A MODULATION FEATURE SET FOR ROBUST AUTOMATIC SPEECH RECOGNITION IN ADDITIVE NOISE AND REVERBERATION

Xiaoyu Liu, Roozbeh Sadeghian, Stephen A. Zahorian

Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY, 13902, USA

{xliu6, rsadegh1, zahorian}@binghamton.edu

ABSTRACT

In this paper, a feature set referred to as Discrete Cosine Series (DCS) is proposed for noise robust Automatic Speech Recognition (ASR). Unlike many other robust algorithms which use various forms of "long term" processing, DCS uses a small frame spacing to facilitate separating speech from noise and also for other benefits. Spectral and temporal modulations are performed separately using only a small number of modulation filters. ASR experiments show the effectiveness of individual components of the DCS algorithm. The DCS features yield higher accuracy ASR for both additive noise and reverberation, as compared to several other advanced robust algorithms.

Index Terms— Noise and reverberation robust, ASR, small frame spacing, discrete cosine, modulation

1. INTRODUCTION

For many years, numerous efforts have been devoted to developing signal processing based robust speech features to improve ASR in additive noise and reverberation. Even using Deep Neural Networks (DNNs) as state-of-the-art ASR recognizers, which, by themselves, are able to obtain learned patterns robust to extraneous variabilities [1,2], highly robust "raw" features are still crucial, because they effectively reduce the mismatch between clean and corrupted data.

Various forms of "long term" processing are often used in noise robust front ends. For example, in the Power function-based Power Distribution Normalization (PPDN) method [3], a frame length of 100 ms, rather than the typically used 25 ms is used to capture the slowly varying property of noise. The Long-Term Log-Spectral Subtraction (LTLSS) [4,5] uses a window longer than 1 second to model the far-field Room Impulse Responses (RIRs) in reverberation. In addition to long term frames, long term Linear Prediction (LP) usually deploys hundreds or even thousands of LP coefficients to estimate the inverse filter for the RIRs, such as in [6,7]. Long term operations are usually computationally costly, and due to large inter-frame correlation, speech reconstruction is often needed before final feature extraction, such as in [3,8], which again adds complexity.

Another shortcoming of many algorithms is that they usually have good effects for either additive noise or reverberation only, rather than for both, due to different mechanisms of corruption. Thus, one might need to first use algorithms dedicated to de-reverberation, such as the Blind Spectral Weighting (BSW) [9], Weighted Prediction Error (WPE) [10] etc., for a preprocessing pass, and append other algorithms to reduce additive noise. Again, feature extraction becomes multi-stage processing, which is less efficient, and also causes more spectral distortion for clean data. Some algorithms bring improvements for both scenarios. Delta-Spectral Cepstral Coefficients (DSCCs) [11] reduce spectral mismatch by computing the delta features in the spectral domain. Relative spectra (RASTA) [12] uses a modulation filter over the log spectrum to remove the slowly varying distortion, and Power-Normalized Cepstral Coefficients (PNCCs) [13] employ temporal masking and spectral subtraction for reverberation and noise respectively.

In our previous work [14], we developed a modulation feature algorithm, which improved phoneme recognition for clean speech, but lacked noise robustness. The new Discrete Cosine Series (DCS) method proposed in this paper re-develops [14]. DCS extracts features that are robust to both additive noise and reverberation in a single pass of processing. Static and modulation features are computed independently from a power-law scaled and an unscaled gammatone spectrogram respectively. The entire processing is based on a very small (2-ms) frame spacing, which improves robustness, and does not require speech reconstruction. DCS employs only a small number of 1-D modulation filters, which reduces computations. In this paper, we first show the individual contribution of the major components in DCS, and we then show that DCS outperforms several other widely-used noise robust methods for large vocabulary ASR.

2. STRUCTURE OF THE DCS FRONT END

2.1. Static feature extraction

The left side of Figure 1 depicts the static feature extraction. Input speech is pre-emphasized and segmented into 25 ms frames. The initial frame spacing is 2 ms, instead of 10 ms which is typically used for ASR. The motivation is that noise is generally slowly varying compared with speech. To allow a model which separates the slowly changing nature of noise from the rapidly varying speech, the signal is sampled with a high frame rate (500 frames/sec). Thus, by observing noise and

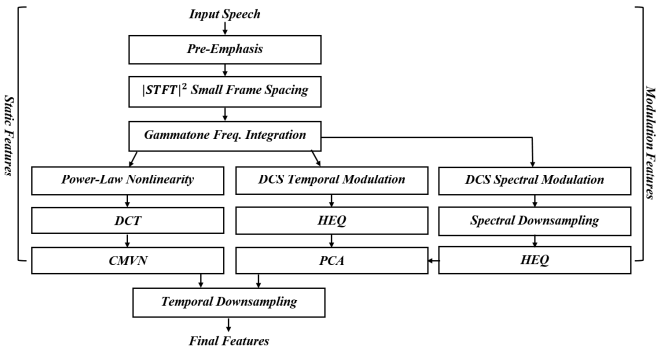

Fig.1. Block diagram of the DCS algorithm.

speech signals every 2 ms, the relative rate of change between the two components is effectively "magnified." A 512-point FFT is computed for each frame, generating a 256-point magnitude-squared FFT power spectrogram.

This FFT power spectrum is weighted by a magnitude-squared 40-channel gammatone filterbank over the frequency range of 200 Hz to 8000 Hz. For each channel, the area under the squared transfer function is normalized to 1. The resulting gammatone spectrogram is mapped to a perceptual loudness scale by a power-law nonlinearity $(\cdot)^{0.1}$. We used gammatone frequency integration and power-law scaling because of their known effects of noise robustness as reported in [12,13,15,16]. Next, a DCT converts the spectrogram to 13 cepstral coefficients. Finally, the mean and variance of each cepstrum is normalized by utterance-based CMVN [17].

2.2. DCS temporal and spectral modulations

The modulation features consist of temporal and spectral modulations. These modulation features are computed over the original gammatone spectrogram without power-law scaling; otherwise, the recognition accuracy was found to severely degrade. We hypothesize this is because the amplitude-scaled power spectra of the clean speech and the noise are no longer additive, and they become dependent in a nonlinear way. Thus, it's difficult for the modulation filters to separate the interlaced speech and noise by their relative rate of change.

Both the temporal and spectral modulations take a discrete cosine form. The temporal modulation feature $DCS_{T,i}(t_c, f_c)$ at the time-frequency bin (t_c, f_c) produced by the ith temporal filter is computed by a one-dimensional convolution as:

$$DCS_{T,i}(t_c, f_c) = \int_{-\frac{1}{2}}^{\frac{1}{2}} G_{t_c}(t, f_c) \cdot \cos(\pi i t)\, dt\,, i = 1,2,3,4 \quad (1)$$

in which $G_{t_c}(t, f_c)$ is the unscaled gammatone spectrogram for the channel centered at f_c, and the normalized range of t from -0.5 to 0.5 denotes the length of the temporal filters, with zero aligned with the middle frame at t_c. We used 4 temporal DCS filters, each spanning 50 frames. This is a filter length of 100 ms for the 2 ms frame spacing. Figure 2 (left) depicts these filters, with modulation frequencies 5, 10, 15 and 20 Hz respectively. These modulation frequencies approximately cover the range of the meaningful modulation frequencies of human auditory

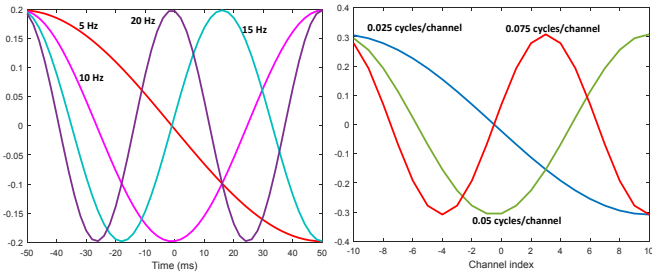

Fig.2. The 4 DCS temporal filters (left) and the 3 spectral filters (right) used in this work with modulation frequencies labelled.

systems from 2 to 22 Hz as reported in [18,19]. The 1-D convolutions are implemented by sliding the 4 filters through the entire $G(t, f)$ plane along the time axis, yielding 160 (40×4) temporal modulation features for each frame.

Three spectral modulation filters provided the best ASR performance experimentally, and thus were used in DCS. The spectral modulations are also computed over the original unscaled spectrogram $G(t, f)$. As shown in Figure 1, the spectral modulation is a separate block from the temporal modulation. This independent modulation requires only 7 (4+3) one-dimensional convolutions in total, which is highly computationally efficient. In a manner similar to that used in Eq. (1), the jth spectral DCS filter at (t_c, f_c) is defined by:

$$DCS_{S,j}(t_c, f_c) = \int_{-\frac{1}{2}}^{\frac{1}{2}} G_{f_c}(t_c, f) \cdot \cos(\pi j f)\, df\,, j = 1,2,3 \quad (2)$$

in which the normalized frequency range covers 20 channels centered at f_c. Figure 2 (right) depicts the spectral filters with modulation frequencies of 0.025, 0.05 and 0.075 cycles/channel.

Spectral downsampling is conducted in a similar way as in the Gabor filterbank (GBFB) [20] to reduce the correlations between the spectral modulation features. The 40 spectral modulation terms produced by each DCS are decimated by a factor of 4. The feature yielded by the central channel (1722 Hz) is always preserved because presumably this channel contains the richest spectral information. Totally 30 (10×3) spectral modulation terms are preserved, and thus there are 190 temporal and spectral modulation features (160+30) for each frame.

2.3. Peripheral processing and computational cost

It has been observed in previous studies that Histogram Equalization (HEQ) improves feature robustness [21,22]. In addition to this motivation for using HEQ, we also observed that the DCS modulation features are very non-Gaussian, as shown in Figure 3 (left), which degrades GMM modeling for HMMs. Thus, we implemented sentence-based HEQ to gaussianize each modulation term as shown in Figure 3 (right). In Figure 4, we plot the overall effect of the DCS (temporal) processing versus the regular delta features. It can be seen that DCS reduces the feature mismatch between clean and corrupted data.

Principal Component Analysis (PCA) is used to reduce the 190 modulation features to 32 uncorrelated terms, which are then appended to the 13 static features. The final total dimension of 45 features is practical for the HMMs. We used PCA because it

provides good performance, and is also relatively simple to implement compared with other neural-network-based methods as in [23,24,25]. The PCA is applied only to the modulation features, excluding the static gammatone spectrogram, which is processed by a separate DCT. Incorporating the entire feature set into PCA was found to degrade ASR performance, which might be due to the power-law scaling, which was used only for the unmodulated spectrogram.

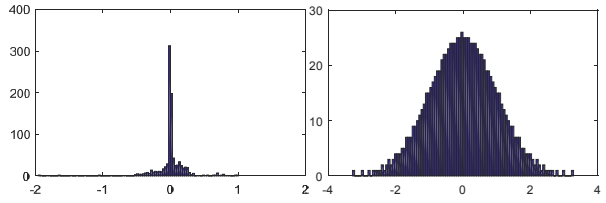

Fig.3. Histogram of the 10th modulated channel by the 4th temporal DCS filter before (left) and after (right) HEQ.

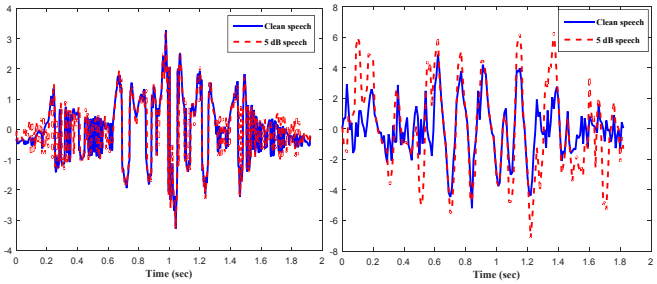

Fig.4. Gaussianized 10th gammatone channel modulated by the 4th temporal DCS (left) vs. regular delta over the 10th log-gammatone channel (right) in clean and 5 dB speech corrupted by white noise.

Note that in addition to modulations, the CMVN, HEQ and the PCA autocorrelation matrix are all based on a 2 ms frame spacing. This is because these peripheral operations benefit from the collection of slowly varying statistics on a fine time scale. In the last step, a temporal downsampling decimates the features by a factor of 6, resulting in a final feature spacing of 12 ms.

DCS has relatively low computational cost compared with long-frame-based methods. Although more frames are created in the beginning, since the filterbank and modulations are linear operations, the 2 ms frame spacing and decimation require less runtime than the time expensive long frame length followed by speech reconstruction and another pass of feature extraction. DCS uses only 7 disjoint modulation filters, all implemented by 1-D convolutions, which also makes the algorithm efficient.

3. EXPERIMENTAL EVALUATION

The Wall Street Journal (WSJ) SI-284 and Nov'92 (5K words) were used as training and testing sets respectively to obtain all the results reported in this section. We used the Kaldi toolkit [26] and the provided WSJ scripts which follow the pipeline of "triphone->feature splicing->LDA->MLLT->fMLLR (SAT) ->feature splicing->LDA->DNN" as recommended in [27]. The last LDA was used only for whitening the features. The DNN had 4 hidden layers each with 1024 hyperbolic tangent neurons. The minibatch size was fixed at 128, and the dev93 set was used to schedule the learning rate according to the frame accuracy. The feature splicing used a 9-frame window, and the DNN parameters were initialized according to a normal distribution. The 20K word trigram language model was used in decoding.

Table 1 lists the word accuracy of different components in DCS. The baseline used 39 MFCC features, including up to 2nd order deltas. The notation DCS-T and DCS-TS denote using only the temporal filters and both temporal and spectral filters respectively (45 features in both cases), with the numbers 2 and 10 denoting the frame spacing. The feature spacing in DCS-TS-10 was also 10 ms (no temporal decimation) and the modulation frequencies were not changed (that means the temporal filters spanned 10 frames, which were still 100 ms as in DCS-TS-2). The 39 GFCC features used the same power-law-scaled gammatone as in DCS, but with regular delta terms. In Table 1, the training data was clean. The white and babble noise were obtained from the NOISEX-92 database [28]. The street noise was recorded in a busy street, and the bar noise was recorded in a bar with many speakers and music. The music noise contains piano sounds. A random segment from the long noise recording was added to each test sentence scaled to different SNRs. The far-field RIRs (2 meters between speaker and microphone) for reverberation were retrieved from the REVERB CHALLENGE [29]. Three rooms with reverberation time T_{60}=0.25, 0.5 and 0.7 seconds were simulated.

From Table 1, the power-law-gammatone considerably improves the log-Mel used for the MFCC features. The temporal modulation provides another major improvement over the delta terms used in GFCC. The spectral modulation is not very effective for musical noise and reverberation compared with its use for other cases. The values in the parentheses are the percentage error rate reductions provided by 2 ms processing, relative to DCS-TS-10. The small frame spacing has relatively significant effects for both noise and reverberation, considering that the DCS-TS-10 already yields strong performance.

Table 2 shows the results for matched multi-condition training/testing. The training, development and evaluation sets were all renoised. For each sentence, a noise type was randomly (uniformly) selected with a SNR (or T_{60}) also randomly chosen from clean, 20, 15, 10, and 5 dB (or 0.25, 0.5 and 0.7 s for T_{60}). Table 2 has the same trend as observed in Table 1. Compared with the baseline MFCC result, the best DCS setting reduces the error rate by 25.9% relatively.

Table 3 evaluates DCS (DCS-TS-2) and several other advanced algorithms in mismatched clean training cases, and Table 4 repeats the experiments for matched multi-condition training/testing. We used the MATLAB code in [30] and [13] for RASTA-PLP and PNCC respectively without modifications. For DSCC, we replaced the Mel filterbank in the original code [11] with the same gammatone filterbank as in DCS, which improves the original DSCC. All the optimized values for the parameters proposed in the original works for these methods were used.

From these results, DCS outperforms other methods in most cases, except for several low SNR musical noise situations, for which RASTA performs better. Also, it's not easy to obtain large improvements for both additive noise and reverberation, as can be seen in the RASTA and PNCC results. RASTA is substantially better than PNCC and DSCC in additive noise, but it's not the case for reverberation. The temporal masking module in PNCC makes a large contribution to reduce reverberation, but

Table 1. Word accuracy (%) of the *WSJ Nov'92* for mismatched training/testing using different components of DCS.

SNR	MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
clean	94.2	94.5	94.7	**95.1** (2.0)	95.0
White 20 dB	92.3	93.0	94.4	**94.6** (6.9)	94.2
White 15 dB	87.1	91.3	92.7	**93.2** (11.7)	92.3
White 10 dB	70.0	86.3	90.2	**90.6** (11.3)	89.4
White 5 dB	30.6	69.9	83.9	**84.9** (11.7)	82.9
White 0 dB	15.0	36.2	65.3	**67.5** (10.7)	63.6

SNR	MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
clean	94.2	94.5	94.7	**95.1** (2.0)	95.0
Street 20 dB	93.2	94.2	94.5	**95.0** (5.7)	94.7
Street 15 dB	92.3	93.1	93.9	**94.8** (3.7)	94.6
Street 10 dB	89.6	90.6	93.0	**93.7** (6.0)	93.3
Street 5 dB	85.5	87.1	90.2	**91.3** (17.1)	89.5
Street 0 dB	76.3	79.7	82.8	**84.5** (17.1)	81.3

SNR	MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
Babble 20 dB	83.7	91.0	93.6	**93.9** (4.7)	93.6
Babble 15 dB	72.0	86.4	90.9	**92.5** (14.8)	91.2
Babble 10 dB	58.8	74.7	83.9	**87.1** (24.1)	83.0
Babble 5 dB	26.7	52.1	67.6	**72.4** (14.6)	67.7
Babble 0 dB	10.9	19.4	38.4	**42.0** (4.8)	39.1

SNR	MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
Bar 20 dB	89.0	92.2	93.8	**94.3** (5.0)	94.0
Bar 15 dB	81.5	89.3	92.0	**92.7** (9.9)	91.9
Bar 10 dB	71.4	83.6	87.6	**89.1** (8.4)	88.1
Bar 5 dB	49.6	65.3	76.7	**77.8** (5.5)	76.5
Bar 0 dB	15.0	30.8	46.3	**49.5** (8.3)	44.9

SNR	MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
Music 20 dB	81.9	90.1	91.2	**93.4** (9.6)	92.7
Music 15 dB	73.5	84.3	**91.4**	90.7 (10.6)	89.6
Music 10 dB	61.7	74.0	**85.7**	85.5 (12.1)	83.5
Music 5 dB	40.9	59.1	**76.9**	75.7 (11.0)	72.7
Music 0 dB	17.3	33.2	**54.4**	54.0 (4.0)	52.1

Reverb. Time	MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
T_{60}=0.25s	85.4	85.5	**90.0**	89.9 (9.8)	88.8
T_{60}=0.5s	35.3	47.3	**68.5**	68.4 (11.0)	64.5
T_{60}=0.7s	22.3	35.1	**61.3**	60.4 (13.7)	54.1

Table 3. Word accuracy (%) of the *WSJ Nov'92* for mismatched training/testing using different algorithms

SNR	MFCC	DSCC	RASTA-PLP	PNCC	DCS
clean	94.2	94.6	94.6	94.3	**95.1**
White 20 dB	92.3	92.7	93.9	93.3	**94.6**
White 15 dB	87.1	90.5	91.8	91.5	**93.2**
White 10 dB	70.0	83.0	87.6	87.5	**90.6**
White 5 dB	30.6	63.2	74.8	75.4	**84.9**
White 0 dB	15.0	20.3	42.4	46.5	**67.5**

SNR	MFCC	DSCC	RASTA-PLP	PNCC	DCS
clean	94.2	94.6	94.6	94.3	**95.1**
Street 20 dB	93.2	91.3	94.2	94.0	**95.0**
Street 15 dB	92.3	89.4	93.6	93.4	**94.8**
Street 10 dB	89.6	85.8	92.4	91.9	**93.7**
Street 5 dB	85.5	81.7	90.0	89.5	**91.3**
Street 0 dB	76.3	73.2	81.9	82.9	**84.5**

SNR	MFCC	DSCC	RASTA-PLP	PNCC	DCS
Babble 20 dB	83.7	86.9	93.3	92.8	**93.9**
Babble 15 dB	72.0	78.0	91.8	90.0	**92.5**
Babble 10 dB	58.8	63.4	87.1	82.4	**87.1**
Babble 5 dB	26.7	39.8	**73.5**	63.1	72.4
Babble 0 dB	10.9	16.0	31.2	31.2	**42.0**

SNR	MFCC	DSCC	RASTA-PLP	PNCC	DCS
Bar 20 dB	89.0	90.8	93.1	91.9	**94.3**
Bar 15 dB	81.5	85.9	91.4	90.3	**92.7**
Bar 10 dB	71.4	77.8	88.0	84.6	**89.1**
Bar 5 dB	49.6	58.9	76.3	68.3	**77.8**
Bar 0 dB	15.0	27.0	45.0	38.0	**49.5**

SNR	MFCC	DSCC	RASTA-PLP	PNCC	DCS
Music 20 dB	81.9	85.2	93.1	91.5	**93.4**
Music 15 dB	73.5	78.8	**92.0**	88.2	90.7
Music 10 dB	61.7	69.1	**89.6**	83.1	85.5
Music 5 dB	40.9	52.0	**80.8**	71.2	75.7
Music 0 dB	17.3	27.5	**59.5**	48.2	54.0

Reverb. Time	MFCC	DSCC	RASTA-PLP	PNCC	DCS
T_{60}=0.25s	85.4	85.7	85.9	84.9	**89.9**
T_{60}=0.5s	35.3	51.7	48.1	55.3	**68.4**
T_{60}=0.7s	22.3	39.7	37.9	45.5	**60.4**

Table 2. Word accuracy (%) of the *WSJ Nov'92* for matched multi-condition training/testing using different components of DCS.

MFCC	GFCC	DCS-T-2	DCS-TS-2	DCS-TS-10
88.8	90.0	90.9	**91.7** (6.7)	91.1

Table 4. Word accuracy (%) of the *WSJ Nov'92* for matched multi-condition training/testing using different algorithms.

MFCC	DSCC	RASTA-PLP	PNCC	DCS
88.8	90.0	90.9	89.1	**91.7**

Table 5. Word accuracy (%) of the *WSJ Nov'92* in reverberation (clean training) using GFCC and DCS combined with SSF.

Reverb Time	MFCC	SSF-GFCC	SSF-DCS
Clean	**94.2**	93.8	93.5
T_{60}=0.25s	85.4	89.2	**89.5**
T_{60}=0.5s	35.3	78.7	**80.8**
T_{60}=0.7s	22.3	74.0	**78.0**

PNCC does not approach RASTA in additive noise. DCS shows the best results for both noise and reverberation, with especially large improvements for reverberation.

Finally, DCS can be combined with other methods. In Table 5, the SSF algorithm in [31] was used to de-reverberate and resynthesize the speech, followed by DCS to extract features. In Table 6, the noise power normalization (PN) in PNCC was used to subtract the noise from the gammatone spectrogram in DCS. The "hybrid" DCS improves the original SSF and PNCC.

Table 6. Word accuracy (%) of the *WSJ Nov'92* (clean training) using power normalization as in PNCC combined with DCS.

SNR	MFCC	PNCC	PN-DCS
Clean	94.2	94.3	**94.3**
Babble 10 dB	58.8	82.4	**88.3**
Babble 5 dB	26.7	63.1	**75.1**
Babble 0 dB	10.9	31.2	**43.7**
Music 10 dB	61.7	83.1	**88.6**
Music 5 dB	40.9	71.2	**79.8**
Music 0 dB	17.3	48.2	**60.2**

4. CONCLUSIONS

A noise and reverberation robust DCS algorithm was proposed in this work. Temporal and spectral modulations are computed using only a small number of DCS filters based on a small frame spacing, which is an effective way to reduce the effects of slowly varying noise typically accounted for by long term frames. Future work includes removing DCT and PCA from DCS to let DNN learn speech patterns from the original features.

5. REFERENCES

[1] D. Yu, M. L. Seltzer, J. Li, J. T. Huang and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," in *Proc. Int. Conf. Learn. Represent.*, 2013.

[2] F. Seide, G. Li, X. Chen and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 24-29.

[3] C. Kim and R. M. Stern, "Power function-based power distribution normalization algorithm for robust speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2009, pp. 188-193.

[4] D. Gelbart and N. Morgan, "Double the trouble: Handling noise and reverberation in far-field automatic speech recognition," in *Proc. ICSLP*, Sep. 2002, pp. 2185-2188.

[5] D. Gelbart and N. Morgan, "Evaluating long-term spectral subtraction for reverberant ASR," in *Proc. ASRU*, Madonna di Campiglio, Italy, 2001, pp.103-106.

[6] K. Kinoshita, M. Delcroix, T. Nakatani, and M. Miyoshi, "Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction," in *IEEE transactions on audio, speech, and language processing*, vol.17, no.4, May 2009, pp. 1-12.

[7] M. Wu and D. Wang, "A two-stage algorithm for one-microphone reverberant speech enhancement," in *IEEE transactions on audio, speech and language processing*, vol. 14, no.3, May 2006, pp. 774-784.

[8] K. Kumar, R. Singh, B. Raj and R. M. Stern, "Gammatone sub-band magnitude-domain dereverberation for ASR," in *Proc. ICASSP*-2011, pp. 4604-4607.

[9] S. O. Sadjadi and J. H. L. Hansen, "Blind spectral weighting for robust speaker identification under reverberation mismatch," in *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 22, issue 5, May 2014, pp. 937-945.

[10] T. Yoshioka and M. J. F. Gales, "Environmentally robust ASR front-end for deep neural network acoustic models," in *Computer Speech and Language*, vol. 31, no. 1, May 2015, pp. 65-86.

[11] K. Kumar, C. Kim and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," in *Proc. ICASSP*-2011, pp. 4784-4787.

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

[13] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*-2012, pp. 4101-4104.

[14] S. A. Zahorian, H. Hu, Z. Chen, and J. Wu, "Spectral and temporal modulation features for phonetic recognition," in *Proc. INTERSPEECH-2009*, pp. 1071-1074.

[15] X. Zhao and D. Wang, "Analyzing noise robustness of MFCC and GFCC features in speaker identification," in *Proc. ICASSP*-2013, pp. 7204-7208.

[16] J. Qi, D. Wang, Y. Jiang and R. Liu, "Auditory features based on gammatone filters for robust speech recognition," in *Proc. ISCAS* 2013, pp. 305-308.

[17] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," in *Speech Commun*., vol. 25, 1998, pp. 133-147.

[18] N. Kanedera, T. Arai, H. Hermansky and M. Pavel, "On the relative importance of various components of the modulation spectrum for automatic speech recognition," in *Speech Commun*. vol.28, issue 1, May 1999, pp. 43-55.

[19] T. Chi, P. Ru and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," in *J. Acoust. Soc. Am*. vol. 118, no. 2, Aug. 2005, pp. 887-906.

[20] M. R. Schädler, B. T. Meyer and B. Kollmeier, "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," in *J. Acoust. Soc. Am*., vol. 131, issue 5, 2012, pp. 4134-4151.

[21] Á. de la Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," in *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, May 2005, pp. 355-366.

[22] X. Xiao, J. Li, E. S. Chng and H. Li, "Maximum likelihood adaptation of histogram equalization with constraint for robust speech recognition," in *Proc. ICASSP*-2011, pp. 5480-5483.

[23] T. Gramss, "Word recognition with the feature finding neural network (FFNN)," in *Proc. International Workshop Neural Networks Signal Process.*, 1991, pp. 289-298.

[24] H. Hermansky, D. P. W. Ellis and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*-2000, vol. 3, pp. 1635-1638.

[25] H. Lei, B. T. Meyer and N. Mirghafori, "Spectro-temporal Gabor features for speaker recognition," in *Proc. ICASSP*-2012, pp. 4241-4244.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer and K. Veselý, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Signal Processing Society, 2011.

[27] S. P. Rath, D. Povey, K. Veselý and J. H. Černocký, "Improved feature processing for deep neural networks," in *Proc. INTERSPEECH*-2013, pp. 109-113.

[28] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems," in *Speech Commun*., vol. 12, issue 3, July 1993, pp. 247-251.

[29] The room impulse responses and the MATLAB code used to create simulated reverberant databases in this work can be retrieved from the REVERB CHALLENGE website: http://reverb2014.dereverberation.com/data.html.

[30] D. Ellis. (2006) PLP and RASTA in MATLAB [Online]. Available: http://labrosa.ee.columbia.edu/matlab/rastamat/.

[31] C. Kim and R. M. Stern, "Nonlinear enhancement of onset for robust speech recognition," in *Proc. INTERSPEECH*-2010, pp. 2058-2061.