Genova, September 24-26

Paper presented at the 2nd European Confrence on Speech Communication and Techonology - Eurospeech 91 Proceedings, vol.2 pp 499-502

SPECTRAL SUBTRACTION FOR FRONT-END NOISE REDUCTION IN A SPEECH RECOGNIZER

Carlos J. Teixeira and Isabel M. Trancoso

INESC / IST
INESC, R. Alves Redol, 9, 1000 LISBON, Portugal
representing the SUNSTAR consortium[1]

ABSTRACT

The goal of this work is the design of a front-end processor to be used with state-of-the-art word recognizers, capable of reducing stationary noise from a speech signal transmitted over the public telephone network, to be implemented on a single DSP32C board, integrating an echo canceller module as well. The environment in which the speech signal is produced can vary from the typical office environment (with noise emerging from keyboards, air conditioning, etc.) to public buildings, public phone booths and even homes. Besides this environmental noise, the speech signal is also corrupted by telephone line noise and its original amplitude and phase characteristics are also subject to channel distortion. The selected noise reduction method is the classic spectral subtraction algorithm. This selection was primarily motivated by its relative low complexity, in view of the real-time operation requirements.The tests performed so far with real environmental and telephone line noise signals confirm the effectiveness of integrating this module as a preprocessor for speech recognition in a public telephone network environment.

Keywords: noise reduction, spectral subtraction, speech recognition.

1. INTRODUCTION

The potentially vast market for applications of speech recognition poses increasing demands on the performance of these systems. The problem addressed in this paper concerns the services offered through the public telephone network. In this context, the recognition task is hindered by the presence of telephone line noise, as well as by noise resulting from non-ideal environmental conditions: the user may be seated at his office, with typical background noise such as keyboards, air conditioning, ringing telephones, printers, etc.; or he may be using a phone box in a street, inside a public building such as railway stations or shopping centres, or even at home. Additional amplitude and phase distortion are introduced by the telephone channel. This type of distortion, however, will not be taken into account in this paper.

There are several ways of dealing with the presence of noise in recognition systems. The adoption of large noise-corrupted training sets is a relatively common approach. Other more sophisticated methods use integrated modeling of speech and noise (Varga, 1990). Here, a pre-processing approach was selected, for the sake of enhancing the modularity of the application. Noise reduction (NR) is performed on the speech signal, and the output of the module is available in both analog and digital representations, thus being combinable with any already developed speech recognizer. Some of the most common pre-processing techniques use more than one microphone for adaptive cancelling. In the context of the public telephone network, however, this type of approaches is not feasible.

The selected noise reduction method is the spectral subtraction algorithm proposed in (Boll, 1979). One of the most relevant factors in this selection was the relative low complexity of the algorithm, which makes its real-time implementation feasible on a DSP32C board incorporating an echo canceller module as well. Another major motivation was the fact that the method does not involve previous training, and is known to produce good results with quasi-stationary noise (Van Compernolle, 1989).The algorithm was adapted for real-time implementation, namely in what concerns the speech/non-speech detection and the updating of the spectral amplitude estimate of noise.

This paper is structured into three major sections: the necessarily brief overview of the basic algorithm; the description of implementation details and the module assessment. The final section summarizes this work, pointing to future developments.

2. ALGORITHM DESCRIPTION

The adopted method suppresses stationary noise from speech by subtracting the spectral noise bias calculated during non-speech activity. The algorithm assumes that the input speech signal x_i has an additive noise component n_i (i.e., $x_i = s_i + n_i$), which can be regarded as stationary on a short-term basis. Hence, the signal amplitude which is estimated during a noisy segment may be considered equal to the amplitude estimated during the immediately following corrupted speech segments. This type of noise reduction algorithm only removes the effect of noise from the spectral amplitude, disregarding its effect on phase.

The block diagram of the algorithm is depicted in Fig.1. For each input signal frame, its DFT $\{X_k\}$ is computed and, based on the energy of the frame, a speech/non-speech decision is

[1]The work in the SUNSTAR project is done within the framework of the ESPRIT programme and partly funded by the Commission of the European Communities. The following companies form the SUNSTAR consortium: Jydsk Telefon (DK), to which INESC is associated, Alcatel FACE Standard (I), Fraunhofer Gesellschaft (D), and Telefónica I+D (E).

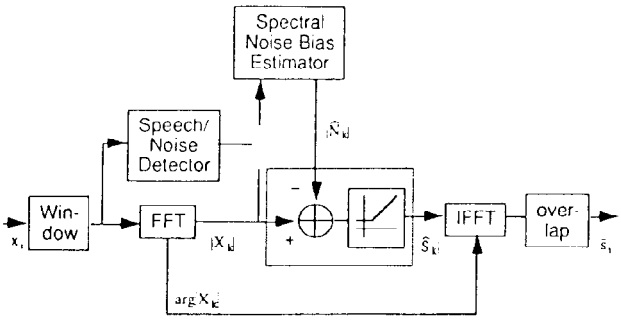

Fig. 1 Block diagram of spectral subtraction algorithm.

taken. For non-speech frames, the DFT signal is used to update an estimate of the spectral noise bias $|\hat{N}_k|$. This noise bias is then subtracted from the magnitude of the DFT signal, and the resulting signal is rectified if negative, in order to obtain positive spectral amplitudes ($|\hat{S}_k|$). Finally, an inverse DFT is performed, in order to estimate $\hat{s}_i$.

Five main blocks can be identified in the diagram: the direct and inverse DFTs, the speech/noise detection, the spectral noise bias estimation and the spectral magnitude subtraction. In addition to these, two other blocks are used for input-output buffering and windowing. Typically, a 50% overlap between analysis windows is adopted. The window length is chosen to be approximately twice as large as the maximum expected pitch period. For a sampling rate of 8 kHz, Hamming windows of 256 samples (16 ms), shifted by 128 samples are adopted. This implies a fixed processing delay of 48 ms between input and output samples.

The direct and inverse DFT signals are computed using the Fast Fourier Transform algorithm. The same routine is used for the direct and the inverse transformation.

The purpose of the speech/noise detector block is to decide whether a particular segment holds noise-corrupted speech or just noise. As there are frontier segments where the speech activity begins or ends, a binary decision cannot be rigourous. However, by adopting a sufficiently short segment size, the detection errors do not seriously compromise the overall performance.

The noise reduction module uses an energy-based speech/non-speech detector. The first subblock in this detector computes an estimate of the energy in each input frame. If this estimate is lower than a given threshold, then that frame is classified as noise, otherwise it will be labeled as corrupted speech. Based on this decision, one of two related energy estimates (speech or noise) is recursively updated using an exponential forgetting factor. In the last subblock, a new threshold is set up above the noise energy estimate, by an amount which is proportional to the difference between this estimate and the corrupted speech energy estimate. For non-speech frames, the magnitude of the DFT signal is used to update the spectral magnitude noise bias. This is also done recursively, using an exponential forgetting factor, in the spectral noise bias estimator block. This noise bias is then subtracted from $|X_k|$ for both speech and non-speech frames. After subtraction, the resulting values having negative magnitudes are set to zero.

A non-detailed fluxogram of the spectral subtraction algorithm is shown in Fig.2. Notice that all the processing that takes place between the two FFT blocks is repeated for k=1,...,256.

The algorithm has two major limitations: the noise must be locally quasi-stationary, and the average energy of the noise signal must be lower than the one of the speech signal. Whilst the first limitation is intrinsic to the spectral subtraction method, the second one derives from the particular implementation of the speech/noise detector. However, this implementation is quite fast and the detector is not very sensitive to variations in signal-to-noise ratio, for values above 5 dB.

3. REAL-TIME IMPLEMENTATION

The NR module has been implemented on a single DSP32C board from Loughborough Sound Images Limited. The code can be downloaded to the normal I/O bus of a PC/AT and has been originally written in assembly, structured into three main modules: the one which applies a Hamming window to the input signal and adds half-overlapped buffers in order to reconstruct the output signal, the one which implements the spectral subtraction using several library routines, and the interface module. Excluding this module, the software takes about 4 Kbytes for variables and 3 Kbytes for code and constants, and the execution time is less than 6 ms, for each speech segment of 16 ms, i.e., close to 35 % real time. More than half of this time is spent on the direct and inverse FFT routine.

The interface module has been developed by other partners (Lindberg, 1990), and includes different downloading PC programs. One of these programs allows real-time access to the the A/D and D/A converters of the DSP32C board. This is particularly useful for real-time demonstration purposes. The total execution time takes about 40 % of the real time.

Another program performs batch processing of signal files at about the triple of the real time (with a 28.8 ms hard disk average seek time). This is essential for testing, so that other software tools may have access to the signal files. There is also the possibility of having real-time digital signal connection to other DSP boards via ST-BUS. This will be the final path to the recognizer but was not tested so far.

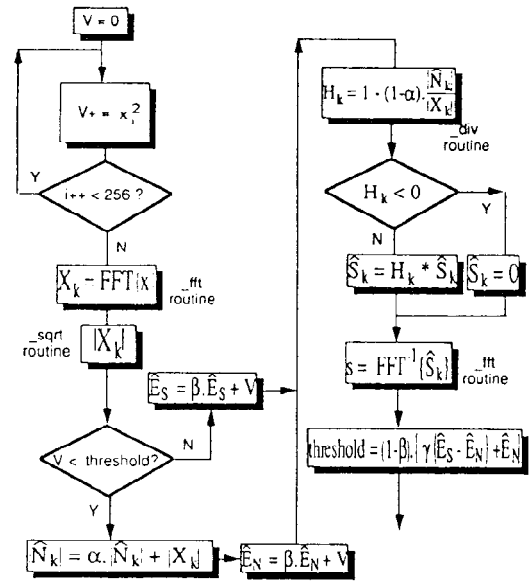

Fig. 2 Fluxogram of spectral subtraction algorithm.

In the present software version, the user is not allowed to change the empirically tuned constants of the algorithm. To change this limitation, a new assembly code must be produced. The commands for the NR module are therefore very simple, being directly issued from the DOS environment of the PC.

The above mentioned figure of 40% execution time means that a single DSP32C is fast enough to support further signal processing. An echo canceller, developed within the consortium, could therefore be included in the same board.

4. MODULE ASSESSMENT

The algorithm assessment has been done in two steps: the first phase included only informal listening tests and measurements of the signal-to-noise ratio (SNR) improvement; in the second phase, the noise reduction module has been tested with a connected-word speech recognizer based on Hidden Markov modeling, and the error rate has been compared with the results obtained without front-end processing.

Although further testing with larger databases is envisaged during the coming months, the preliminary tests reported in this section involved a very restricted database of 4 speakers (2 male, 2 female), with a vocabulary of 20 words (the numbers 0 to 10, "yes", "no", and a few one and two-word utterances such as "alarm-call", for instance).

The noise material has been collected by the SUNSTAR partners and includes both environmental and telephone line noise examples. Among the almost one hundred noise samples available, 4 were selected as representing the most typical noise categories encountered in the project applications: an internal telephone line noise, an external telephone line noise, an environmental office-type noise (printer) and an environmental non-office noise (railway station).

The pre-recorded noise and clean speech files were 1 minute long, with a sampling frequency of 8 kHz. In order to simulate different input signal-to-noise ratios, the noise files were artificially scaled before being added to the clean speech files. A specific program was developed in C, for scaling purposes, in which the user enters the names of the input speech and noise files and the desired SNR. The program computes the scaling factor, without taking into account the non-speech regions of the clean speech file. The speech/non-speech detector referred above was also used for this purpose. All tests have been performed for 2 target SNRs: 5 and 15 dB.

The first type of assessment consisted of the subjective comparison of the perceptual quality of the input and output signals for the NR module. These tests involved only two trained listeners. The quality of the processed signal was judged more clear than the one of the noise-corrupted signal. This improvement, however, was achieved at the cost of an unpleasant tonal distortion which increased with the input noise intensity.

The comparison of global and segmental signal-to-noise ratios provided a first objective measure of the performance of the module. Both signal-to-noise ratios were measured only during speech segments, relative to the clean speech signal. For the higher input SNR, the global improvement was not significant ($\approx$ 1 dB). For the lower, however, an average 3 dB improvement was achieved. This performance was not homogeneous at all: as expected, the improvement was much more pronounced for the most stationary-like types of noise, such as internal and external telephone line noise (average 6 and 4 dB, respectively), than for the highly non-stationary printer noise (1 dB). With the public building noise, an intermediate improvement was achieved (2 dB). Although no definite conclusions can be drawn for such a limited number of speakers, it is nevertheless worth mentioning the better objective performance of the module with the female voices (average 4 dB improvement for the lower input SNR, as compared with an average 2.7 dB, for male voices). Table 1 summarizes the SNR results. The indicated values are the differences between the obtained global SNR and the imposed input SNR.

As the client for the NR module is a speech recognizer, more significant objective measurements could be obtained by measuring success/failure recognition rates. The recognition system was trained with non-native english speakers, and tested with other non-native speakers of very different accents, and using different microphone sets. This was the reason for not obtaining very good recognition rates, even with clean speech.

The tests with internal and external telephone line noise showed no improvements for input SNR=15 dB, compared with the results obtained without noise reduction and, in some cases, a slight degradation was observed; for input SNR=5 dB, however, the number of correctly recognized words increased on average by 5, out of the 20 words of the application vocabulary. For the non-office noise, the improvement was equally relevant. The performance with the printer noise, however, clearly demonstrated the non adequacy of the method for non-stationary noise. A slight degradation is even observed in some cases, which can be attributed to the fact that the speech/non-speech detector is often activated at the wrong times. Table 2 summarizes the improvements on recognition scores. The indicated values are the differences between the number of correctly recognized words with and without noise reduction.

Table 1

Differences between output and input SNR.

TYPE OF NOISE / INPUT SNR	5 [dB]	15 [dB]
Internal telephone line noise	5.6	1.7
External telephone line noise	3.5	1.1
Environmental office noise	1.2	1.1
Environmental non-office noise	2.3	0.6
Average	3.2	1.1

Table 2

Differences between the number of correctly recognized words with and without noise reduction.

TYPE OF NOISE / INPUT SNR	5 dB	15 dB
Internal telephone line noise	5.3	1.0
External telephone line noise	4.0	-2.7
Environmental office noise	-0.6	-1.3
Environmental non-office noise	5.3	0.3

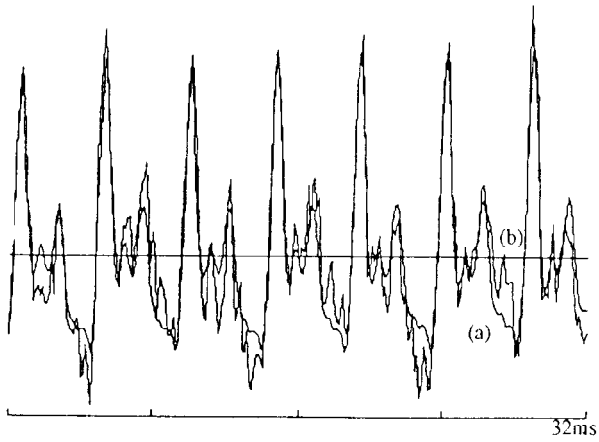

Fig.3 Waveforms of clean (a) and noisy (b) speech signals.

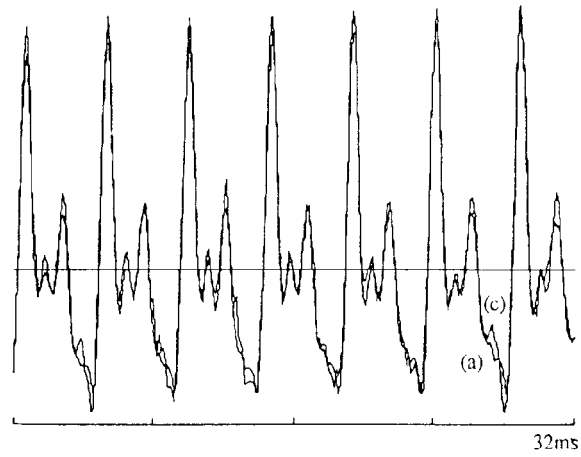

Fig.4 Waveforms of clean (a) and processed (c) speech signals.

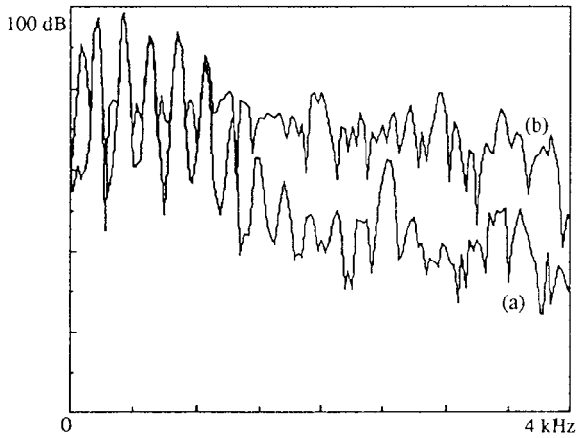

Fig.5 Short-time spectral amplitude plots of clean (a) and noisy (b) speech signals.

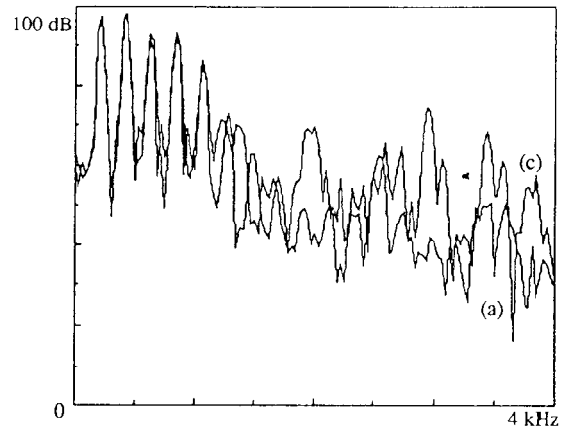

Fig.6 Short-time spectral amplitude plots of clean (a) and processed (c) speech signals.

Figures 3 through 6 illustrate the performance of the noise reduction algorithm with internal telephone line noise, and an input SNR of 5 dB. Three superimposed signals are are shown: (a) indicates the clean speech signal, (b) the noise-corrupted one and (c) the processed signal. The first two figures represent time-domain waveforms, whereas the last ones represent short-time spectral amplitude plots. The lower frequency regions of the speech spectra are particularly well recovered by the noise reduction module. Some noise bands are even attenuated by more than 20 dB.

CONCLUSIONS

A front-end processing algorithm for speech recognition was implemented in real time, capable of reducing stationary and quasi-stationary noise, in the context of the public telephone network. The tests performed so far with real environmental and telephone line noise signals confirm the effectiveness of the module in these conditions. For other types of noise, however, more sophisticated approaches must be followed which explore speech specific features, as opposed to using only energy measurements. In this context, noise reduction methods integrated with the recognition system seem better suited to modeling speech and noise than front-end processing techniques.

ACKNOWLEDGEMENTS

The authors would like to thank Profs. Paul Dalsgaard and Børge Lindberg from the Speech Technology Centre of the University of Aalborg, for their help and advice throughout this work, and also their colleagues Carlos Ribeiro, Luís Oliveira and Prof. António Serralheiro from INESC, for their cooperation in the implementation and testing of the module.

REFERENCES

S. Boll (1979), "Suppression of Acoustic Noise in Speech Using Spectral Subtraction", *IEEE Trans. Acoustic, Speech, and Signal Proc.*, Vol. ASSP-27, pp.113-120.

B. Lindberg (1990), "3R Software Documentation, Part 1, Functional Description", Speech Technology Centre, Institute of Electronic Systems, University of Aalborg, Denmark.

D. Van Compernolle (1989), "Noise Adaptation in a Hidden Markov Model Speech Recognition System" *Computer Speech and Language* , no. 3, pp. 151-167.

A. Varga and R. Moore (1990), "Hidden Markov Model Decomposition of Speech and Noise", *IEEE Proc. ICASSP 90*, pp. 845-848.