Power Function-Based Power Distribution Normalization Algorithm for Robust Speech Recognition

Chanwoo Kim[1] and Richard M. Stern[2]

Department of Electrical and Computer Engineering[2]
and Language Technologies Institute[1,2]
Carnegie Mellon University, Pittsburgh PA 15213 USA
[1] chanwook@cs.cmu.edu
[2] rms@cs.cmu.edu

***Abstract*—A novel algorithm that normalizes the distribution of spectral power coefficients is described in this paper. The algorithm, called power-function-based power distribution (PPDN) is based on the observation that the ratio of arithmetic mean to geometric mean changes as speech is corrupted by noise, and a parametric power function is used to equalize this ratio. We also observe that a longer "medium-duration" observation window (of approximately 100 ms) is better suited for parameter estimation for noise compensation than the briefer window typically used for automatic speech recognition. We also describe the implementation of an online version of PPDN based on exponentially weighted temporal averaging. Experimental results shows that PPDN provides comparable or slightly better results than state of- the-art algorithms such as vector Taylor series for speech recognition while requiring much less computation. Hence, the algorithm is suitable for both real-time speech communication or as a real-time preprocessing stage for speech recognition systems.**

Index Terms: Power distribution, equalization, ratio of arithmetic mean to geometric mean, medium-duration window

I. Introduction

Even though many speech recognition systems have provided satisfactory results in clean environments, one of the biggest problems in the field of speech recognition is that recognition accuracy degrades significantly if the test environment is different from the training environment. These environmental differences might be due to additive noise, channel distortion, acoustical differences between different speakers, etc. Many algorithms have been developed to enhance environmental robustness of speech recognition systems (*e.g.*[1], [2], [3], [4], [5], [6], [7], [8], [9]). Cepstral mean normalization (CMN) [10] and mean-variance Normalization (MVN) (*e.g.*[1]) are the simplest kinds of these techniques [11]. In these approaches, it is assumed that the mean or the mean and variance of the cepstral vectors should be the same for all utterances. These approaches are especially useful if the noise is stationary and its effect can be approximated by a linear function in the cepstral domain. Histogram Equalization (HEQ) (*e.g.* [2]) is a more powerful approach that assumes that the cepstral vectors of all the utterances have the same probability density function. Histogram normalization can be applied either in the waveform domain (*e.g.* [12]), the spectral domain (*e.g.* [13]), or the cepstral domain (*e.g.*[14]). Recently it has been observed that applying histogram normalization to delta cepstral vectors as well as the original cepstral vectors can also be helpful for robust speech recognition [2].

Even though many of these simple normalization algorithms have been applied successfully in the feature (or cepstral) domain rather than in the time or spectral domains, normalization in the power or spectral domain has some advantages. First, temporal or spectral normalization can be easily used as a preprocessing stage for any kinds of feature extraction systems and can be used in combination with other normalization schemes. In addition, these approaches can be also used as part of a speech enhancement scheme. In the present study, we perform normalization in the spectral domain, resynthesizing the signal using the inverse Fast Fourier Transform (IFFT) and combined with the overlap-add method (OLA).

One characteristic of speech signals is that their power level changes very rapidly while the background noise power usually changes more slowly. In the case of stationary noise such as white or pink noise, the variation of power approaches zero if the length of the analysis window becomes sufficiently large, so the power distribution is centered at a specific level. Even in the case of non-stationary noise like music noise, the noise power does not change as fast as the speech power. Because of this, the distribution of the power can be effectively used to determine the extent to which the current frame is affected by noise, and this information can be used for equalization. One effective way of doing this is measuring the ratio of arithmetic mean to geometric mean (*e.g.* [15]). This statistic is useful because if power values do not change much, the arithmetic and geometric mean will have similar values, but if there is a great deal of variation in power the arithmetic mean will be much larger than the geometric mean. This ratio is directly related to the shaping parameter of the gamma distribution, and it also has been used to estimate the signal-to-noise ratio (SNR) [16].

In this paper we introduce a new normalization algorithm based on the distribution of spectral power. We observe that the the ratio of the arithmetic mean to geometric mean of power in a particular frequency band (which we subsequently refer to as the *AM–GM ratio* in that band) depends on the amount of noise in the environment [15]. By using values of the AM–GM ratio obtained from a database of clean speech, a nonlinear transformation (specifically a power function) can be exploited to transform the output powers so that the AM–GM ratio in each frequency band of the input matches the corresponding ratio observed in the clean speech used for training the normalization system. In this fashion speech can re-synthesized resulting in greatly improved sound quality as well as better recognition results for noisy environments. In many applications such as voice communication or real-time speech recognition, we want the normalization to work in online pipelined fashion, processing speech in real time. In this paper we also introduce a method to find appropriate power coefficients in real time.

As we have observed in previous work [15], [17], even though windows of duration between 20 and 30 ms are optimal for speech analysis and feature extraction, longer-duration windows between 50 ms and 100 ms tend to be better for noise compensation. We also explore the effect of window length in power-distribution normalization and find the same tendency is be observed for this algorithm as well.

The rest of the paper is organized as follows: Sec. II describes our power-function-based power distribution normalization algorithm at a general level. We describe the online implementation of the normalization algorithm in Sec. III. Experimental results are discussed in Sec.IV and we summarize our work in Sec. V.

II. Power function based power distribution normalization algorithm

A. Structure of the system

Figure 1 shows the structure of our power-distribution normalization algorithm. The input speech signal is pre-emphasized and then multiplied by a medium duration (100-ms) Hamming window. This signal is represented by $x_i[n]$ in Fig. 1 where i denotes the frame index. We use a 100-ms window length and 10 ms between frames. The reason for using the longer window will be discussed later. After windowing, the FFT is computed and integrated over frequency using gammatone weighting functions to obtain the power $P(i,j)$ in the i^{th} frame and j^{th} frequency band as shown below:

$$P(i,j) = \sum_{k=0}^{N-1} |X(i,e^{j\omega_k}) H_j(e^{j\omega_k})|^2 \tag{1}$$

where k is a dummy variable representing the discrete frequency index, and N is the FFT size. The discrete frequency ω_k is defined by $\omega_k = \frac{2\pi k}{N}$. Since we are using a 100-ms window, for 16-kHz audio samples N is 2048. $H_j(e^{j\omega_k})$ is the spectrum of the gammatone filter bank for the j^{th} channel

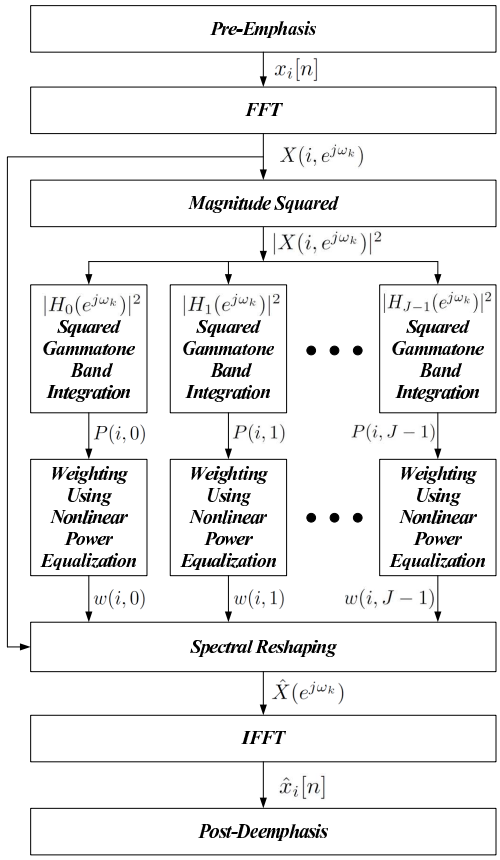

Fig. 1. The block diagram of the power-function-based power distribution normalization system.

evaluated at frequency index k, and $X(i,e^{j\omega_k})$ is the short-time spectrum of the speech signal for this i^{th} frame. J in Fig. 1 denotes the total number of gammatone channels, and we are using $J = 40$ for obtaining the spectral power. After power equalization, which will be explained in the following subsections, we perform spectral reshaping and compute the IFFT using OLA to obtain enhanced speech.

B. Normalization based on the AM–GM ratio

In this subsection, we examine how the frequency-dependent AM–GM ratio behaves. As describe previously, the AM–GM ratio of of $P(i,j)$ for each channel is given by the following equation:

$$g(j) = \frac{\frac{1}{I}\sum_{i=0}^{I-1} P(i,j)}{\left(\prod_{i=0}^{I-1} P(i,j)\right)^{\frac{1}{I}}} \tag{2}$$

where I represents the total number of frames. Since addition is easier to handle than multiplication and exponentiation to $1/I$, we will use the logarithm of the above ratio in the following discussion.

$$G(j) = \log\left(\sum_{i=0}^{I-1} P(i,j)\right) - \frac{1}{I}\sum_{i=0}^{I-1} \log P(i,j) \tag{3}$$

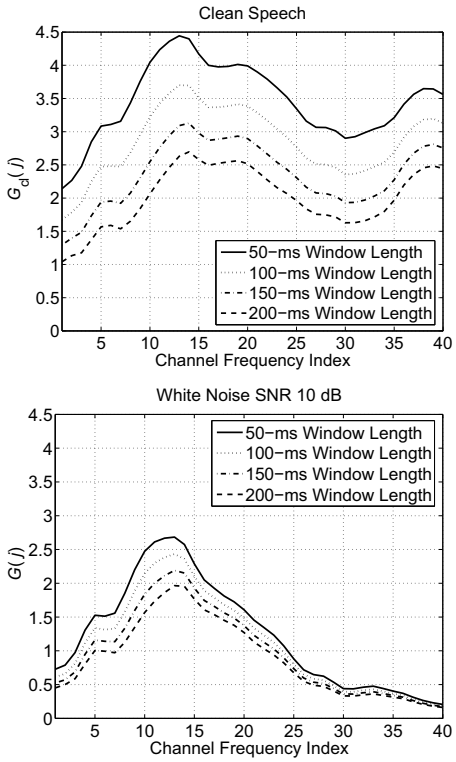

Fig. 2. The logarithm of the AM–GM ratio of spectral power of clean speech (upper panel) and of speech corrupted by 10-dB white noise (lower panel). Data were collected from 1,600 training utterances of the Resource Management database.

Figure 2 illustrates $G(j)$ for clean and noisy speech corrupted by 10-dB additive white noise. It can be seen that as noise is added the values of $G(j)$ generally decrease. We define the function $G_{cl}(j)$ to be the value of $G(j)$ obtained from clean training speech. We now proceed to normalize differences in $G(j)$ using a power function.

$$\tilde{P}_{cl}(i,j) = k_j P(i,j)^{a_j} \tag{4}$$

In the above equation, $P(i,j)$ is the medium-duration power of the noise-corrupted speech, and $\tilde{P}_{cl}(i,j)$ is the normalized medium-duration power. We want the AM–GM ratio representing normalized spectral power to be equal to the corresponding ratio at each frequency of the clean database. The power function is used because it is simple and the exponent can be easily estimated. We proceed to estimate k_j and a_j using this criterion.

Substituting $\tilde{P}_{cl}(i,j)$ into (3) and canceling out k_j, the ratio $\tilde{G}_{cl}(j|a_j)$ from this transformed variable $\tilde{P}_{cl}(i,j)$ can be represented by the following equation:

$$\tilde{G}_{cl}(j|a_j) = \log\left(\frac{1}{I}\sum_{i=0}^{I-1} P(i,j)^{a_j}\right) - \frac{1}{I}\sum_{i=0}^{I-1}\log P(i,j)^{a_j} \tag{5}$$

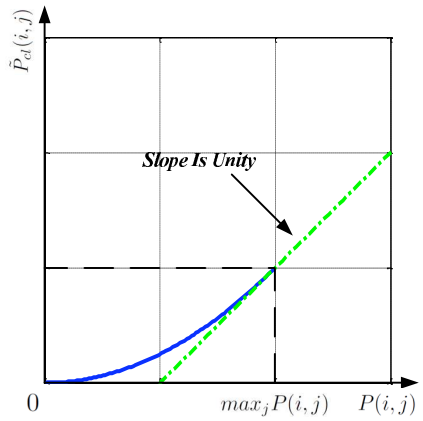

Fig. 3. The assumption about the relationship between $P_{cl}(i,j)$ and $P(i,j)$.

For a specific channel j, we see that a_j is the only unknown variable in $\tilde{G}_{cl}(j|a_j)$. Now, from the following equation:

$$\tilde{G}_{cl}(j|a_j) = G_{cl}(j) \tag{6}$$

we can obtain a value for a_j using the Newton-Raphson method.

The parameter k_j in Eq. (4) is obtained by assuming that the derivative of $\tilde{P}_{cl}(i,j)$ with respect to $P(i,j)$ is the unity at $max_i P(i,j)$ for this channel j, we set up the following constraint:

$$\left.\frac{d\tilde{P}_{cl}(i,j)}{dP(i,j)}\right|_{max_i P(i,j)} = 1 \tag{7}$$

The above constraint is illustrated in Fig 3. The meaning of the above equation is that the slope of the nonlinearity is unity for the largest power of the j^{th} channel. This constraint might look arbitrary, but it makes sense for additive noise case, since the following equation will hold:

$$P(i,j) = P_{cl}(i,j) + N(i,j) \tag{8}$$

where $P_{cl}(i,j)$ is the true clean speech power, and $N(i,j)$ is the noise power. By differentiating the above equation with respect to $P(i,j)$ we obtain:

$$\frac{dP_{cl}(i,j)}{dP(i,j)} = 1 - \frac{dN(i,j)}{dP(i,j)} \tag{9}$$

At the peak value of $P(i,j)$, the variation of $N(i,j)$ will be much smaller for a given variation of $P(i,j)$, which means that the variation of $P(i,j)$ around its largest value would be mainly due to variations of the speech power rather than the noise power. In other words, the second term on the right hand side of Eq. (9) would be very small, yielding Eq.(7). By substituting (7) into (4), we obtain a value for k_j:

$$k_j = \frac{1}{a_j}\max_i P(i,j)^{1-a_j} \tag{10}$$

Using the above equation with (4), we see that the weight for $P(i,j)$ is given by:

$$w(i,j) = \frac{\tilde{P}_{cl}(i,j)}{P(i,j)} = \frac{1}{a_j}\left(\frac{P(i,j)}{\max_i P(i,j)}\right)^{a_j-1} \quad (11)$$

After obtaining the weight $w(i,j)$ for each gammatone channel, we reshape the original spectrum $X(i,e^{j\omega_k})]$ using the following equation for the i^{th} frame:

$$\hat{X}(i,e^{j\omega_k}) = \sqrt{\sum_{j=0}^{J-1}(w(i,j)|H_j(e^{j\omega_k})|)^2}X(i,e^{j\omega_k}) \quad (12)$$

As mentioned before, $H_j(e^{j\omega_k})$ is the spectrum of the j^{th} channel of the gammatone filter bank, and J is the total number of channels. $\hat{X}(i,e^{j\omega_k})$ is the resultant enhanced spectrum. After doing this, we compute the IFFT of $\hat{X}(i,e^{j\omega_k})$ to retrieve the time-domain signal and perform de-emphasis to compensate for the effect of the previous pre-emphasis. The speech waveform is resynthesized using OLA.

C. Medium-duration windowing

Even though short-time windows of 20 to 30 ms duration are best for feature extraction for speech signals, in many applications we observe that longer windows are better for normalization purposes (*e.g.* [15] [17]). The reason for this is that noise power changes more slowly than the rapidly-varying speech signal. Hence, while good performance is obtained using short-duration windows for ASR, longer-duration windows are better for parameter estimation for noise compensation. Figure describes recognition accuracy as a function of window length. As can be seen in the figure a window of length between 75 ms and 100 ms provides the best parameter estimation for noise compensation and normalization. We will refer to a window of approximately this duration as a "medium-duration window".

III. ONLINE IMPLEMENTATION

In many applications the development of a real-time "online" algorithm for speech recognition and speech enhancement is desired. In this case we cannot use (5) for obtaining the coefficient a_j, since this equation requires the knowledge about the entire speech signal. In this section we discuss how an online algorithm of the power equalization algorithm can be implemented. To resolve this problem, we define two terms $S_1(i,j|a_j)$ and $S_2(i,j|a_j)$ with a forgetting factor λ of 0.9 as follows.

$$S_1(i,j|a_j) = \lambda S_1(i,j-1) + (1-\lambda)Q_i(j)^{a_j} \quad (13)$$

$$S_2(i,j|a_j) = \lambda S_2(i,j-1) + (1-\lambda)\ln Q_i(j)^{a_j} \quad (14)$$

$$a_j = 1,2,...,10$$

In our online algorithm, we calculate $S_1(i,j|a_j)$ and $S_2(i,j|a_j)$ for integer values of a_j in $1 \le a_j \le 10$ for each frame. From (5), we can define the online version of $G(j)$ using $S_1(i,j)$ and $S_2(i,j)$.

$$\tilde{G}_{cl}(i,j|a_j) = \log(S_1(i,j|a_j)) - S_2(i,j|a_j) \quad a_j = 1,2,..10 \quad (15)$$

Now, $\hat{a}(i,j)$ is defined as the solution to the equation:

$$\tilde{G}_{cl}(i,j|\hat{a}(i,j)) = G_{cl}(j) \quad (16)$$

Note that the solution would depend on time, so the estimated power coefficient $\hat{a}(i,j)$ is now a function of both the frame index and the channel. Since we are updating $G_{cl}(i,j|a_j)$ for each frame using integer values of a_j in $1 \le a_j \le 10$, we use linear interpolation of $\tilde{G}_{cl}(i,j|a_j)$ with respect to a_j to obtain the solution to (16). For estimating k_j using (10), we need to obtain the peak power. In the online version, we define the following online peak power $M(i,j)$.

$$M(i,j) = \max(\lambda M(i,j-1), P(i,j)) \quad (17)$$

$$Q(i,j) = \lambda Q(i,j-1) + (1-\lambda)M(i,j) \quad (18)$$

Instead of directly using $M(i,j)$, we use the smoothed online peak $Q(i,j)$. Using $Q(i,j)$ and $\hat{a}(i,j)$ with (11), we obtain:

$$w(i,j) = \frac{1}{\hat{a}(i,j)}\left(\frac{P(i,j)}{Q(i,j)}\right)^{\hat{a}(i,j)-1} \quad (19)$$

Using $w(i,j)$ in (12), we can normalize the spectrum and resynthesize speech using IFFT and OLA. In (17) and (18), we use the same λ of 0.9 as in (13) and (14). In our implementation, we use the first 10 frames for estimating the initial values of the $\hat{a}(i,j)$ and $Q(i,j)$, but after performing this initialization, no look-ahead buffer is used in processing the remaining speech.

Figure 5 depicts spectrograms of the original speech corrupted by various types of additive noise, and corresponding spectrograms of processed speech using the online PPDN explained in this section. As seen in 5(b), for additive Gaussian white noise, improvement is observable even at 0-dB SNR. For the 10-dB music and 5-dB street noise samples, which are more realistic, as shown in 5(d) and 5(f), we can clearly observe that processing provides improvement. In the next section, we present speech recognition results using the online PPDN algorithm.

IV. SIMULATION RESULTS OF THE ONLINE POWER EQUALIZATION ALGORITHM

In this section we describe experimental results obtained on the DARPA Resource Management (RM) database using the online processing as described in Section III. We first observe that the online PPDN algorithm improves the subjective quality of speech, as can be assessed by the reader by comparing processed and unprocessed speech in the demo package at http://www.cs.cmu.edu/~robust/archive/algorithms/PPDN_ASRU2009/DemoPackage.zip

For quantitative evaluation of PPDN we used 1,600 utterances from the DARPA Resource Management (RM) database for training and 600 utterances for testing. We used

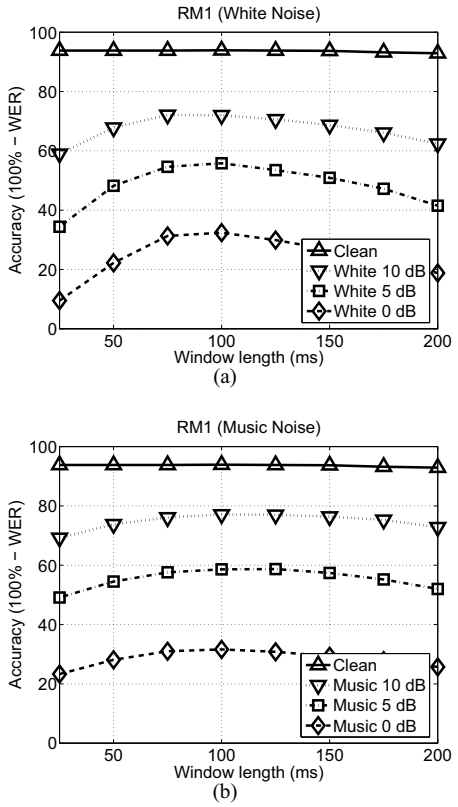

Fig. 4. Speech recognition accuracy as a function of window length for noise compensation corrupted by white noise and background music.

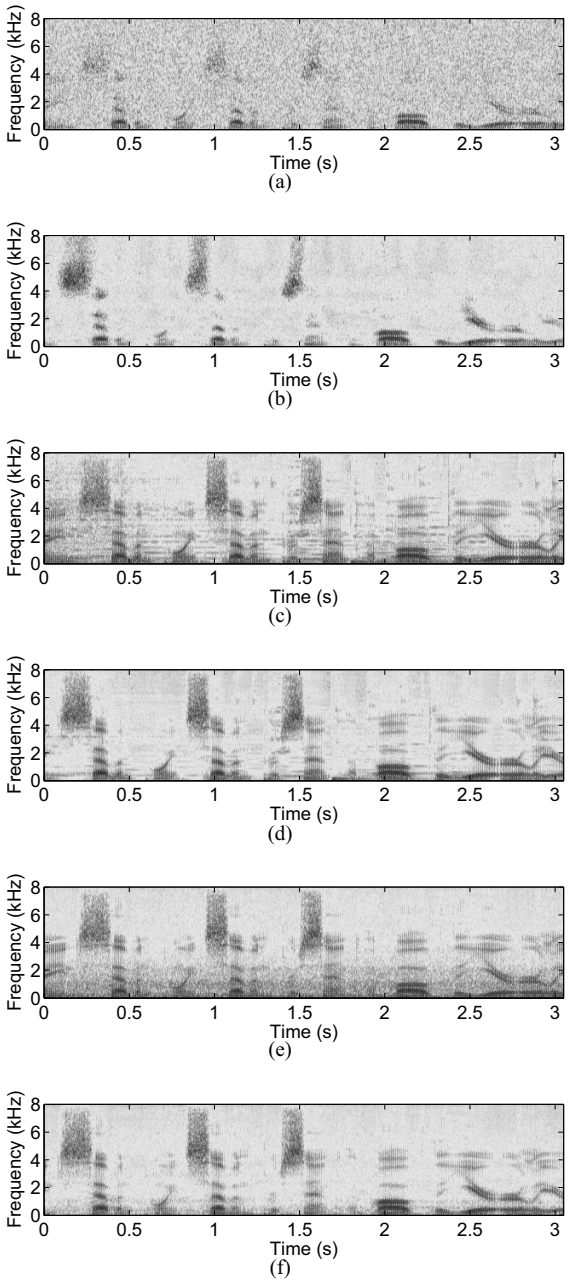

Fig. 5. Sample spectrograms illustrating the effects of online PPDN processing. (a) original speech corrupted by 0-dB additive white noise, (b) processed speech corrupted by 0-dB additive white noise (c) original speech corrupted by 10-dB additive background music (d) processed speech corrupted by 10-dB additive background (e) original speech corrupted by 5-dB street noise (f) processed speech corrupted by 5-dB street noise

`SphinxTrain 1.0` for training the acoustic models, and `Sphinx 3.8` for decoding. For feature extraction we used `sphinx_fe` which is included in `sphinxbase 0.4.1`. In Fig. 6(a), we used test utterances corrupted by additive white Gaussian noise, and in Fig. 6(b), noise recorded on a busy street was added to the test set. In Fig. 6(c) we used test utterances corrupted by musical segments of the DARPA Hub 4 Broadcast News database.

We prefer to characterize improvement as amount by which curves depicting WER as a function of SNR shift laterally when processing is applied. We refer to this statistic as the "threshold shift". As shown in these figures, PPDN provided 10-dB threshold shifts for white noise, 6.5-dB threshold shifts for street noise and 3.5-dB shifts for background music. Note that obtaining improvements for background music is not easy.

For comparison, we also obtained similar results using the state-of-the-art noise compensation algorithm Vector Taylor series (VTS) [3]. For PPDN, further application of Mean Variance Normalization (MVN) showed slightly better performance than applying CMN. However for VTS, we could not observe any performance improvement by applying MVN in addition, so we compared the MVN version of PPDN and the CMN version of VTS. For white noise, the PPDN algorithm outperforms VTS if the SNR is equal to or less than 5 dB, and the threshold shift is also larger. If the SNR is greater than or equal to 10 dB, VTS provides doing somewhat better recognition accuracy. In street noise, PPDN and VTS exhibited similar performance. For background music, which is considered to be more difficult, the PPDN algorithm produced threshold shifts of approximately 3.5 dB along with better accuracy than VTS for all SNRs.

A MATLAB implementation of the software used for these experiments is available at http://www.cs.cmu.edu/~robust/archive/algorithms/PPDN_ASRU2009/DemoPackage.zip.

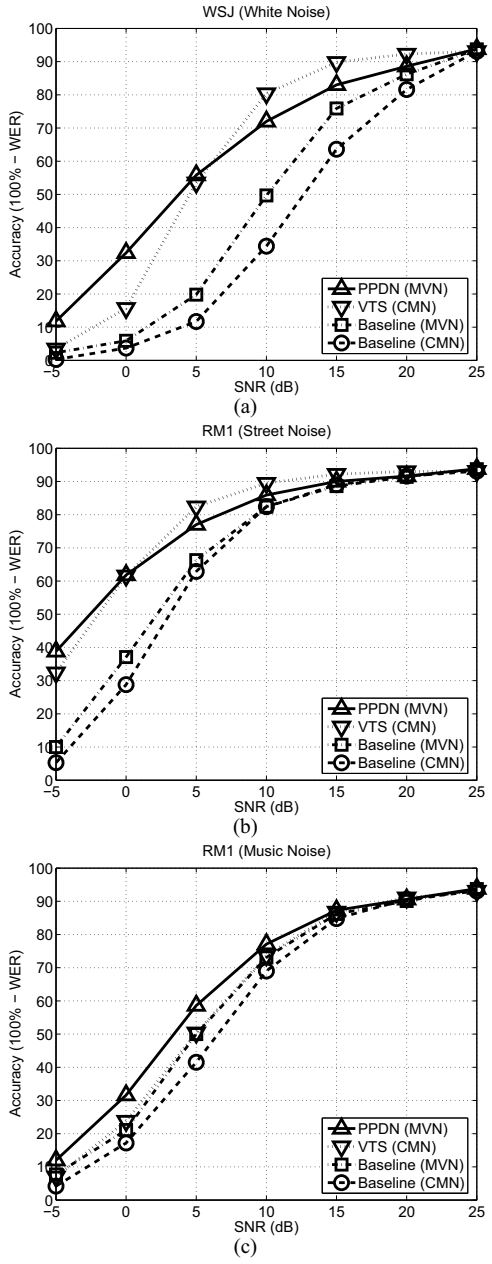

Fig. 6. Comparison of recognition accuracy for the DARPA RM database corrupted by (a) white noise, (b) street noise, and (c) music noise.

V. Conclusions

We describe a new power equalization algorithm, PPDN, that is based on applying a power function that normalizes the ratio of the arithmetic mean to the geometric mean of power in each frequency band. PPDN is simple and easier to implement than many other normalization algorithms. PPDN is quite effective against additive noise and provides comparable or somewhat better performance than the VTS algorithm. Since PPDN resynthesizes the speech waveform it can also be used for speech enhancement or as a pre-processing stage in conjunction with other algorithms that work in the cepstral domain. PPDN can also be implemented as an online algorithm without any lookahead buffer. This characteristic the algorithm potentially useful for applications such as real-time speech recognition or real-time speech enhancement. We also noted above that windows used to extract parametric information for noise compensation should be roughly 3 times the duration of those that are used for feature extraction. We used a window length of 100 ms for our normalization procedures.

VI. Acknowledgements

This research was supported by NSF (Grant IIS-0420866). The authors are thankful to Prof. Suryakanth Gangashetty for helpful discussions.

References

[1] P. Jain and H. Hermansky, "Improved mean and variance normalization for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*.

[2] Y. Obuchi,N. Hataoka, and R. M. Stern, "Normalization of time-derivative parameters for robust speech recognition in small devices," *IEICE Transactions on Information and Systems*, vol. 87-D, no. 4, pp. 1004–1011, Apr. 2004.

[3] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, May. 1996.

[4] C. Kim, Y.-H. Chiu, and R. M. Stern, "Physiologically-motivated synchrony-based processing for robust automatic speech recognition," in *INTERSPEECH-2006*, Sept. 2006, pp. 1975–1978.

[5] B. Raj and R. M. Stern, "Missing-Feature Methods for Robust Automatic Speech Recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 101–116, Sept. 2005.

[6] B. Raj, M. L. Seltzer, and R. M. Stern, "Reconstruction of Missing Features for Robust Speech Recognition," *Speech Communication*, vol. 43, no. 4, pp. 275–296, Sept. 2004.

[7] R. M. Stern, B. Raj, and P. J. Moreno, "Compensation for environmental degradation in automatic speech recognition," in *Proc. of the ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, Apr. 1997.

[8] R. Singh, B. Raj, and R. M. Stern, "Model compensation and matched condition methods for robust speech recognition," in *Noise Reduction in Speech Applications*, G. M. Davis, Ed. CRC Press, 2002, pp. 245–275.

[9] R. Singh, R. M. Stern, and B. Raj, "Signal and feature compensation methods for robust speech recognition," in *Noise Reduction in Speech Applications*, G. M. Davis, Ed. CRC Press, 2002, pp. 219–244.

[10] B. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *Journal of the Acoustical Society of America*, vol. 55.

[11] X. Huang, A. Acero, H-W Won, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Upper Saddle River, NJ: Prentice Hall, 2001.

[12] R. Balchandran and R. Mammone, "Non-parametric estimation and correction of non-linear distortion in speech system," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, May. 1998.

[13] S. Molau, M. Pitz, and H. Ney, "Histogram based normalization in the acoustic feature space," in *Proc. of Automatic Speech Recognition*, Nov. 2001.

[14] S. Dharanipragada and M. Padmanabhan, "A nonlinear unsupervised adaptation technique for speech recognition," in *Proc. Int Conf. Spoken Language Processing*, Oct. 2001.

[15] C. Kim and R. M. Stern, "Feature extraction for robust speech recognition using a power-law nonlinearity and power-bias subtraction," in *INTERSPEECH-2009*, Sept. 2009.

[16] ——, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *INTERSPEECH-2008*, Sept. 2008, pp. 2598–2601.

[17] C. Kim, K. Kumar, B. Raj, and R. M. Stern, "Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain," in *INTERSPEECH-2009*, Sept. 2009.