WIND NOISE REDUCTION USING NON-NEGATIVE SPARSE CODING

Mikkel N. Schmidt, Jan Larsen

Technical University of Denmark
Informatics and Mathematical Modelling
Richard Petersens Plads, Building 321
2800 Kgs. Lyngby

Fu-Tien Hsiao

IT University of Copenhagen
Multimedia Technology
Rued Langgaards Vej 7
2300 Copenhagen S.

ABSTRACT

We introduce a new speaker independent method for reducing wind noise in single-channel recordings of noisy speech. The method is based on non-negative sparse coding and relies on a wind noise dictionary which is estimated from an isolated noise recording. We estimate the parameters of the model and discuss their sensitivity. We then compare the algorithm with the classical spectral subtraction method and the Qualcomm-ICSI-OGI noise reduction method. We optimize the sound quality in terms of signal-to-noise ratio and provide results on a noisy speech recognition task.

1. INTRODUCTION

Wind noise can be a major problem in outdoor recording and processing of audio. A good solution can be to use a high quality microphone with a wind screen; this is not possible, however, in applications such as hearing aids and mobile telephones. Here, we typically have available only a single-channel recording made using an unscreened microphone. To overcome the wind noise problem in these situations, we can process the recorded signal to reduce the wind noise and enhance the signal of interest. In this paper, we deal with the problem of reducing wind noise in single-channel recordings of speech.

There exists a number of methods for noise reduction and source separation. When the signal of interest and the noise have different frequency characteristics, the Wiener filter is a good approach to noise reduction. The idea is to attenuate the frequency regions where the noise is dominant. In the case of speech and wind noise, however, this approach leads only to limited performance, since both speech and wind noise are non-stationary broad-band signals with most of the energy in the low frequency range as shown in Figure 1.

Another widely used approach is spectral subtraction [1]. Here, the idea is to subtract an estimate of the noise spectrum from the spectrum of the mixed signal. Spectral subtraction takes advantage of the non-stationarity of the speech signal by reestimating the noise spectrum when there is no speech activity. During speech activity, the noise is assumed stationary, and for this reason the method is best suited for situations where the noise varies slowly compared to the speech. This is not the case for wind noise. As illustrated in Figure 2, wind noise changes rapidly and wind gusts can have very high energy.

A number of methods for separating non-stationary broad-band signals based on source modeling have been proposed. The idea is to first model the sources independently and then model the mixture using the combined source models. Finally, the sources can be reconstructed individually for example by refiltering the mixed signal. Different models for the sources have been proposed, such as a hidden Markov model with a Gaussian mixture model [2], vector quantization [3, 4], and non-negative sparse coding [5]. A limitation of these approaches is that each source must be modeled prior to the separation. In the case of wind noise reduction, this means that we must model both the speech and the wind noise beforehand.

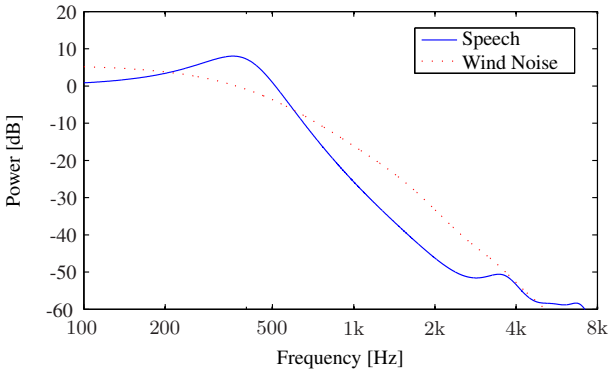

Fig. 1. Average spectrum of speech and wind noise. Both speech and wind noise are broad-band signals with most of the energy in the low frequency range. The spectra are computed using the Burg method based on a few seconds of recorded wind noise and a few seconds of speech from eight different speakers.

Binary spectral masking is a source separation method, where the main assumption is that the sources can be separated by multiplying the spectrogram by a binary mask. This is reasonable when each time-frequency bin is dominated by only one source. Thus, the problem of separating signals is reduced to that of estimating a binary time-frequency mask. One approach to estimating the mask is to use a suitable classification technique such as the relevance vector machine [6]. Similar to the source modeling approach, however, both the sources must be known in advance in order to estimate the parameters of the classifier.

A completely different approach to source separation is computational auditory scene analysis (CASA). Here, the idea is to simulate the scene analysis process performed by the human auditory system. We will not discuss this further in this paper.

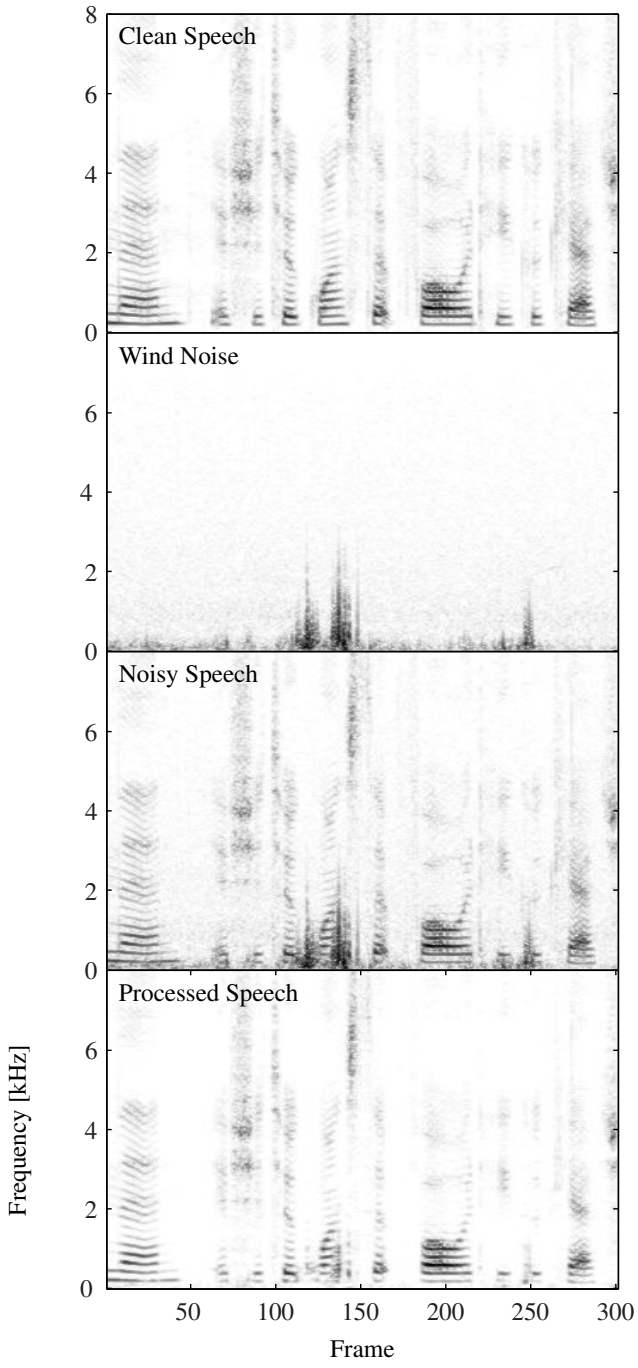

Fig. 2. Example spectrograms and the result of the algorithm. Spectrograms of clean speech and wind noise: Both speech and wind noise are non-stationary broad-band signals. Speech has both harmonic and noise-like segments and sometimes short pauses between words. Wind noise is characterized by a constant broad-band background noise and high energy broad-band wind gusts. There is a large overlap between the speech and noise in the noisy recording. In the processed signal, a large part of the noise is removed.

2. METHOD

In this work, we propose a new method for noise reduction, which is related to the source modeling approach using non-negative sparse coding. The key idea is to build a speaker independent system, by having a source model for the wind noise but not for the speech.

We assume that the speech signal and the wind noise are additive in the time domain, i.e., we assume that the noise is not so strong, that we have problems with saturation. Then, the noisy signal, $x(t)$, can be written as

$$x(t) = s(t) + n(t), \tag{1}$$

where $s(t)$ is the speech signal, and $n(t)$ is the wind noise. If we assume that the speech and wind noise are uncorrelated, this linearity applies in the power spectral domain as well.

In line with Berouti et al. [7], we represent the signal in the time-frequency domain as an element wise exponentiated short time Fourier transform

$$\boldsymbol{X} = |\mathrm{STFT}\{x(t)\}|^{\gamma}. \tag{2}$$

When the exponent, γ, is set to 2 the representation is the power spectrogram and the above mentioned linearity holds on average. Although using $\gamma \neq 2$ violates the linearity property, it often leads to better performance; in the sequel, we estimate a suitable value for this parameter.

2.1. Non-negative sparse coding

The idea in non-negative sparse coding (NNSC) is to factorize the signal matrix as

$$\boldsymbol{X} \approx \boldsymbol{D}\boldsymbol{H}, \tag{3}$$

where $\boldsymbol{D}$ and $\boldsymbol{H}$ are non-negative matrices which we refer to as the dictionary and the code. The columns of the dictionary matrix constitute a source specific basis and the sparse code matrix contains weights that determine by which amplitude each element of the dictionary is used in each time frame. It has been shown that imposing non-negativity constraints leads to a parts-based representation, because only additive and not subtractive combinations are allowed [8]. Enforcing sparsity of the code leads to solutions where only a few dictionary elements are active simultaneously. This can lead to better solutions, because it forces the dictionary elements to be more source specific.

There exists different algorithms for computing this factorization [9, 10, 11, 12]. In the following we use the method proposed by Eggert and Körner [10], which is perhaps not the most efficient method, but it has a very simple formulation and allows easy implementation. The NNSC algorithm starts with randomly initialized matrices, $\boldsymbol{D}$ and $\boldsymbol{H}$, and alternates the following updates until convergence

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \bullet \frac{\bar{\boldsymbol{D}}^{\top}\boldsymbol{X}}{\bar{\boldsymbol{D}}^{\top}\bar{\boldsymbol{D}}\boldsymbol{H}+\lambda}, \tag{4}$$

$$\boldsymbol{D} \leftarrow \bar{\boldsymbol{D}} \bullet \frac{\boldsymbol{X}\boldsymbol{H}^{\top}+\bar{\boldsymbol{D}}\bullet\left(\boldsymbol{1}\left(\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}^{\top}\bullet\bar{\boldsymbol{D}}\right)\right)}{\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}^{\top}+\bar{\boldsymbol{D}}\bullet\left(\boldsymbol{1}\left(\boldsymbol{X}\boldsymbol{H}^{\top}\bullet\bar{\boldsymbol{D}}\right)\right)}. \tag{5}$$

Here, $\bar{\boldsymbol{D}}$ is the columnwise normalized dictionary matrix, $\boldsymbol{1}$ is a square matrix of suitable size with all elements equal to 1, and the bold operators indicate pointwise multiplication and division. The parameter λ determines the degree of sparsity in the code matrix.

2.2. Non-negative sparse coding of a noisy signal

When the sparse coding framework is applied to a noisy signal and we assume that the sources are additive, we have

$$\boldsymbol{X} = \boldsymbol{X}_s + \boldsymbol{X}_n \approx [\boldsymbol{D}_s\ \boldsymbol{D}_n] \begin{bmatrix} \boldsymbol{H}_s \\ \boldsymbol{H}_n \end{bmatrix} = \boldsymbol{D}\boldsymbol{H}, \qquad (6)$$

where the subscripts, s and n, indicate speech and noise. Inherent in the sparse coding approach, however, is a permutation ambiguity; the order of the columns of $\boldsymbol{D}$ can be changed as long as the rows of $\boldsymbol{H}$ are changed correspondingly. Consequently, we need a mechanism to fix or determine which components pertain to which source. One method is to precompute the source dictionaries using isolated recordings of the sources [5]. Another idea is to devise an automatic grouping rule as argued by Wang and Plumbley [14]. We suggest to precompute the source dictionary for only one of the sources, the wind noise, and to learn the dictionary of the speech directly from the noisy data. This results in a method which is independent of the speaker.

We modify the NNSC algorithm so that only $\boldsymbol{D}_s$, $\boldsymbol{H}_s$, and $\boldsymbol{H}_n$ are updated. This gives us the following update equations

$$\boldsymbol{H}_s \leftarrow \boldsymbol{H}_s \bullet \frac{\bar{\boldsymbol{D}}_s^\top \boldsymbol{X}}{\bar{\boldsymbol{D}}_s^\top \bar{\boldsymbol{D}}\boldsymbol{H} + \ell_s}, \qquad \boldsymbol{H}_n \leftarrow \boldsymbol{H}_n \bullet \frac{\bar{\boldsymbol{D}}_n^\top \boldsymbol{X}}{\bar{\boldsymbol{D}}_n^\top \bar{\boldsymbol{D}}\boldsymbol{H} + \ell_n}, \quad (7)$$

$$\boldsymbol{D}_s \leftarrow \bar{\boldsymbol{D}}_s \bullet \frac{\boldsymbol{X}\boldsymbol{H}_s^\top + \bar{\boldsymbol{D}}_s \bullet \left(\boldsymbol{1}\left(\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}_s^\top \bullet \bar{\boldsymbol{D}}_s\right)\right)}{\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}_s^\top + \bar{\boldsymbol{D}}_s \bullet \left(\boldsymbol{1}\left(\boldsymbol{X}\boldsymbol{H}_s^\top \bullet \bar{\boldsymbol{D}}_s\right)\right)}. \qquad (8)$$

We have introduced different sparsity parameters for the speech and noise because we hypothesize that having different sparsity for the speech and noise can improve the performance of the algorithm.

To reduce the wind noise in a recording we first compute the NNSC decomposition of an isolated recording of the wind noise using Equation (4–5). We discard the code matrix and use the noise dictionary matrix to compute the NNSC decomposition of the noisy signal using Equation (7–8). Finally we estimate the clean speech as

$$\hat{\boldsymbol{X}}_s = \bar{\boldsymbol{D}}_s \boldsymbol{H}_s. \qquad (9)$$

To compute the waveform of the processed signal, we invert the STFT using the phase of the noisy signal.

3. EXPERIMENTAL RESULTS

To evaluate the algorithm we first used a test set consisting of eight phonetically diverse sentences from the Timit database. The sentences were spoken by different speakers, half of each gender. The speech signals were normalized to unit variance. We recorded wind noise outdoors using a setup emulating the microphone and amplifier in a hearing aid. We used half a minute of wind noise for estimating the noise dictionary. The signals were sampled at 16 kHz and the STFT were computed with a 32 ms Hanning window and 75% overlap. We mixed speech and wind noise at signal-to-noise ratios (SNR) of 0, 3, and 6 dB. In all our experiments the stopping criterion for the algorithm was when the relative change in the squared error was less than 10^{-4} or at a maximum of 500 iterations. As for most non-negative matrix factorization methods, the NNSC algorithm is prone to finding local minima and thus a suitable multi-start or multi-layer approach could be used [13]. In practice, however, we obtained good solutions using only a single run of the NNSC algorithm.

3.1. Initial setting of parameters

To find good initial values for the parameters of the algorithm, we evaluated the results on an empirically chosen range of values for each of the parameters shown below.

- $\gamma \in \{.5, \underline{.6}, .7, .8\}$ The exponent of the short time Fourier tranform.
- $\lambda_n \in \{\underline{.2}, .5\}$ The sparsity parameter used for learning the wind noise dictionary.
- $N_s \in \{32, \underline{64}, 128\}$ The number of components in the speech dictionary.
- $N_n \in \{4, 16, \underline{64}, 128\}$ The number of components in the wind noise dictionary.
- $\ell_s \in \{\underline{.05}, .1, .2\}$ The sparsity parameter used for the speech code during separation.
- $\ell_n \in \{\underline{0}, .1\}$ The sparsity parameter used for the noise code during separation.

For each of the 576 combinations of parameter settings, we computed the average increase in SNR. In total, more than six hours of audio was processed. The underlined parameter settings gave the highest increase in SNR. We used these parameter settings as a starting point for our furhter experiments. An example of the result of the algorithm is illustrated in Figure 2.

3.2. Importance and sensitivity of parameters

Next, we varied the parameters one by one while keeping the others fixed to the value chosen above. In these experiments, the input SNR was fixed at 3 dB. Figure 3–8 show the results; the box plots shows the median, upper and lower quartiles, and the range of the data. In the following we comment on each parameter in detail.

- γ (See Figure 3) The exponent of the STFT appears to be quite important . The best results in terms of SNR is achieved around $\gamma = 0.7$, although the algorithm is not particularly sensitive as long as γ is chosen around 0.5–1. Noticably, results are significantly worse when using the power spectrogram representation, $\gamma = 2$. The estimated value of the exponent corresponds to a cube root compression of the power spectrogram which curiously is an often used approximation to account for the nonlinear human perception of intensity.
- λ_n (See Figure 4) The sparsity parameter used in estimating the wind noise dictionary does not significantly influence the SNR . Qualitatively, however, there is a difference between low and high sparsity. Listening to the processed signals we found that with a less sparsified noise dictionary, the noise was well removed, but the speech was slightly distorted. With a more sparsified dictionary, there was more residual noise. Thus, this parameter can be used to make a tradeoff between residual noise and distortion.
- N_s (See Figure 5) The number of components in the speech dictionary is a very important parameter. Naturally, a reasonable number of components is needed in order to be able to model the speech adequately. Qualitatively, when using too few components, the result is a very clean signal consisting only of the most dominant speech sounds, most often the vowels. Interestingly though, having too many components also reduces the performance, since excess components can be used to model the noise. In this study we found

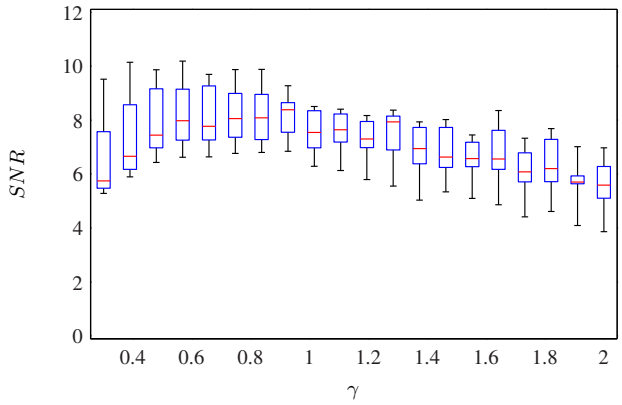

Fig. 3. Exponent of the short time Fourier transform versus SNR. The best performance is achieved around $\gamma = 0.7$. The algorithm is not very sensitive to γ as long as it is chosen around 0.5–1.

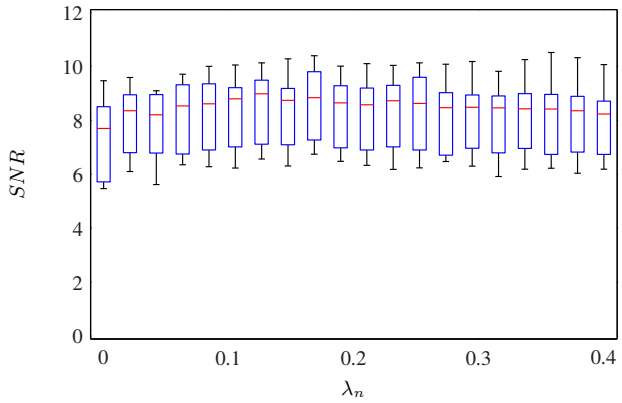

Fig. 4. Sparsity parameter for the precomputation of the wind noise dictionary versus SNR. The method is not particularly sensitive to the selection of this parameter.

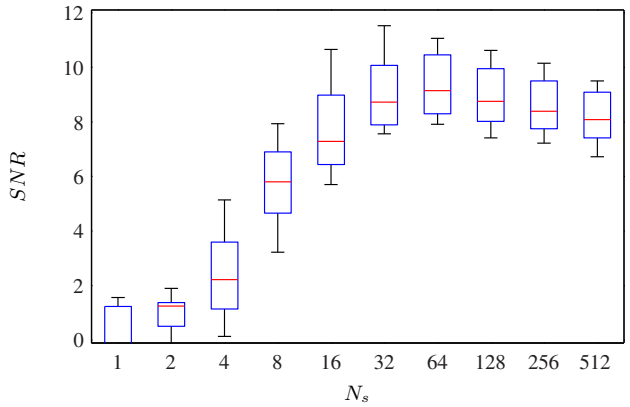

Fig. 5. Number of components in the speech dictionary versus SNR. The best performance on the test set is achieved at $N_s = 64$. Using too few or too many components reduces the performance.

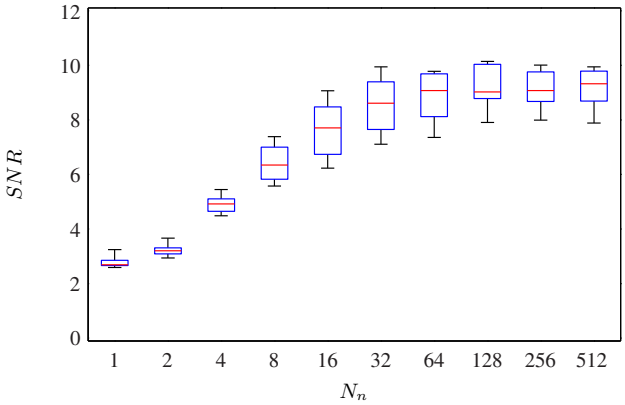

Fig. 6. Number of components in the wind noise dictionary versus SNR. The results indicate that there should be at least $N_n = 32$ noise components.

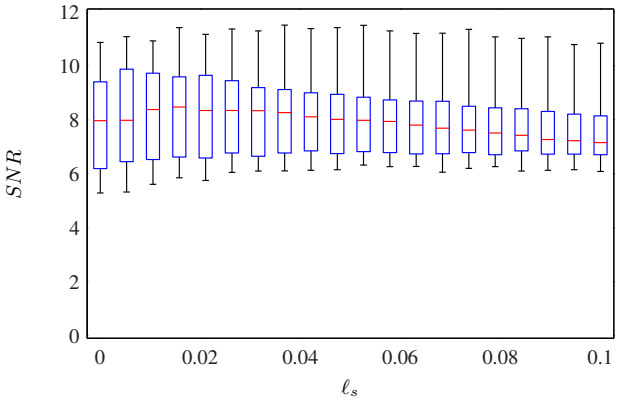

Fig. 7. Sparsity parameter for the speech versus SNR. The method is not particularly sensitive to the selection of this parameter.

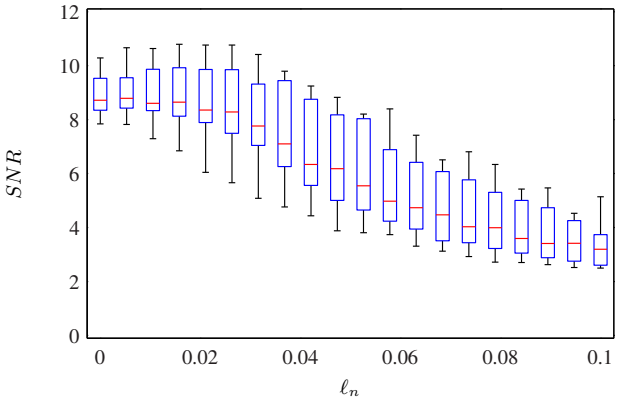

Fig. 8. Sparsity parameter for the noise versus SNR. The method is very sensitive to the selection of this parameter, and it appears that no sparsity, $\ell_n = 0$, leads to the best performance.

that $N_s = 64$ components gave the best results, but we expect that it is dependent on the length of the recordings and the setting of the sparsity parameters etc.

N_n (See Figure 6) The number of components in the wind noise dictionary is also important. Our results indicate that at least $N_n = 32$ components must be used and that the performance does not decrease when more components are used. Since the noise dictionary is estimated on an isolated recording of wind noise, all the elements in the dictionary will be tailored to fit the noise.

ℓ_s (See Figure 7) The sparsity parameter used for the speech code does not appear very important when we look at the SNR, although slightly better results are obtained around $\ell_s = 0.02$. When we listen to the signals, however, there is a huge difference. When the parameter is close to zero, the noise in the processed signal is mainly residual wind noise. When the parameter is chosen in the high end of the range, there is not much wind noise left, but the speech is distorted. Thus, although not reflected in the SNR, this parameter balances residual noise and distortion similar to the sparsity parameter used for estimating the wind dictionary.

ℓ_n (See Figure 8) The sparsity parameter used for the wind noise during separation should basically be set to zero. Both qualitatively and in terms of SNR, imposing sparsity on the noise code only worsens performance. This makes sense, since the sparsity constrains the modeling ability of the noise dictionary, and consequently some of the noise is modeled by the speech dictionary.

3.3. Comparison with other methods

We compared our proposed metod for wind noise reduction to two other noise reduction methods. We used a test set consisting of 100 sentences from the GRID corpus. The sentences were spoken by a single female speaker. We mixed the speech with wind noise at different signal-to-noise ratios in the range 0–6 dB to see how the algorithm works under different noise conditions. All parameter settings were chosen as in the previous experiments.

We compared the results with the noise reduction in the Qualcomm-ICSI-OGI frontend for automatic speech recognition [15], which is based on adaptive Wiener filtering. We also compared to a simple spectral subtraction algorithm, implemented with an "oracle" voice activity detector. During non-speech activity we set the signal to zero and when speech was present we subtracted the spectrum of the noise taken from the last non-speech frame.

We computed two quality measures: i) the signal to noise ratio averaged over the 100 sentences and ii) the word recognition rate using an automatic speech recognition (ASR) system. The features used in the ASR were 13 Mel frequency cepstral coefficients plus Δ and $\Delta\Delta$ coefficients, and the system was based on a hidden Markov model with a 16 component Gaussian mixture model for each phoneme. The results are given in Figure 9– 10.

In terms of SNR, our proposed algorithm performs well (see Figure 9). The spectral subtraction algorithm also increases the SNR in all conditions, whereas the Qualcomm-ICSI-OGI algorithm actually decreases the SNR. In terms of word recognition rate the Qualcomm-ICSI-OGI algorithm gives the largest quality improvement (see Figure 10). This might not come as a surprise, since the algorithm is specifically designed for preprocessing in an ASR system. At low SNR, our proposed algorithm does increase the word recognition rate, but at high SNR, it is better not to use any noise reduction at all. The spectral subtraction algorithm performs much worse than using the original noisy speech in all conditions.

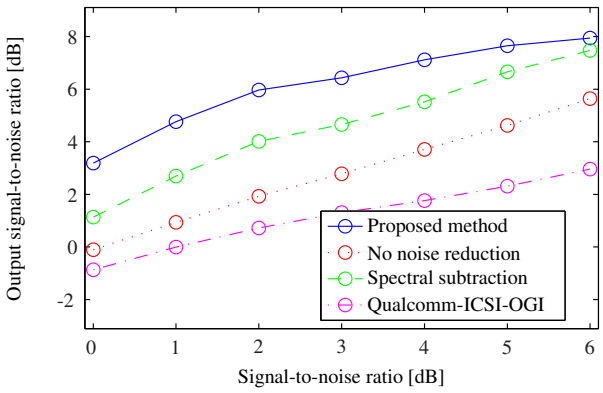

Fig. 9. Output SNR versus input SNR. In terms of SNR, the proposed algorithm performs well.

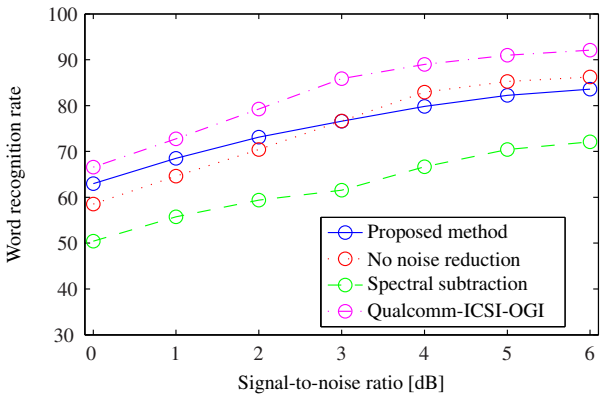

Fig. 10. Word recognition rate on a speech recognition task versus input SNR. The Qualcomm-ICSI-OGI algorith which is designed for this purpose performs best. At low SNR our proposed algorithm gives better results than using the noisy speech directly.

4. DISCUSSION

We have presented an algorithm for reducing wind noise in recordings of speech based on estimating a source dictionary for the noise. The main idea was to make a system based on non-negative sparse coding, using a pre-estimated source model only for the noise. Our results show that the method is quite effective, and informal listening test indicate that often the algorithm is able to reduce sudden gusts of wind where other methods fail. In this work, we studied and optimized the performance in terms of signal-to-noise ratio, which is a simple but limited quality measure. Possibly, the algorithm will perform better in listening test and in speech recognition tasks, if the parameters are carefully tuned for these purposes, e.g., by optimizing a perceptual speech quality measure or word recognition rate.

5. REFERENCES

[1] Steven F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113–120, 1979.

[2] Sam T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems*, 2000, pp. 793–799.

[3] Sam T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Eurospeech*, 2003, pp. 1009–12.

[4] Daniel P. W. Ellis and Ron J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *International Conference on Acoustics, Speech and Signal Processing*, may 2006, pp. 957–960.

[5] Mikkel N. Schmidt and Rasmus K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006.

[6] Ron J. Weiss and Daniel P. W. Ellis, "Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking," in *Statistical and Perceptual Audio Processing, Workshop on*, 2006.

[7] M Berouti, R Schwartz, and J Makhoul, "Enhancement of speech corrupted by acoustic noise," in *International Conference on Acoustics, Speech and Signal Processing*, 1979, vol. 4, pp. 208–211.

[8] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[9] P.O. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing, IEEE Workshop on*, 2002, pp. 557–565.

[10] Julian Eggert and Edgar Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, 2004, vol. 4, pp. 2529–2533.

[11] Chih-Jen Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation (to appear)*, 2007.

[12] Dongmin Kim, Suvrit Sra, and Inderjit S. Dhillon, "Fast newton-type methods for the least squares nonnegative matrix approximation problem," in *Data Mining, Proceedings of SIAM Conference on*, 2007.

[13] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization," *Electronic Letters*, vol. 42, no. 16, pp. 947–958, 2006.

[14] B. Wang and M. D. Plumbley, "Musical audio stream separation by non-negative matrix factorization," in *DMRN Summer Conference, Glasgow, Proceedings of the*, july 2005.

[15] Andre Adami, Lukás Burget, Stephane Dupont, Hari Garudadri, Frantisek Grezl, Hynek Hermansky, Pratibha Jain, Sachin Kajarekar, Nelson Morgan, and Sunil Sivadas, "Qualcomm-icsi-ogi features for asr," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2002, pp. 21–24.