Robust Speech Recognition using a Small Power Boosting Algorithm

Chanwoo Kim [1], Kshitiz Kumar [2], and Richard M. Stern [3]

Department of Electrical and Computer Engineering [2,3]
and Language Technologies Institute [1,3]
Carnegie Mellon University, Pittsburgh PA 15213 USA
[1] chanwook@cs.cmu.edu
[2] kshitizk@ece.cmu.edu
[3] rms@cs.cmu.edu

***Abstract*—In this paper, we present a noise robustness algorithm called Small Power Boosting (SPB). We observe that in the spectral domain, time-frequency bins with smaller power are more affected by additive noise. The conventional way of handling this problem is estimating the noise from the test utterance and doing normalization or subtraction. In our work, in contrast, we intentionally boost the power of time-frequency bins with small energy for both the training and testing datasets. Since time-frequency bins with small power no longer exist after this power boosting, the spectral distortion between the clean and corrupt test sets becomes reduced. This type of small power boosting is also highly related to physiological nonlinearity. We observe that when small power boosting is done, suitable weighting smoothing becomes highly important. Our experimental results indicate that this simple idea is very helpful for very difficult noisy environments such as corruption by background music.**

Index Terms: Robust speech recognition, physiological modeling, rate-level curve, weight smoothing

I. Introduction

The performance of speech recognition systems in clean environments has improved impressively in the decades following the introduction of statistical modeling based on Hidden Markov Models (HMMs) [1] (*e.g.*[2]). Nevertheless, obtaining good performance in environments that are different from the training environment remains a challenging problem. Environmental differences include additive noise, channel distortion, speaker differences, and so on. Many algorithms have been proposed to deal with this problem which show significant improvement in performance for quasi-stationary noise (*e.g.* [3], [4], [5], [6], [7]). Unfortunately these same algorithms frequently do not show significant improvements in more difficult transitory environments such as background music (*e.g.* [8]).

Recent studies show that for non-stationary disturbances such as background music or background speech, algorithms based on missing features (*e.g.* [9], [10]) or auditory processing are more promising (*e.g* [11], [12], [13], [14]). Still, the improvement in non-stationary noise remains less than the improvement that is observed in stationary noise. In previous work [12], we also observed that the "threshold point" of the auditory nonlinearity plays an important role in in improving performance in additive noise. Let us imagine a specific time-frequency bin with small power. Even if a relatively small distortion is applied to this time-frequency bin, due to the nature of the compressive nonlinearity the distortion can become quite large.

In this paper, we explain the structure of the small boosting (SPB) algorithm in two different ways. In the first approach, we apply small power boosting to each time-frequency bin in the spectral domain, and then resynthesize speech (SPB-R). The resynthesized speech is fed to the feature extraction system. This approach is conceptually straightforward but less computationally efficient (because of the number of FFTs and IFFTs that must be performed). In the second approach, we use SPB to obtain feature values directly (SPB-D). This approach does not require IFFT operations and the system is consequently more compact. As we will discuss below, effective implementation of SPB-D requires smoothing in the spectral domain.

II. The Principle of Small Power Boosting

Before presenting the structure of the SPB algorithm, we first review how we obtain spectral power in our system, which is similar to the system in [15]. Pre-emphasis in the form of $H(z) = 1 - 0.97z^{-1}$ is applied to an incoming speech signal sampled at 16 kHz. A short-time Fourier transform (STFT) is calculated using Hamming windows of duration of 25.6 ms. Spectral power is obtained by integrating the magnitudes of the STFT coefficients over a series of weighting functions [16]. This procedure is represented by the following equation:

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

In the above equation i and j represent the frame and channel indices respectively, N is the FFT size, and $H_j(e^{jw_k})$ is the frequency response of the j^{th} Gammatone channel. $X(i;e^{j\omega_k})$ is the STFT for the i^{th} frame. w_k is defined by $\omega_k = \frac{2\pi k}{N}, \quad 0 \le k \le N-1$.

In Fig. 1(a), we can observe the distributions of $\log(P(i,j))$ for clean speech, speech in 0-dB music, and speech in 0-dB

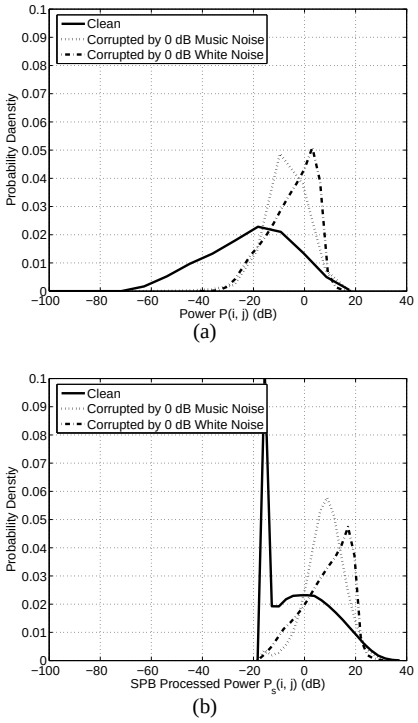

Fig. 1. Comparison of the Probability Density Functions (PDFs) obtained in three different environments : clean speech, 0-dB additive background music, and 0-dB additive white noise. Probability density functions (PDFs) were obtained using the conventional log nonlinearity (upper panel) and using SPB with a power boosting coefficient of 0.02.

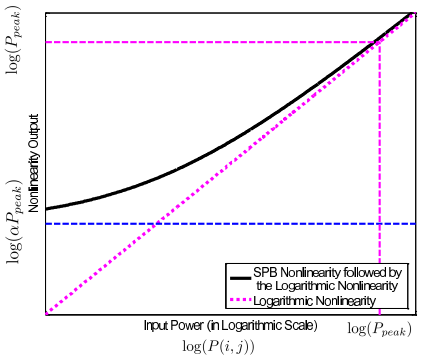

Fig. 2. The total nonlinearity consists of small power boosting and the subsequent logarithmic nonlinearity in the SPB algorithm

white noise. We used a subset of 50 utterances to obtain these distributions from the training portion of the DARPA Resource Management 1 (RM1) database. In plotting the distributions, we scaled each waveform to set the 95 percentile of $P(i,j)$ to be 0 dB. We note in Fig. 1(a) that higher values of $P(i,j)$ are (unsurprisingly) less affected by the additive noise, but the values that are small in power are severely distorted by additive noise. While the conventional approach to this problem is spectral subtraction (*e.g.* [17]), this goal can also be achieved by intentionally boosting power for all utterances, thereby rendering the small-power regions less affected by the additive noise. We implement the SPB algorithm with the following nonlinearity:

$$P_s(i,j) \;=\; \sqrt{P(i,j)^2 + (\alpha P_{peak})^2} \qquad (2)$$

Using the above equation, if $P(i,j) >> \alpha P_{peak}$, then we obtain $P_s(i,j) \approx P(i,j)$. For small power region (if $P(i,j) << \alpha P_{peak}$), we obtain $P_s(i,j) \approx \alpha P_{peak}$.

We will call α the "small power boosting coefficient" or "SPB coefficient". P_{peak} is defined to be the 95 percentile in the distribution of $P(i,j)$. In our algorithm, further explained in Section III and III, after obtaining $P_s(i,j)$, either resynthesis or smoothing is done. After that, the logarithmic nonlinearity is followed. Thus, if we plot the entire nonlinearity defined by (2) and the subsequent logarithmic nonlinearity, then is represented by Fig. 2. Suppose that the power of clean speech at a specific time-frequency bin $P(i,j)$ is corrupted by additive noise ν. The log spectral distortion is represented by the following equation:

$$\begin{aligned} d(i,j) &= \log(P(i,j)+\nu) - \log(P(i,j)) \\ &= \log\left(1 + \frac{1}{\eta(i,j)}\right) \end{aligned} \qquad (3)$$

where $\eta(i,j)$ is the Signal-to-Noise Ratio (SNR) for this time-frequency bin defined by:

$$\eta(i,j) = \frac{P(i,j)}{\nu} \qquad (4)$$

Applying the nonlinearity of (2) and the logarithmic nonlinearity, the remaining distortion is represented by:

$$\begin{aligned} d_s(i,j) &= \log(P_s(i,j)+\nu) - \log(P_s(i,j)) \\ &= \log\left(1 + \frac{1}{\sqrt{\eta(i,j)^2 + \left(\frac{\alpha P_{peak}}{\nu}\right)^2}}\right) \end{aligned} \qquad (5)$$

The largest difference between $d(i,j)$ and $d_s(i,j)$ occurs when $\eta(i,j)$is relatively small. For small power regions even if ν is not large, $\eta(i,j)$ will become relatively large, and in (3), the distortion will diverge to infinity as $\eta(i,j)$ approaches zero. In contrast, in (5), even if $\eta(i,j)$ approaches zero, the distortion converges to $\log\left(1+\frac{\nu}{\alpha P}\right)$.

Consider now the power distribution for SPB-processed powers. Fig. 1(b) compares the distributions for the same condition as Fig. 1(a). We can clearly see that the distortion is greatly reduced. As can be seen, SPB reduces the spectral distortion and provides robustness to additive noise. However, as described in our previous paper [15], all nonlinearities motivated by human auditory processing such as the "S"-shaped nonlinearity and the power-law nonlinearity curves also use this characteristic, but these approaches are less effective than the SPB approach described in the paper. The key difference, though, is that in other approaches, the nonlinearity is directly applied for each time-frequency bin. As will be discussed in Section IV, directly applying the non-linearity results in reduced variance for regions of small power, thus reducing the ability to discriminate small differences in power

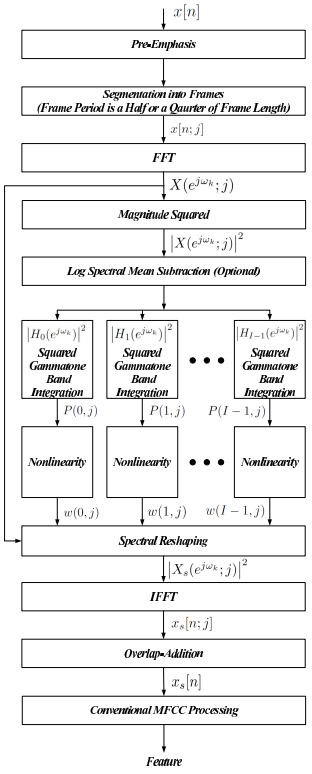

Fig. 3. Small power boosting algorithm which resynthesizes speech (SPB-R). Conventional MFCC processing is followed after resynthesizing the speech.

and finally to differentiate speech sounds. We explain this issue in detail in Section IV.

III. Small Power Boosting with Re-synthesized Speech (SPB-R)

In this section, we discuss the SPB system which resynthesizes speech as an intermediate stage in feature extraction. The entire block-diagram for this approach is shown in Fig. 3. The blocks leading up to *Overlap-Addition* (OLA) are for small power boosting and resynthesizing speech, which is finally fed to conventional feature extraction. The only difference between the conventional MFCC features and our features is the use of the gammatone-shaped frequency integration with the equivalent rectangular bandwidth (ERB) scale [18] instead of the triangular integration with with the MEL scale [19]. The advantages of gammatone-integration are described in [15], where gammatone-based integration was found to be more helpful in additive noise environments. In our system we use an ERB scale with 40 channels spaced between 130 Hz and 6800 Hz. From (2), the weighting coefficient $w(i,j)$ for each time-frequency is bin is given by:

$$w(i,j) = \frac{P_s(i,j)}{P(i,j)} = \sqrt{1 + \left(\frac{\alpha P_{peak}}{P(i,j)}\right)^2} \tag{6}$$

Using $w(i,j)$, we apply the spectral reshaping expressed in [20]:

$$\mu_g(i,k) = \frac{\sum_{j=0}^{J-1} w(i,j) \left|H_j\left(e^{j\omega_k}\right)\right|}{\sum_{j=0}^{J-1} \left|H_j\left(e^{j\omega_k}\right)\right|} \tag{7}$$

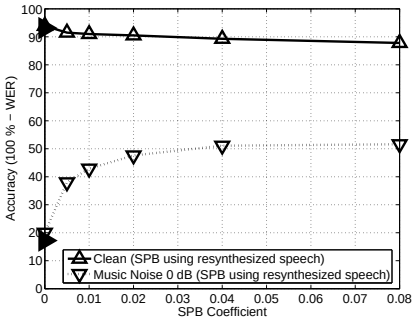

Fig. 4. Word error rates obtained using the SPB-R algorithm as a function of the value of the SPB Coefficient. The filled triangles at the y-axis represent the baseline MFCC performance for clean speech (upper triangle) and for additive background music noise at 0 dB SNR (lower triangle), respectively.

where I is the total number of channels, and k is a dummy index for discrete frequency. The reconstructed spectrum is obtained from the original spectrum $X\left(i;e^{j\omega_k}\right)$ by using $\mu_g(i,k)$ in (7) as follows:

$$X_s\left(i;e^{j\omega_k}\right) = \mu_g(i,k) X\left(i,e^{j\omega_k}\right) \tag{8}$$

Speech is resynthesized using $X_s\left(i;e^{j\omega_k}\right)$ by performing IFFT and using OLA with hamming windows of 25 ms duration and 6.25 ms intervals between adjacent frames, which satisfy the OLA constraint for undistorted reconstruction. Fig. 4 plots the WER against the SPB coefficient α. Experimental configuration is as described in Section VI. As can be seen in that figure, increasing the boosting coefficient results in much better performance for highly non-stationary noise even at 0 dB SNR, meanwhile losing a small performance for clean. Based on that trade-off between the clean and noisy performance, we may select the SPB coefficient α in 0.01-0.02.

IV. Small Power Boosting with Direct Feature Generation (SPB-D)

In the previous section we discussed the SPB-R system which resynthesizes speech as an intermediate step. Because resynthesizing the speech is quite computationally costly, we discuss in this section an alternate approach that generates SPB-processed features without the resynthesis step. A direct approach towards that end would be to simply apply the Discrete Cosine Transform (DCT) to the SPB-processed power $P_s(i,j)$ terms in (2). Since this direct approach is basically a feature extraction system itself, it will of course require that the window length and frame period used for segmentation into frames for SPB processing be the same values as are used in conventional feature extraction. Hence we use a window length of 25.6 ms with 10 ms between successive frames. We refer to this direct system as Small Power Boosting with Direct Feature Generation (SPB-D), and it is illustrated in Fig. 5.

Comparing the WER corresponding to $M = 0$ and $N = 0$ in Fig. 6 to the performance of SPB-R in Fig. 4, it is easily observed that SPB-D in the original form described above performs far worse than the SPB-R algorithm. These differences in performance are reflected in the corresponding

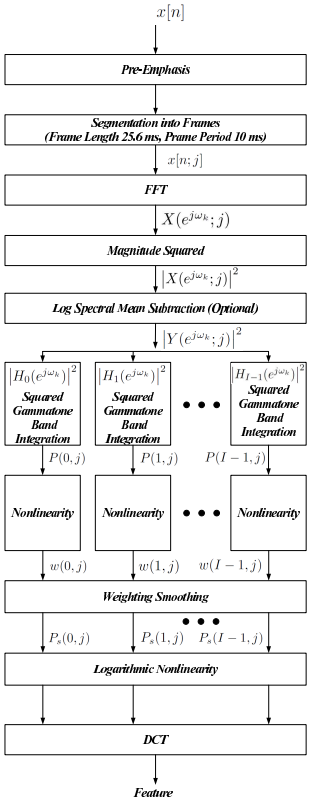

Fig. 5. Small power boosting algorithm with direct feature generation (SPB-D)

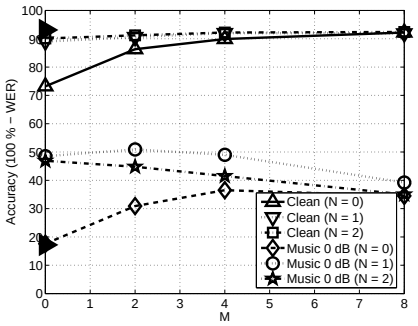

Fig. 6. The effects of weight smoothing on performance of the SPB-D algorithm for clean speech for speech corrupted by additive background music at 0 dB. The filled triangles at the y-axis represent the baseline MFCC performance for clean (upper triangle) and 0 dB additive background music (lower triangle) respectively. The SPB coefficient α was 0.02.

spectrograms, as can be seen by comparing Fig. 7(c) to the SPB-R-derived spectrogram in Fig. 7(b)). In Fig. 7(c), the variance in small power regions is very small (concentrated at αP_{peak} in Fig. 2 and (2)), thus losing the power to discriminate sounds which have small power. Small variance is harmful in this context because PDFs in the training data will be modeled by Gaussians with very narrow peaks. As a consequence small perturbation in the feature values from their means lead to large changes in log-likelihood scores. Hence we should avoid variances that are too small in magnitude.

We also note that there exist large overlaps in the shape

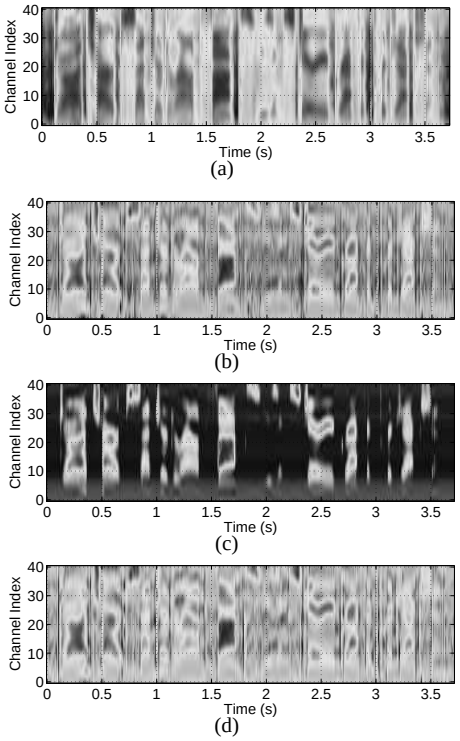

Fig. 7. Spectrograms obtained from a clean speech utterance using different processing: (a) conventional MFCC processing, (b) SPB-R processing, (c) SPB-D processing without any weight smoothing, and (d) SPB-D processing with weight smoothing $M = 4, N = 1$ in (9). A value of 0.02 was used for the SPB coefficient α. (2)

of gammatone-like frequency responses, as well as an overlap between successive frames. Thus, the gain in one time-frequency bin is correlated with that in an adjacent time-frequency bin. In the SPB-R approach, similar smoothing was achieved implicitly by the spectral reshaping from (7) and (8), and in the OLA process. With the SPB-D approach the spectral values must be smoothed explicitly.

Smoothing of the weights can be done horizontally (along time) as well as vertically (along frequency). These smoothed weights are obtained by:

$$\tilde{w}(i,j) = \exp\left(\frac{\sum_{i'=i-M}^{i+M}\sum_{j'=j-N}^{j+N}\log\left(w(i',j')\right)}{(2M+1)(2N+1)}\right) \qquad (9)$$

where, M and N respectively indicate smoothing along the time and frequency axes. The averaging in (9) is performed in the logarithmic domain (equivalent to geometric averaging) since the dynamic range of $w(i,j)$ is very large. (If we had performed a normal arithmetic averaging instead of geometric averaging in (9), the resulting averages would be dominated inappropriately by the values of $w(i,j)$ of greatest magnitude.)

Results of speech recognition experiments using different values of M and N are reported in Fig. 6. The experimental configuration is the same as was used for the data shown in Fig. 4. We note that the smoothing operation is quite helpful, and that with suitable smoothing the SBP-D algorithm works as well as the SPB-R. In our subsequent experiments, we used values of $N = 1$ and $M = 4$ in the SPB-D algorithm

with 40 gammatone channels. The corresponding spectrogram obtained with this smoothing is shown in Fig. 7(d), which is similar to that obtained using SPB-R in Fig. 7(b).

V. LOG SPECTRAL MEAN SUBTRACTION

In this section, we discuss log spectral mean subtraction (LSMS) as an optional pre-processing step in the SPB approach and we compare the performance between LSMS computed for each frequency index and LSMS computed for each gammatone channel. LSMS is a standard technique which has been commonly applied for robustness to environmental mismatch (*e.g.* [21]), and this technique is mathematically equivalent to the well known cepstral mean normalization (CMN) procedure. Log spectral mean subtraction is commonly performed for $\log\left(P(i,j)\right)$ for each channel j as shown below.

$$\tilde{P}(i,j) = \frac{P(i,j)}{\exp(\frac{1}{2L+1}\sum_{i'=i-L}^{i+L}\log\left(P(i',j)\right))} \tag{10}$$

Hence, this normalization is performed between the squared gammatone integration in each band and the nonlinearity. It is also reasonable to apply LSMS for $\log\left(X(i;e^{j\omega_k})\right)$ for each frequency index k before performing the gammatone frequency integration. This can be expressed as:

$$\tilde{X}(i;e^{j\omega_k}) = \frac{\left|X(i;e^{j\omega_k})\right|}{\exp(\frac{1}{2L+1}\sum_{i'=i-L}^{i+L}\log\left(|X(i';e^{j\omega_k})|\right))} \tag{11}$$

Fig. 8 depicts the results of speech recognition experiments using the two different approaches to LSMS (without including SPB). In that figure, the moving average window length indicates the length corresponding to $2L+1$ in (10) and (11). We note that the approach in (10) provides slightly better performance for white noise, but that the performance difference diminishes as the window length increases. However, the LSMS based on (11) shows consistently better performance in the presence of background music, which is consistent across all window lengths. This may be explained due to the rich discrete harmonic components in music, which makes frequency-index-based LSMS more effective. In the next section we examine the performance obtained when LSMS as described by (11) is used in combination with SPB.

VI. EXPERIMENTAL RESULTS

In this section we present experimental results using the SPB-R algorithm described in Section III and the SPB-D algorithm described in Section IV. We also examine the performance of SPB is combination with LSMS as described in Section V. We conducted speech recognition experiments using the CMU `Sphinx 3.8` system with `Sphinxbase 0.4.1`. For training the acoustic model, we used `SphinxTrain 1.0`. For the baseline MFCC feature, we used `sphinx_fe` included in `Sphinxbase 0.4.1`. All experiments in this and previous sections were conducted under identical condition, with delta and delta-delta components appended to the original features. For training and testing we used subsets of 1600 utterances and 600 utterances respectively from the DARPA Resource Management (RM1) database. To evaluate

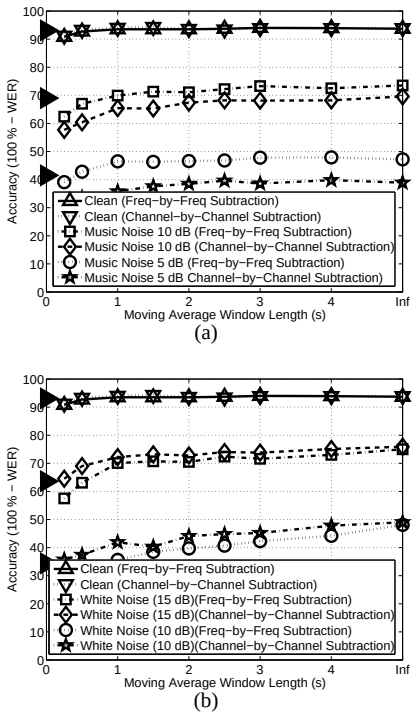

Fig. 8. The effect of Log Spectral Subtraction for (a) background music and (b) white noise as a function of the moving window length. The filled triangles at the y-axis represent baseline MFCC performance.

the robustness of the feature extraction approaches we digitally added white Gaussian noise and background music noise. The background music was obtained from musical segments of the DARPA HUB 4 database.

In Fig. 9, SPB-D is the basic SPB system described in Section IV. While we noted in a previous paper [20] that gammatone frequency integration is provides better performance than conventional triangular frequency integration the effect is minor in these results. Thus, the performance boost of SPB-D over the baseline MFCC is largely due to the SPB nonlinearity in (2) and subsequent gain smoothing. SPB-D-LSMS refers to the combination of the SPB-D and LSMS techniques. For both the SPB-D and SPB-D-LSMS systems we used a window length of 25.6 ms with $10ms$ between adjacent frames. Even though not explicitly plotted in this figure, SPB-R shows nearly the same performance as SPB-D as mentioned in IV and shown in Fig. 4.

We prefer to characterize improvement in recognition accuracy by the amount of lateral threshold shift provided by the processing. For white noise, SPB-D and SPB-D-LSMS provides an improvement of about 7 dB to 8 dB compared to MFCC, as shown in Fig. 9. SPB-R-LSMS results in slightly smaller threshold shift. For comparison, we also conduct experiments using the Vector Taylor Series (VTS) algorithm [4], as shown in Fig. 9. For white noise, the performance of SPB family is slightly worse than that obtained using VTS.

Compensation for the effects of music noise, on the other hand, is considered to be much more difficult (*e.g.* [8]). The SPB family of algorithms provides a very impressive improvement in performance with background music. An

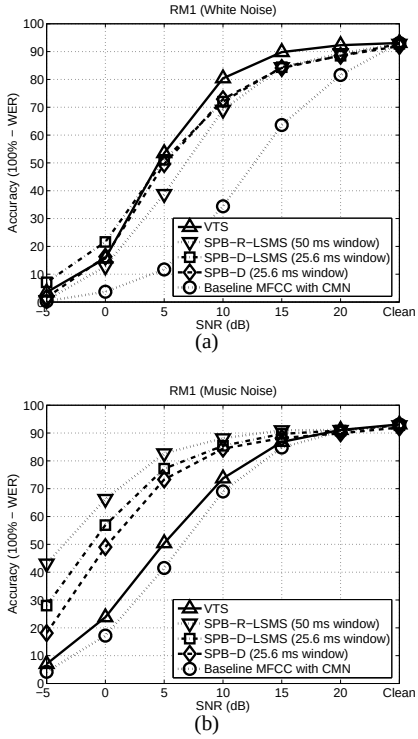

Fig. 9. Comparison of recognition accuracy between VTS, SPB-CW and MFCC processing: (a) additive white noise, (b) background music.

implementation of SPB-R-LSMS with window durations of 50 ms provides the greatest threshold shift (amounting to about 10 dB), and SPB-D provides a threshold shift of around 7 dB. VTS provides a performance improvement of about 1 dB for the same data.

Open Source MATLAB code for SPB-R and SPB-D can be found at http://www.cs.cmu.edu/~robust/archive/algorithms/SPB_ASRU2009. The code in this directory was used for obtaining the results in this paper.

VII. Conclusion

In this paper, we presented a robust speech recognition algorithm named Small Power Boosting (SPB), which is very helpful for difficult noise environment such as music noise. Our contribution is summarized in the following. First, we examine the PDFs obtained from clean and noisy environments, and observe that small power region is most vulnerable to noise. Based on the observation, we intentionally boost the small power region. We also noted that we should not boost power in each time-frequency bin independently as adjacent time-frequency bins are highly correlated. This can be achieved implicitly in SPB-R and by applying weighting smoothing in SPB-D. We also observed that directly applying nonlinearity results in too small variance for small power regions, which is harmful for robustness and speech sound discrimination. Finally, we also observe that for music noise LSMS for each frequency index is more helpful than doing this for each channel index.

VIII. Acknowledgements

This research was supported by NSF (Grant IIS-0420866). The authors are thankful to Prof. Bhiksha Raj for helpful advice, and to Hiroyuki Segi for helpful discussions.

References

[1] J. K Baker, "The dragon system - An overview," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 23, no. 1, pp. 24–29, Feb. 1975.

[2] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of IEEE*, vol. 64, no. 4, pp. 532–556, Apr. 1976.

[3] A. Acero, and R. M. Stern, "Environmental Robustness in Automatic Speech Recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (Albuquerque, NM)*, vol. 2, Apr. 1990, pp. 849–852.

[4] 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.

[5] P. Pujol, D. Macho, and C. Nadeu, "On real-time mean-and-variance normalization of speech recognition features," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, vol. 1, May 2006, pp. 773–776.

[6] 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.

[7] 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.

[8] B. Raj, V. N. Parikh, and R. M. Stern, "The effects of background music on speech recognition accuracy," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, vol. 2, Apr. 1997, pp. 851–854.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. Kim, "Auditory processing of speech signals for robust speech recognition in real-world noisy environments," *IEEE Trans. Speech and Audio Processing*, vol. 7, no. 1, pp. 55–69, Jan. 1999.

[14] H. Hermansky, "Perceptual linear prediction analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[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] P. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. H. Allerhand, "Complex sounds and auditory images," in *Auditory and Perception*. Oxford, UK: Y. Cazals, L. Demany, and K. Horner, (Eds), Pergamon Press, 1992, pp. 429–446.

[17] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[18] B. .C. J. Moore and B. R. Glasberg, "A revision of Zwicker's loudness model," *Acustica - Acta Acustica*, vol. 82.

[19] J. Volkmann, S. S. Stevens, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch (A)," *J. Acoust. Soc. Am.*, vol. 8, no. 3, pp. 208–208, Jan 1937.

[20] 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.

[21] D. Gelbart and N. Morgan, "Evaluating long-term spectral subtraction for reverberant ASR," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2001, pp. 103–106.