 kHz.

To evaluate the ASR performance, a training set, a development test set (Dev.) and a final evaluation test set (Eval.) are provided. The SimData set consists of 1484 utterances from 10 speakers for Dev. and 2176 utterances from 28 speakers for Eval., respectively. The RealData set consists of 179 utterances from 5 speakers for Dev. and 372 utterances from 10 speakers for Eval., respectively. For a multi-condition training set with 7861 anechoic utterances from 92 speakers, 24 RIRs (cf. Fig. 3) and several types of stationary noise signals were recorded according to the 6 reverberant conditions mentioned above. Unlike for our workshop paper [12] that covered the *1ch* ASR task in both the *full batch processing* and the *utterance-based batch processing* mode, here we focus on the *1ch* scenarios only in the *utterance-based batch processing* mode for which each utterance is processed separately, since this provides the maximum potential for real-time applications.

5.2 ASR framework

The baseline ASR back-end recipe provided by the REVERB challenge is based on the HTK [13], with MFCC-D-DD features (dimension of 39) using CMS. For this paper, we use the open-source Kaldi ASR toolkit [14] to test various back-end technologies in combination with our front-end proposals. The text prompts of the utterances are based on the WSJ 5K corpus [61], and a bigram language model (LM) is generated. In order to further improve the LM accuracy, the standard 5K trigram LM is employed here. Multi-condition training is employed and WERs are used to assess the ASR performance. The average WERs of SimData and RealData are calculated as $\sum_{\text{set}}(\mathrm{W}_{\text{ER}} \cdot N_{\text{utt}})/\sum N_{\text{utt}}$, where W_{ER} represents WER of each test set with the corresponding amount of test utterances N_{utt}. The log-mel-spectrogram is calculated with frame length of 25 ms and frame shift of 10 ms.

5.3 ASR performance with speech enhancement

The proposed SE algorithm in Fig. 1 is applied to both, the multi-condition training set and the test sets. The parameter settings for the proposed algorithm described in Section 2 are summarized in Table 1.

5.3.1 Performance of (T_{60}, DRR) *Estimation*

As aforementioned in Section 2.6, 28 classes of the parameter pair (T_{60}, DRR) (cf. Fig. 3) which is needed as input for the SE algorithm have been defined based on the provided RIRs for training, which we assume, cover the necessary (T_{60}, DRR) parameter pairs of the testing set. The corresponding noises with DSNR of 20 dB for training

Table 1 Parameter settings for the speech enhancement (SE) algorithm described in Section 2

f_s	STFT block	$\xi_{\min}$ (5)	τ_s (10)	L_e (10)	$(\mu, \gamma, p_0, p_\infty)$ (16)-(18)	$G_{\min}$ (19)	K_d (20)
16 kHz	32 ms	−30 dB	16 ms	3	(0.5,0.5,0.5,1.0)	−10 dB	80

are added as shown in Fig. 2. The number of neurons in the input layer is 600, i.e. the dimension of the GBFB features [28]. The temporal context considered by the MLP is limited to 1 frame (i.e., no splicing is applied). The number of hidden units is 2048, while the number of output units is the amount of (T_{60}, DRR) classes to estimate.

As depicted in Fig. 5, the true values (blue/dark dots) of (T_{60}, DRR) are calculated according to the provided RIRs for SimData, which have been published after the REVERB challenge workshop and are only used here for analysis. For each utterance from the REVERB challenge Dev., one point for T_{60} is shown in the upper panel of Fig. 5 and the corresponding point for DRR in the lower panel, i.e., 1484 points for SimData and 179 for RealData. The true (T_{60}, DRR) pairs for RealData are not known (cf. missing blue/dark points in the right part of Fig. 5). The estimated (T_{60}, DRR) pair which is mapped from the output of the proposed MLP estimator (cf. Fig. 3) is shown in Fig. 5 by the green/light dots. In general, the provided RIRs for MLP training cover the (T_{60}, DRR) range of the test sets, except for one test set from 'Room2Far' of SimData, which consists of some RIRs with very low DRRs. As it can be seen in Fig. 5, most estimated (T_{60}, DRR) are close to the true values, although some deviations between blue/dark and green/light points can also be observed. Due to a too small amount of RIRs for training, 'Room2Far' of SimData shows larger deviations. Nevertheless, it seems that such deviation will influence the estimation of the corresponding reverberation effect less that it could be expected from Fig. 5. Regarding the estimated T_{60}, mostly overestimation can be observed, but at the same time, a trend of overestimation regarding DRRs occurs, which indicates that the reverberation effect with higher T_{60} and DRR at the same time behaves similar as with lower T_{60} and DRR. On the one hand, for higher T_{60}, more reverberation effect is perceived, but on the other hand, the higher the DRR is, the less the reverberation effect is. Such a phenomenon can be also observed in 'Rooms3Near', since the training RIRs with T_{60} around 700 ms and DRRs around 6 and 8 dB are quite rare as plotted in Fig. 3 (with labels 17 and 18), so that the MLP model for these sets may not be trained sufficiently.

Accordingly, most of the RealData utterances show higher T_{60} estimated (at 800 or 1000 ms), compared to the true values of approx. 700 ms [60] (given by the REVERB challenge). This might be explained by the fact that RealData shows larger mismatch with the multi-condition training data w.r.t. different noise types with lower (than 20 dB) DSNR, which may affect the MLP model as if more reverberation effect exists. Therefore, (T_{60}, DRR) estimates of RealData show higher T_{60} and lower DRRs, even for the 'Near' test set.

5.3.2 Performance of ASR with GMM-HMMs

Instead of applying D-DD to the MFCC features, an alternative feature post-processing from the literature [62]

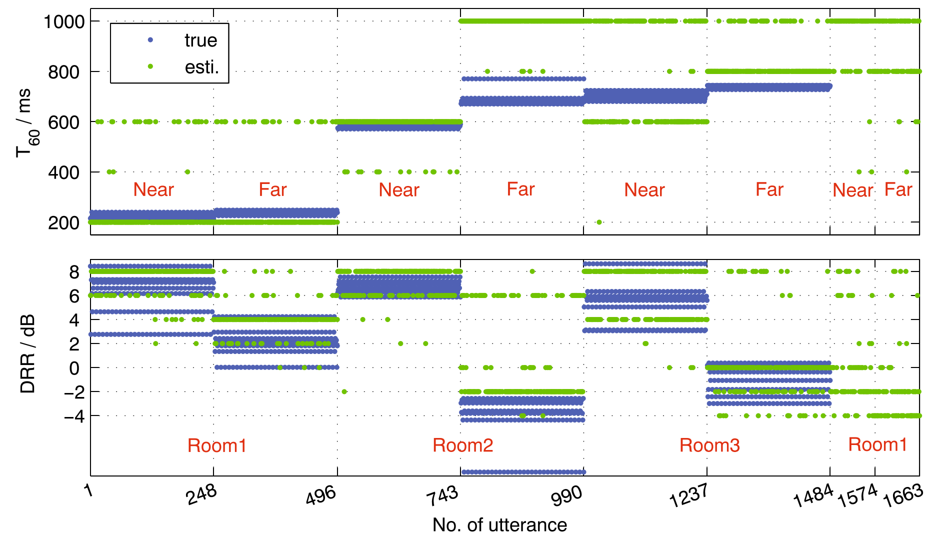

Fig. 5 Performance of the (T_{60}, DRR) estimator (cf. Section 2.6) for Dev. The true (T_{60}, DRR) of SimData are depicted for comparison. The acoustic conditions (room sizes and speaker-microphone distances) are according to Section 5.1

is used, for which the context frames are spliced and subsequently transformed via LDA and maximum likelihood linear transform (MLLT), which was shown to be more effective than MFCC-D-DD for feature extraction in reverberant conditions (average 5 % absolute WERs reduction) [10]. This also indicates that temporal dynamic information extracted by such spliced short-term spectral features is useful to improve the ASR performance in reverberant environments. Here, 9 consecutive frames (4 for left and 4 for right, i.e., $L = R = 4$) of 13 MFCCs were used, and the feature vector is projected to a 40-dimensional subspace, which were the optimal parameters found in [63].

Note that (T_{60}, DRR) are known for the multi-condition training data and the estimated (T_{60}, DRR) values from Section 5.3.1 are used for the test sets. For comparison, the previous SE algorithm used in [12] with only T_{60} estimate [40], and the proposed SE system but with true (T_{60}, DRR) (only for SimData) are processed using the same Kaldi-based ASR framework.

It can be observed from Fig. 6 that based on knowledge of T_{60} only, the proposed dereverberation algorithm (cf. (10)) does not increase the recognition performance in some reverberant scenarios by comparing the red line (with cross markers) to the blue line (with circular markers), e.g., 'Near' test sets of SimData whose DRRs are larger than 0 dB, as shown in Fig. 5. This phenomenon has also been noticed in our workshop paper [12] using HTK-based ASR framework. If DRRs are used together with T_{60} (cf. (13)-(14)) as shown by the green line (with square markers), all the test sets for all reverberant conditions benefit from the proposed SE algorithm. It just performs slightly worse than the ideal case for which the true (T_{60}, DRR) are given for SimData (yellow line with diamond markers), showing that the proposed (T_{60}, DRR) estimation via MLP is capable of providing sufficiently correct T_{60} and DRR information to the SE algorithm. In average, 2–3 % absolute WER improvement can be obtained by the proposed SE algorithm using the proposed (T_{60}, DRR) estimator.

5.4 ASR performance of auditory modulation filterbank features

AMFB features are calculated based on the log-mel-spectrogram with 31 dimensions and a subsequent discrete cosine transform along the spectral axis, i.e., the cepstrogram. The AMFB in (21) (as also depicted in Fig. 4 including both real and imaginary parts) is applied to/convoluted with the cepstrogram (cut with the first 13 coefficients from 31 dimension, which is same as 13 MFCCs). Thus, the final AMFB feature dimension is $13 \times 9 = 117$.

5.4.1 Performance with GMM-HMMs

As it can be seen in Fig. 7, AMFB features outperform the MFCC-LDA-MLLT features by approx. 1 % in average. Even though MFCCs with splicing and LDA-MLLT extract temporal information (with $L = R = 4$) and already achieve better performance than D-DD, AMFB features are shown to be more effective to achieve this aim by the dedicated temporal modulation filterbank design. This analysis also indicates that the broad temporal dynamics are crucial for feature extraction in reverberant environments in order to capture more valuable information which may behave robust against the reverberation effect. Moreover, Fig. 7 shows that further WER improvements of approx. 1.5–2 % can be obtained for AMFB features when the proposed SE algorithm is applied. This also indicates that the proposed SE algorithm is able to provide consistent benefits for both types of feature extraction for ASR systems with GMM-HMMs.

5.4.2 Performance with SGMM-HMMs

As aforementioned in Section 4.1, the SGMM approach uses a large UBM to cover the acoustic space and maps this space to a more specific subspace for each HMM state. Here, the amount of the Gaussians that are used for UBM training is computed as 10 times the input feature dimension, and 8000 total clustered phonetic states (sub-states) are defined for SGMM training. As shown

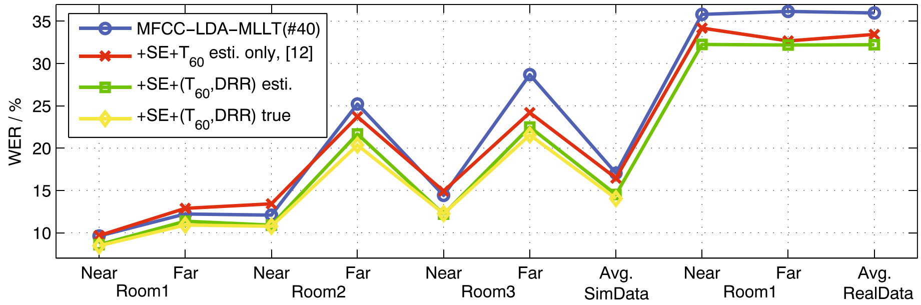

Fig. 6 ASR performance of the proposed SE algorithm for Dev. MFCC-LDA-MLLT features with MVN, a trigram LM and GMM-HMM are used for evaluation

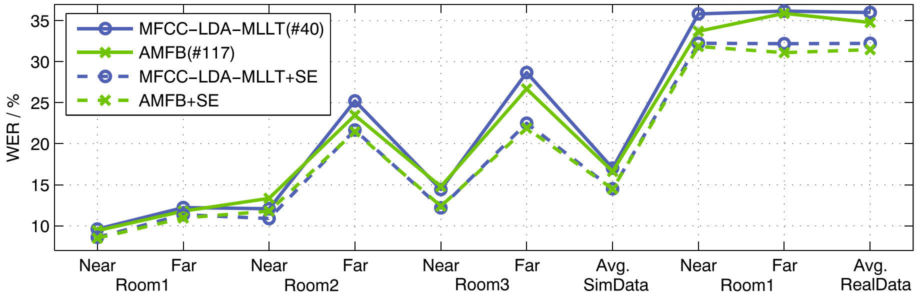

Fig. 7 ASR performance of AMFB features with GMM-HMM for Dev. The SE algorithm (*dashed lines*), MVN, and a trigram LM are used for evaluation. MFCC-LDA-MLLT features are employed as baseline for comparison

in Fig. 8, in general, SGMM-HMM outperforms HMM-GMM by 3–4 % in average for both feature types, i.e., for MFCC-LDA-MLLT as well as AMFB. Similar to the results with GMM-HMMs in Fig. 7, the SE algorithm provides 1–2 % absolute WER improvements also for SGMM-HMM.

It should be mentioned that training of SGMM is substantially more complex than the conventional GMM system [16], particularly when the input features have high dimensions as, e.g., AMFB features with dimension of 117. In general, the complexity of the calculations related to the covariance matrix estimation necessary for training of SGMM are quadratic w.r.t. the feature dimension. In many situations, less complexity might be preferable as long as the performance degrades not substantially. Hence, SGMM-HMM systems would become more efficient in combination with AMFB features when the feature dimensions can be reduced.

5.5 ASR performance of BN features

In order to explore the advantages of auditory modulation, filterbank features with high dimension, without loss of the compatibility to, e.g., SGMM and fMLLR w.r.t. complexity and parameter fine-tuning, BN features can be used. The DNN to generate BN features (denoted as BN{·} in the following) usually takes conventional features with long temporal context [64]. Since AMFB features already explore the temporal information, further context extension is omitted here, i.e., no splicing is applied, to generate BN{AMFB}. For comparison, MFCCs with context extension to 9 frames, i.e., $L = R = 4$, resulting in a dimension of $13 \times 9 = 117$, are used according to the LDA-MLLT features in the aforementioned GMM-HMM ASR performance. Five hidden layers (each with 1024 units) for the DNN are used and the middle layer is defined as the BN layer with 42 units [50]. The DNN training is carried out using stochastic mini-batch gradient descend with a mini-batch size of 512 samples. A learning rate of 0.005 for all layers during pre-training and a final stop learning rate of 0.0005 are used. DNN training was performed on a NVIDIA Tesla K20C GPU.

It can be observed from Fig. 9 that WERs increase by an average of 1–2 % when LDA-MLLT is used to reduce the feature dimension for AMFB features. This might be explained by the fact that AMFB features are dedicated and designed and it may be difficult to separate the high dimension information just via a linear transform. Useful feature information might be lost after such feature dimension reduction, which therefore degrades the ASR performance. In contrast, BN processing is capable of further improving the ASR performances for both feature types, meanwhile, reducing the feature dimension

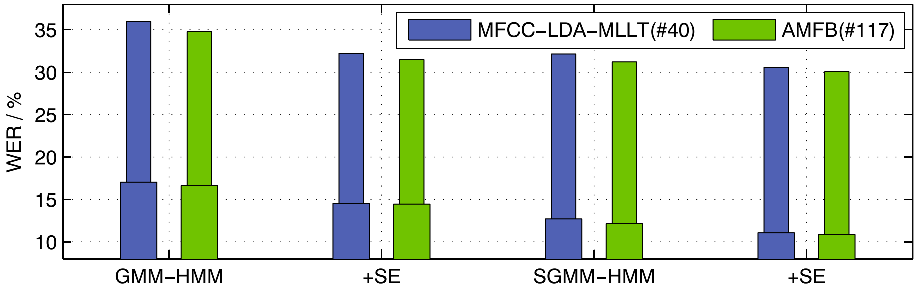

Fig. 8 ASR performance of AMFB features with SGMM-HMM for Dev. The SE algorithm, MVN, and a trigram LM are used for evaluation. The *thick bars* represent the average WERs for SimData and the *thin bars* for RealData. The average results with GMM-HMM in Fig. 7 are illustrated here for comparison

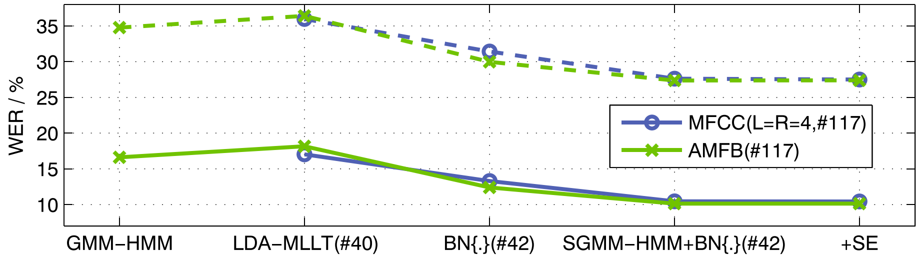

Fig. 9 ASR performance of BN features for Dev. MFCC($L = R = 4$) and AMFB features are used as the input to generate BN {MFCC} and BN {AMFB}, respectively. For comparison, LDA-MLLT is also applied to MFCC($L = R = 4$) and AMFB features to reduce their dimensions to 40. GMM- and SGMM-HMMs are used for evaluation. The SE algorithm is employed to SGMM-HMM afterwards. *Solid lines* are for SimData, and *dashed lines* for RealData

particularly for AMFB features. In general, WER reduction of 4–5 % absolute can be achieved by BN processing with GMM-HMMs.

Furthermore, the SGMM approach provides an average of nearly 3 % absolute WER improvement when replacing GMMs. AMFB features still outperform even spliced MFCCs. Moreover, ASR performance is further improved by 2.5–3 % absolute when BN processing is applied to AMFB features (cf. results in Fig. 8). However, the improvements obtained with the SE algorithm and BN features are rather small, particularly for SimData. It seems that our SE algorithm reduces the data variations during the multi-condition training, which instead, DNNs during BN processing usually prefer [58].

5.6 ASR performance with DNN-HMMs

The DNN uses an acoustic model implemented in Kaldi [65]. The training procedure consists of three phases: during the pre-training stage, a seven-layer deep belief network (DBN) with 2048 neurons for the hidden layers is trained as the stack of the restricted Boltzmann machines using a contrastive divergence algorithm. Since DNNs cannot align context-dependent states to time frames of the training data, an auxiliary GMM triphone system is trained with the ML criterion to provide alignments for the DNN training. Subsequently, the DBN is fine-tuned to classify feature vectors into triphone states using backpropagation via a stochastic gradient descent algorithm [56]. Context-dependent HMM states are replaced by the posterior probabilities, i.e., the softmax output layer of the seven-layer DNN. A validation set is required for training, hence we randomly selected 5 % of the whole training data, i.e., $\lfloor 7861 \times 0.05 \rfloor = 393$, for the validation, and the rest for the training.

5.6.1 Performance of FBANK features as DNN input

As an additional baseline feature, log-mel filterbank (FBANK) coefficients with 40 dimensions are used as DNN input. Additional splicing over 11 frames ($L = R = 5$) is performed for FBANK (and MFCC) features to capture temporal dynamics on feature level. The choice of 11 frames is based on the baseline feature input for the DNN-HMM reported in [17, 57, 58]. Note that splicing is not performed for AMFB features since they already capture temporal information inherently.

As listed in Table 2, AMFB features still perform better than MFCC with context extension for DNN-HMMs, which is consistent with the results obtained with (S)GMM-HMMs. However, FBANK features provide a further 2–3 % WER improvement compared to conventional MFCCs. Apparently, DNNs are capable of making good use of the more detailed information captured by the FBANK features [57], compared to the conventional MFCCs that eliminate spectral fine structure with the aim of obtaining a compact input representation with 13 parameters per time frame. Furthermore, it also shows that FBANK features perform even better than AMFB features, which indicates that the original AMFB features might not be preferable to DNNs, like the conventional MFCCs.

5.6.2 Auditory modulation filterbank features for DNNs

As the results from the previous section show, the different properties of a DNN-based architecture compared to classic approaches result in different baseline features. DNNs do not require decorrelated or highly condensed feature input (e.g., MFCCs), but profit from additional information contained in simple log-mel spectrograms. Based on this observation, we modified the AMFB features: AMFB-FBANK features are calculated by applying the temporal modulation filtering directly to the log-mel-spectrogram (in contrast to previous processing that was based on the *cepstro*gram in Section 5.4), and thus, the final AMFB-FBANK feature dimension is $40 \times 9 = 360$.

As shown in Table 2, WERs are reduced by an average 1.5 % absolute with our modified filter sets, i.e., AMFB-FBANK, compared to the previous processing AMFB features. It can be also seen that AMFB-FBANK features even outperform FBANK features by an average 1%, which indicates that the AMFB design for temporal dynamics

Table 2 ASR WERs (%) of various types of features as DNN input with DNN-HMM for Dev

Dev.	*1ch* ASR task	SimData							RealData		
	Utterance-based batch processing mode	Room1		Room2		Room3		Avg.	Room1		Avg.
	with multi-condition training	Near	Far	Near	Far	Near	Far		Near	Far	
DNN-HMM	MFCC(L=R=5,#143)	6.61	8.82	8.31	17.03	9.97	20.03	11.79	28.70	28.91	28.80
	AMFB(#117)	6.22	7.74	7.49	14.52	9.79	16.64	10.39	25.39	27.00	26.19
	FBANK(L=R=5,#440)	5.80	6.91	7.17	13.90	8.28	15.01	9.50	26.08	25.91	25.99
	FBANK(L=R=5,#440)+SE	5.83	7.23	7.44	14.84	8.68	14.89	9.81	25.60	25.76	25.67
	AMFB-FBANK(#360)	5.19	6.32	7.69	12.50	7.72	13.95	8.89	22.27	26.66	24.45
	AMFB-FBANK(#360)+SE	5.70	6.86	7.44	12.10	7.79	13.33	8.86	22.14	25.56	23.84

extraction is superior to the context extension by splicing for DNN usage. In addition, when the SE algorithm is applied in the DNN-HMM scenario, it seems that its advantage is not obvious as applied for (S)GMM-HMMs, particularly for SimData. This is also consistent with the results for BN processing (cf. Section 5.5). It might be explained by the fact that DNNs perform robust to quite small variations, and sometimes it may hinder the generalization when removing too much variability from the data [58], e.g., by our proposed SE algorithm used for pre-processing of the data. However, for RealData, SE algorithm leads to slightly better results, which might be due to the more evident mismatch between RealData and the training data, for which the SE algorithm alleviates such mismatch.

5.6.3 Performance of the splicing effect

Since DNNs are powerful in transforming features through many layers of nonlinear transformations, it is common to splice the input features with a long context window to learn the temporal dynamic information automatically.

Figure 10 shows that the benefit from splicing FBANK features (without other temporal dynamics) is proportional to the splicing window length, i.e., longer temporal analysis windows should be preferred in DNN architectures, at least in reverberant conditions. However, longer and longer splicing windows would also increase the risk that the stochastic gradient descent algorithm for training might fail to find a better local optimum [66], so that the ASR performance might decrease instead. On the other hand, it seems that longer splicing windows do not help AMFB-FBANK features in general. Figure 10 shows that a splicing window of 3, i.e., $L = R = 1$, is a good choice for AMFB-FBANK features which already extract temporal dynamics internally. In this case, longer splicing windows (higher dimension) might affect the training algorithm to find a better local optimum, for which the parameter tuning is probably required to further improve the DNN performance.

5.7 ASR performance for the REVERB challenge

It has been shown that our proposed front-end technologies are capable of improving the ASR performance in reverberant environments with the REVERB challenge Dev.; at the same time, they were shown to be beneficial in state-of-the-art ASR back-end approaches. Good results were obtained when combining the SE algorithm with auditory modulation features and BN processing for SGMM-HMM, as well as the modified auditory modulation features for DNN-HMM. The results for the REVERB challenge with Eval. are summarized in Table 3. For BN features with SGMM-HMM, discriminative training with bMMI (boosted factor of 0.1 and 4 iterations [54]) and adaption with fMLLR [53] are further employed. For the DNN-HMM system, an additional discriminative training can be performed based on the fine-tuned DNN-HMM

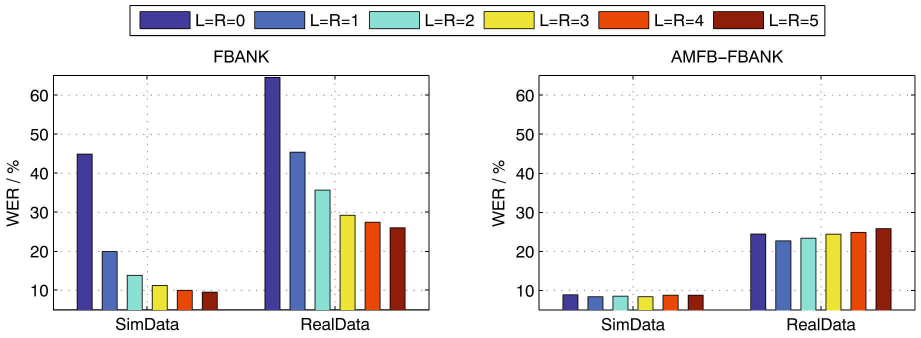

Fig. 10 ASR performance of splicing effect with DNN-HMM for Dev. FBANK and AMFB-FBANK features are used for evaluation. Splicing windows are chosen as 0, 3, 5, 7, 9, and 11 with equal lengths for the left and right extension

Table 3 ASR WERs (%) of our proposed front-end technologies with state-of-the-art back-end strategies for Eval., which can be compared to the original REVERB challenge results. An extended training data (ext.) can be additionally applied to DNN-HMM

REVERB challenge Eval.		*1ch* ASR task	SimData							RealData		
		Utterance-based batch processing mode	Room1		Room2		Room3		Avg.	Room1		Avg.
		with multi-condition training	Near	Far	Near	Far	Near	Far		Near	Far	
Results from challenge		REVERB baseline	20.84	21.72	23.43	38.59	28.43	44.79	29.62	59.09	55.81	57.45
		Our submitted results from [12]	13.64	14.93	16.11	25.65	19.77	30.51	20.09	41.97	42.27	42.12
		Results from [11]	5.90	6.60	7.90	12.20	8.70	13.20	9.10	32.60	32.30	32.50
		Results from [11] (ext.)	5.10	5.60	6.70	11.50	7.60	11.60	8.00	27.10	27.90	27.50
This study	SGMM	BN{MFCC(L=R=4,#117)}(#42)	6.52	7.30	8.31	13.41	9.86	16.85	10.37	25.07	24.48	24.77
		BN{MFCC(L=R=4,#117)}(#42)+SE	6.20	7.52	8.70	14.37	9.72	17.14	10.60	24.75	24.10	24.42
		BN{AMFB(#117)}(#42)	5.78	6.61	8.37	12.60	9.25	14.48	9.51	24.50	24.21	24.35
		BN{AMFB(#117)}(#42)+SE	6.34	7.32	9.08	13.40	9.67	14.94	10.12	24.38	23.90	24.14
	DNN	FBANK(L=R=5,#440)	5.66	7.47	8.44	14.11	9.84	17.20	10.45	28.53	28.46	28.49
		FBANK(L=R=5,#440)+sMBR	5.86	6.91	7.17	11.51	8.70	14.48	9.10	23.22	24.61	23.91
		FBANK(L=R=5,#440)+sMBR+SE	6.32	6.90	7.83	12.46	8.53	14.51	9.42	22.69	24.47	23.58
		AMFB-FBANK(L=R=1,#1080)	5.54	6.22	7.57	11.33	8.17	12.98	8.63	24.15	27.99	26.07
		AMFB-FBANK(L=R=1,#1080)+sMBR	5.71	6.39	7.17	10.53	7.36	11.82	8.16	22.68	23.43	23.05
		AMFB-FBANK(L=R=1,#1080)+sMBR+SE	5.86	6.42	7.22	10.71	7.40	11.94	8.25	21.93	23.16	22.71
	DNN+(ext.)	FBANK(L=R=5,#440)	4.90	6.23	6.49	12.88	8.09	16.02	9.09	26.38	26.43	26.40
		FBANK(L=R=5,#440)+SE(test)	5.22	6.39	7.06	11.65	7.51	13.34	8.52	24.08	25.42	24.75
		FBANK(L=R=5,#440)+sMBR	4.76	6.00	5.85	11.40	7.61	13.97	8.26	25.36	24.85	25.10
		FBANK(L=R=5,#440)+sMBR+SE(test)	4.91	6.05	6.41	10.90	7.01	12.83	8.01	25.14	24.78	24.96
		AMFB-FBANK(L=R=1,#1080)	5.15	5.79	7.28	10.87	8.10	13.25	8.40	23.70	25.05	24.37
		AMFB-FBANK(L=R=1,#1080)+SE(test)	5.17	5.95	7.14	10.75	8.12	13.19	8.38	23.16	24.38	23.77
		AMFB-FBANK(L=R=1,#1080)+sMBR	4.93	6.01	6.37	10.26	7.80	12.13	7.91	24.15	24.92	24.53
		AMFB-FBANK(L=R=1,#1080)+sMBR+SE(test)	5.07	5.86	6.30	10.34	7.66	11.99	7.86	23.89	23.50	23.69

system, using the MBR approach on state labels (sMBR) with 4 iterations [65]. For comparison to results obtained in this study in the *utterance-based batch processing* mode for *1ch* ASR task, we report our previous result [12], as well as results from a second contribution to the REVERB challenge [11] that was very successful and highlighted the improvements by increasing the amount of training data with DNNs.

In general, AMFB features improve the baseline (e.g., MFCCs with (S)GMM-HMM or FBANK features with DNN-HMM). The proposed SE algorithm does not result in obvious improvements for both investigated architectures that included a DNN (BN features and the DNN-HMM) for SimData, however, it improves the performance for RealData. By this, our best results were obtained when combining the proposed front-end technologies with the DNN-HMM. Compared to our previous workshop results [12], absolute WER improvements of 11.93 and 19.41 % are achieved for SimData and for RealData in average, respectively.

Since it is known from many machine learning studies that an extension of training data often improves classification scores, an additional set of experiments was carried out based on DNN-HMM system which provides the best results. The corresponding results are labeled as DNN+(ext.) in Table 3. The extended training data is generated based on [11], which consists of the WSJCAM0 clean training data (7861 utterances), WSJCAM0 training data recorded with the secondary microphone (7387 utterances), and the multi-condition training data but with 20, 15, and 10 dB DSNRs (each with 7861 utterances). Other parameter settings are kept the same as described in Section 5.6. It is shown in [20] that enhancing the training data by a pre-processing strategy may also degrade the DNN performance. Hence, our SE algorithm is not applied to such extended training data, but only to the test data (same as in [11]). It can be seen from Table 3 that an extension of the training data generally improves the performance of DNN-HMM systems for both types of features, presumably since DNNs profit from large feature variance seen during training. Larger improvements are obtained with RealData than with SimData, which is consistent with the observation in [11]. However, 2 % absolute improvements for RealData in average are obtained here, which is smaller than the 5 % improvement for RealData reported in [11] (from 32.5 to 27.5 %). This can probably be explained by the notion in [11] that the parameter tuning was not performed with the DNN-HMM trained without extended training data. AMFB-FBANK features still outperform FBANK features by 1 % absolute in average, and the SE algorithm (applied only to the test data) provides further improvements for both, SimData and RealData. This indicates that our proposed front-end technologies are also beneficial for DNN-HMM back-ends that are trained on large amount of speech data. Furthermore, although the additional discriminative training (by sMBR) provides consistent improvements for the corresponding DNN-HMM, the relative profits are smaller compared to the corresponding boost for the DNN-HMM systems without extended training data. Extending the training data in such *simulated* way consistently improves the performance for SimData, while it does not always help for RealData. Possibly, discriminative DNN training might overfit the extended training data, which as a result does not fit well to distortions of the realistic test data.

6 Conclusions

This study analyzed novel ASR front-end technologies and their compatibility with established and recent back-end schemes with the aim of improving ASR performance in reverberant environments. A dereverberation algorithm for speech enhancement and an auditory inspired feature extraction were evaluated based on the *1ch* ASR task of the REVERB challenge. The enhancement component uses TCS and a parametric MMSE estimator to mitigate the speech distortions and artifacts, together with a more reliable late reverberation PSD estimator. For this, the RIR model was selected with a novel concept for estimating room parameters, i.e., the reverberation time T_{60} and the DRR. The auditory modulation filterbank features have been analyzed, which are obtained by extracting temporal amplitude modulations using a filterbank (AMFB features). Our contribution to the REVERB challenge [12] has shown that the aforementioned techniques are suitable to increase ASR performance also when *combined* and linked with a standard back-end implemented based on a well-established toolkit, i.e., HTK [13]. In this contribution, we analyzed the respective techniques for state-of-the-art implementations of different back-end types, thus, considered traditional GMM-HMM systems, their extension subspace GMM-HMM (SGMM-HMM), as well as deep neural nets serving as acoustic model in a DNN-HMM system.

For conventional GMM-HMM systems, we found similar results as in [12]: The SE algorithm provides consistent improvements, but now covering various reverberant scenarios when a joint estimation of room parameters (T_{60}, DRR) is performed. When auditory AMFB features are extracted from the enhanced signal, a further gain in ASR performance is achieved. It has been also seen from our previous work [12] that, when analyzing the individual benefit obtained by each step, a larger contribution to this benefit came from the auditory feature component. One reason for the improved performance using auditory features is the inclusion of a larger temporal context on feature level compared to MFCCs with D-DD. Hence, in this contribution, we also investigated how temporally spliced MFCCs (down-projected to a lower-dimensional space using LDA-MLLT) perform, since these features also cover a wider temporal context. The spliced LDA-MLLT features even lowered the WER, highlighting the relevance of the inclusion of temporal context on a feature level in reverberant conditions. However, AMFB features produced lower WERs than the LDA-MLLT features. A major difference between these two is that for auditory features a local filtering of the spectrogram is performed (that has the potential to locally increase the speech energy, and could enhance local information such as formant frequencies), in contrast to the spectral modulation analysis in MFCCs.

For SGMM-HMM systems, we obtained generally lower WERs compared to the GMM-HMM system, while the overall trend of results (improvements from speech enhancement and auditory features) was preserved. However, these improvements come at the cost of increased complexity for training, which is especially true for the high-dimensional AMFB features. As a solution to reduce the costs for training, the use of BN features derived from a trained DNN was investigated. The DNN used AMFB features as input, and the relatively low-dimensional BN features were used as a compact input representation for the SGMM-HMM. With this procedure, even better performance than with high-dimensional input was obtained (2–3 % WER reduction), and also the model complexity was considerably reduced.

As a third back-end option, it was analyzed if the findings observed for GMM-based architectures are transferable to hybrid systems that employ a DNN instead of the GMM component. The conventional MFCC baseline was replaced with a spectrogram input, which reflects the capability of DNNs to self-learn the salient patterns in the time-frequency domain when recognizing reverberant speech. Following this notion, we modified the auditory features to directly operate on a time-frequency representation instead of cepstral patterns. Additional splicing was not performed, since we assume that our auditory features capture a sufficient amount of temporal dynamics by modulation filtering (that is intrinsic for AMFB features). With this modification, our auditory features outperform the competitive FBANK baseline, albeit with smaller relative improvements than observed for the (S)GMM-based systems. Although it has been shown that DNNs are capable of processing a raw representation of the input signal (e.g., Tueske et al. have shown that even using the time signal as input to a DNN produces acceptable results [67]), this result shows that a pre-selection of input data (such as the relevant modulation frequencies and the application of knowledge about the auditory system) is an approach that can help lowering WERs in ASR system with DNN-HMM architectures.

For the combination of speech enhancement with deep learning (either in the hybrid model or when using BN features as an intermediate representation), the baseline was not improved on average, and no significant improvements were obtained when combining the speech enhancement with auditory features for SimData of the REVERB challenge. A possible explanation for this effect is the fact that enhancement algorithms not only remove effects of reverberation and additive noise, but also the target speech is partially affected by the enhancement process. This potentially removes fine-grain detail of the target signal. If the removal of unwanted signal parts does not outweigh this disadvantage—for instance because the classifier is well-adapted to the interferences (the variations might be learnt by DNNs), and hence does not profit from its removal—overall performance would be harmed, which is probably the case here. However, we also found that for strong mismatches of training and test data (in the RealData scenario), our SE algorithm is still capable of improving ASR performance with DNNs, presumably since it alleviates the train-test-mismatch and provides a better match between the trained model and test observations. Hence, in these situations that are of special importance when a high robustness of the system is desired, both the auditory feature processing as well as the proposed speech enhancement result in an improved recognition performance.

The use of additional training data for the DNN-HMM system results in further improvements for both, the FBANK baseline and our auditory features. Finally, our SE algorithm is able to provide consistent improvements for both, SimData and RealData, when it is only applied to the test data. A possible explanation is that the robustness of DNN-based recognizer—although being increased by training on an extended data set—is not capable of completely ignoring cues induced by late reverberation and additive noises. Hence, our SE algorithm can provide a further benefit by partially removing the interferences from the test data.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The research leading to these results has received funding from the EU Seventh Framework Programme project Dereverberation and Reverberation of Audio, Music, and Speech (DREAMS) under grant agreement ITN-GA-2012-316969, from the DFG-Cluster of Excellence EXC 1077/1 "Hearing4all", and from the MWK PhD Program "Signals and Cognition".

Author details

[1]Fraunhofer Institute for Digital Media Technology IDMT, Project Group Hearing, Speech and Audio Technology (HSA), Oldenburg, Germany. [2]University of Oldenburg, Department of Medical Physics and Acoustics, Oldenburg, Germany. [3]University of Oldenburg, Cluster of Excellence Hearing4All, Oldenburg, Germany.

Received: 22 February 2015 Accepted: 23 July 2015
Published online: 05 August 2015

References

1. M Wölfel, J McDonough, *Distant Speech Recognition*. (John Wiley & Sons Ltd, United Kingdom, 2009)
2. T Yoshioka, A Sehr, M Delcroix, K Kinoshita, R Maas, T Nakatani, W Kellermann, Making Machines Understand Us in Reverberant Rooms: Robustness against Reverberation for Automatic Speech Recognition. IEEE Signal Process. Mag. **29**(6), 114–126 (2012)
3. EAP Habets, *Single- and Multi-Microphone Speech Dereverberation using Spectral Enhancement. PhD thesis*. (University of Eindhoven, Eindhoven, The Netherlands, 2007)
4. T Nakatani, T Yoshioka, K Kinoshita, M Miyoshi, B-H Juang, Speech dereverberation based on variance-normalized delayed linear prediction. IEEE Trans. Audio, Speech, Lang. Process. **18**(7), 1717–1731 (2010)
5. I Kodrasi, S Goetze, S Doclo, Regularization for partial multichannel equalization for speech dereverberation. IEEE Trans. Audio, Speech Lang. Process. **21**(9), 1879–1890 (2013)
6. N Moritz, J Anemüller, B Kollmeier, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Amplitude modulation spectrogram based features for robust speech recognition in noisy and reverberant environments (Prague, Czech Republic, 2011), pp. 5492–5495
7. A Sehr, R Maas, W Kellermann, Reverberation model-based decoding in the Logmelspec domain for robust distant-talking speech recognition. IEEE Trans. Audio, Speech Lang. Process. **18**(7), 1676–1691 (2010)
8. 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 *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. The REVERB Challenge: a common evaluation framework for dereverberation and recognition of reverberant speech (New Paltz, NY, USA, 2013)
9. B Cauchi, I Kodrasi, R Rehr, S Gerlach, A Jukić, T Gerkmann, S Doclo, S Goetze, in *Proc. of the REVERB Challenge*. Joint dereverberation and noise reduction using beamforming and a single-channel speech enhancement scheme (Florence, Italy, 2014)
10. F Weninger, S Watanabe, JL Roux, JR Hershey, Y Tachioka, J Geiger, B Schuller, G Rigoll, in *Proc. of the REVERB Challenge*. T MERL/MELCO/TUM System for the REVERB Challenge using Deep Recurrent Neural Network Feature Enhancement (Florence, Italy, 2014)
11. M Delcroix, T Yoshioka, A Ogawa, Y Kubo, M Fujimoto, N Ito, K Kinoshita, M Espi, T Hori, T Nakatani, A Nakamura, in *Proc. of the REVERB Challenge*. Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB Challenge (Florence, Italy, 2014)
12. F Xiong, N Moritz, R Rehr, J Anemüller, BT Meyer, T Gerkmann, S Doclo, S Goetze, in *Proc. of the REVERB Challenge*. Robust ASR in reverberant environments using temporal cepstrum smoothing for speech enhancement and an amplitude modulation filterbank for feature extraction (Florence, Italy, 2014)
13. S Young, G Evermann, M Gales, T Hain, D Kershaw, XA Liu, G Moore, J Odell, D Ollason, D Povey, V Valtchev, P Woodland, *The HTK Book (for HTK Version 3.4)*. (Cambridge University Engineering Department, Cambridge, 2009)
14. D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlíček, Y Qian, P Schwarz, J Silovský, G Stemmer, K Veselý, in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. The Kaldi speech recognition toolkit (Big Island, HI, USA, 2011)
15. F Grézl, M Karafiát, S Kontáir, J Černocký, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Probabilistic and bottle-neck features for LVCSR of meetings, vol. 4 (Honolulu, HI, USA, 2007), pp. 757–760
16. D Povey, L Burget, M Agarwal, P Akyazi, F Kai, A Ghoshal, O Glembek, N Goel, M Karafiát, A Rastrow, RC Rose, P Schwarz, S Thomas, The subspace Gaussian mixture model - a structured model for speech recognition. Comput. Speech Lang. **25**(2), 404–439 (2011)
17. G Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, B Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
18. A Sehr, *Reverberation Modeling for Robust Distant-Talking Speech Recognition. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg*, (Germany, 2009)
19. C Breithaupt, R Martin, in *ITG Conference on Voice Communication (Sprachkommunikation)*. DFT-based speech enhancement for robust automatic speech recognition (Aachen, Germany, 2008)
20. M Seltzer, D Yu, Y Wang, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. An Investigation of deep neural networks for noise robust speech recognition (Vancouver, Canada, 2013), pp. 7398–7402
21. C Breithaupt, M Krawczyk, R Martin, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Parameterized MMSE Spectral magnitude estimation for the enhancement of noisy speech (Las Vegas, NV, USA, 2008), pp. 4037–4040
22. R Martin, Noise power spectral density estimation based on optimal smoothing and minimum statistics. IEEE Trans. Speech Audio Process. **9**(5), 504–512 (2001)
23. T Gerkmann, R Martin, On the statistics of spectral amplitudes after variance reduction by temporal cepstrum smoothing and cepstral nulling. IEEE Trans. Signal Process. **57**(11), 4165–4174 (2009)
24. K Lebart, JM Boucher, PN Denbigh, A new method based on spectral subtraction for speech dereverberation. Acta Acustica United Acustica. **87**(3), 359–366 (2001)
25. H Kuttruff, *Room Acoustics*, 4th edn. (Spon Press, London, 2000)
26. EAP Habets, S Gannot, I Cohen, Late reverberant spectral variance estimation based on a statistical model. IEEE Signal Process. Lett. **16**(9), 770–773 (2009)
27. F Xiong, S Goetze, BT Meyer, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Blind Estimation of Reverberation Time based on Spectro-Temporal Modulation Filtering (Vancouver, Canada, 2013), pp. 443–447
28. F Xiong, S Goetze, BT Meyer, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Estimating room acoustic parameters for speech recognizer adaptation and combination in reverberant environments (Florence, Italy, 2014)
29. N Moritz, J Anemüller, B Kollmeier, An auditory inspired amplitude modulation filter bank for robust feature extraction in automatic speech recognition. IEEE Trans. Audio, Speech and Language Processing. **23**(11), 1926–1937 (2015)
30. G Langner, CE Schreiner, Periodicity coding in the inferior colliculus of the Cat. I. Neuronal Mechanisms. J. Neurophysiol. **60**(6), 1799–1822 (1988)
31. N Mesgarani, S David, S Shamma, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Representation of phonemes in primary auditory cortex: how the brain analyzes speech, vol. 4 (Honolulu, HI, USA, 2007), pp. 765–768
32. T Dau, B Kollmeier, A Kohlrausch, Modeling auditory processing of amplitude modulation. I, Detection and masking with narrow-band carriers. J. Acoust. Soc. Am. **102**(5), 2892–2905 (1997)
33. BT Meyer, B Kollmeier, Robustness of spectro-temporal features against intrinsic and extrinsic variations in automatic speech recognition. Speech Commun. **53**(5), 753–767 (2011)
34. SB David, P Mermelstein, Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoustics, Speech Signal Process. **28**(4), 357–366 (1980)
35. B Atal, Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. J. Acoust. Soc. Am. **55**(6), 1304–1322 (1974)
36. B Cauchi, I Kodrasi, R Rehr, S Gerlach, A Jukić, T Gerkmann, S Doclo, S Goetze, Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech. EURASIP Journal on Advances in Signal Processing. **2015**, 61 (2015)
37. C Breithaupt, T Gerkmann, R Martin, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. A Novel A Priori SNR Estimation Approach based on Selective Cepstro-Temporal Smoothing (Las Vegas, NV, USA, 2008), pp. 4897–4900
38. Y Ephraim, D Malah, Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. IEEE Trans. Acoustics, Speech Signal Process. **32**(6), 1109–1121 (1984)
39. H Meutzner, A Schlesinger, S Zeiler, D Kolossa, in *Proc. 2nd CHiME Workshop on Machine Listening in Multisource Environments*. Binaural signal processing for enhanced speech recognition robustness in complex listening environments (Vancouver, Canada, 2013), pp. 7–12
40. J Eaton, ND Gaubitch, PA Naylor, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Noise-robust reverberation time estimation

using spectral decay distributions with reduced computational cost (Vancouver, Canada, 2013), pp. 161–165

41. F Xiong, BT Meyer, S Goetze, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. A study on joint beamforming and spectral enhancement for robust speech recognition in reverberant environments (Brisbane, Australia, 2015), pp. 5043–5047
42. C Breithaupt, *Noise Reduction Algorithms for Speech Communications - Statistical Analysis and Improved Estimation Procedures. PhD thesis*. (Ruhr-Universität Bochum, Bochum, Germany, 2008)
43. KE Muller, Computing the Confluent Hypergeometric Function, M(a,b,x). Numerische Mathematik. **90**(1), 179–196 (2001)
44. R Maas, EAP Habets, A Sehr, W Kellermann, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. On the application of reverberation suppression to robust speech recognition (Kyoto, Japan, 2012), pp. 297–300
45. BT Meyer, SV Ravuri, MR Schädler, N Morgan, in *Interspeech*. Comparing Different Flavors of Spectro-Temporal Features for ASR (Florence, Italy, 2011), pp. 1269–1272
46. MR Schädler, BT Meyer, B Kollmeier, Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition. J. Acoust. Soc. Am. **131**(5), 4134–4151 (2012)
47. S Haykin, *Neural Networks and Learning Machines*, 3rd edn. (Prentice Hall, USA, 2008)
48. QuickNet package. http://wwwl.icsLberkeley.edu/Speech/qn.html
49. MR Schroeder, New method of measuring reverberation time. J. Acoust. Soc. Amer. **37**(3), 409–412 (1965)
50. D Yu, ML Seltzer, in *Proc. Interspeech*. Improved Bottleneck Features using Pretrained Deep Neural Networks (Florence, Italy, 2011), pp. 237–240
51. JK Baker, *Stochastic Modeling for Automatic Speech Recognition. Speech Recognition*. (DR Reddy, ed.), (New York: Academic, 1975)
52. BH Juang, S Levinson, M Sondhi, Maximum likelihood estimation for multivariate mixture observations of Markov chains. IEEE Trans. Inform. Theory. **32**(2), 307–309 (1986)
53. D Povey, K Yao, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. A basis method for robust estimation of constrained MLLR (Prague, Czech Republic, 2011), pp. 4460–4463
54. D Povey, D Kanevsky, B Kingsbury, B Ramabhadran, G Saon, K Visweswariah, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Boosted MMI for Model and Feature-Space Discriminative Training (Las Vegas, NV, USA, 2008), pp. 4057–4060
55. M Gibson, T Hain, in *Proc. Interspeech*. Hypothesis spaces for minimum Bayes risk training in large vocabulary speech recognition (Pittsburgh, Pennsylvania, USA, 2006), pp. 2406–2409
56. DE Rumelhart, GE Hinton, RJ Williams, Learning Internal Representations by Error Propagation. Parallel distributed processing: Explorations in the microstructure of cognition. **1: Foundations. MIT Press** (1986). ISBN:0-262-68053-X
57. A Mohamed, GE Dahl, G Hinton, Acoustic modeling using deep belief networks. IEEE Trans. Audio Speech Lang. Process. **20**(1), 14–22 (2012)
58. D Yu, ML Seltzer, J Li, J-T Huang, F Seide, in *Proc. of ICLR*. Feature learning in deep neural networks - studies on speech recognition tasks, (2013). arXiv:1301.3605v3
59. T Robinson, J Fransen, D Pye, J Foote, S Renals, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. WSJCAM0: A British English Speech Corpus for Large Vocabulary Continuous Speech Recognition (Detroit, Michigan, USA, 1995), pp. 81–84
60. M Lincoln, I McCowan, J Vepa, HK Maganti, in *IEEE Workshop on Automatic Speech Recognition and Understanding*. The Multi-Channel Wall Street Journal Audio Visual Corpus (MC-WSJ-AV): Specification and Initial Experiments (San Juan, Puerto Rico, 2005), pp. 357–362
61. J Garofalo, D Graff, D Paul, D Pallett, in *Linguistic Data Lconsortium (LDC)*. CSR-I (WSJ0) Complete (Philadelphia, USA, 2007)
62. RA Gopinath, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Maximum Likelihood Modeling with Gaussian Distributions for Classification, vol. 2 (Seattle, WA, USA, 1998), pp. 661–664
63. Y Tachioka, T Narita, F Weninger, S Watanabe, in *Proc. of the REVERB Challenge*. Dual system combination approach for various reverberant environments with dereverberation Techniques (Florence, Italy, 2014)
64. F Grézl, P Fousek, in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Optimizing Bottle-Neck Features for LVCSR (Las Vegas, NV, USA, 2008), pp. 4729–4732
65. K Veselý, A Ghosal, L Burget, D Povey, in *Proc. Interspeech*. Sequence-discriminative training of deep neural networks (Lyon, France, 2013), pp. 2345–2349
66. J Li, D Yu, J-T Huang, Y Gong, in *IEEE Workshop on Spoken Language Technology*. Improving Wideband Speech Recognition using Mixed-Bandwidth Training Data in CD-DNN-HMM (Miami, FL, USA, 2012), pp. 131–136
67. Z Tüske, P Golik, R Schlüter, H Ney, in *Proc. Interspeech*. Acoustic modeling with deep neural networks using raw time signal for LVCSR (Singapore, 2014), pp. 890–894