Data-driven Speech Denoising Using Noise Profiles

Kamil Ekštein and Pavel Král
Dept. of Computer Science and Engineering, Faculty of Applied Sciences,
University of West Bohemia, Plzeň, Czech Republic
{kekstein, pkral}@kiv.zcu.cz

Abstract— **This paper describes a targeted, undemanding data-driven signal processing approach to identify, control, and suppress a specific background noise which is present in a recording together with a spoken utterance.**

A background noise (like e.g. the sound of an engine onboard a bus) negatively influences the ASR[1] system performance by distorting the speech signal spectrum. Thus it is necessary to preprocess the signal in order to suppress the noise (or preferably to dispose of it). A reliable outcome has been currently achieved using source separation techniques like e.g. ICA[2] or PCA[3] [3]. Unfortunately these techniques are both computationally demanding and excessively aggressive to the speech.

The hereinafter depicted denoising process at first uses a newly proposed metric based upon a frequency-energy combination matrix to identify the nature of the background noise in the signal. Then a noise profile is selected from a library and applied onto the signal by means of spectral shaping (specifically e.g. smoothed spectral subtraction).

The method is asymmetrical: The noise profiles are computed from recordings of specific background noises taken in railway station halls, buses, cars, etc. The profile processing is performed offline. The online signal processing is efficient in terms of computational costs—it uses the precomputed profiles and works in real time.

The method is particularly suitable for utterances where the background noise level reaches the level of the speech (i.e. when the noise is at about 0 to -6 dB).

I. INTRODUCTION

The performance of contemporary ASR systems unwinds above all from the quality of the training phase and the quality and quantity of the used training material. The robustness of the systems against a background noise depends on whether a background noise was present in the training material and thus the system is trained to a noised speech.

The ASR systems that were trained with enough material gathered in a real environment in which the system is expected to operate reach impressive results of more than 97% accuracy. However, if such a system is operated in an acoustically different environment with different kinds of background noise the performance drops significantly—in many cases below an acceptable level for a real-life operation (see table IV below).

Nowadays most of the state-of-the-art ASR systems are trained using a large set of recordings from a wide variety of acoustic environments with diverse noises. Nonetheless there are still systems and applications where the training material does not (and sometimes generally can not) cover all possible acoustic environments and noises influencing the recordings to recognize. In this case there must be a **denoising** (or noise-conditioning) **algorithm** present to reduce the negative impact of an unknown acoustic environment onto the performance of the system.

II. DENOISING GENERALLY

There are generally two large groups of denoising techniques: (i) simple **filter-based** techniques and more advanced (ii) **source separation** techniques.

Simple denoising techniques are usually based on apriori considerations about the character of the noise or more generally of the whole acoustic environment. These techniques have low computational demands and load and mostly comprise of a FIR[4] or IIR[5] filter acting as a *gate* (or the so-called *pass filter*).

At the beginning a representative amount of the recorded material is observed and analysed using spectral analysis techniques. These techniques reveal the presence of a spectrally localised noise and a pass filter can be designed to remove it (or at least reduce it).

However, such a denoising strategy can be used only in the cases when the noise is stable, i.e. it occurs in a certain fixed region of the frequency spectrum. Such a situation happens for example when the recording gear is of poor quality and the AC[6] network frequency 50 Hz leaks into the signal.

Unfortunately under real conditions the noise is variable and only few ASR systems are deployed in a situation where the background noise is at least quasi-stable.

The more advanced—both from computational and implementational point of view—denoising techniques are based upon the so-called *source separation* (see [3]). These can be regarded as "demixing" the recorded signal into a clean speech and a background noise. If the sole noise signal was known it could be easy to subtract it from the mixture of the speech and the noise. Of course such an approach is highly theoretical and under real conditions it is impossible to isolate the noise from the mixture mainly for the fact that the character (or features) of the noise is apriory unknown. The

[1] Automatic Speech Recognition
[2] Independent Component Analysis
[3] Principal Component Analysis
[4] Finite Impulse Response filter; a digital filter without a feedback.
[5] Infinite Impulse Response filter; a digital filter with one or more feedback loops (wherefore it can become unstable and produce an unlimited output).
[6] Alternating Current; an electric current whose direction reverses cyclically.

TABLE I
BACKGROUND NOISES AND ACOUSTIC ENVIRONMENTS

Environment	**Closer specification**	**Length** [min:sec]
Diesel-powered car	in city traffic, upto 50 km·h^{-1}	17:23
Diesel-powered car	on motorway, upto 130 km·h^{-1}	22:04
Petrol-powered car	in city traffic, upto 50 km·h^{-1}	11:41
Petrol-powered car	on motorway, upto 130 km·h^{-1}	27:15
Small aircraft	piston engine, level flight	15:11
Diesel-powered bus	in city traffic	14:07
Trolleybus	in city traffic	15:47
Trolleybus	with aux diesel generator	5:21
Tram	modern (2000), in city traffic	20:37
Tram	old (late 1970s), in city traffic	27:06
Railway station hall		13:22
Busy city crossroad	in centre of Plzeň, 167k inhab.	9:37

only way to fight this fundamental lack of knowledge is to *estimate* the features of the noise. If the apriori estimate is close enough to reality (i.e. to the character of the background noise present in the recording) the method proves excellent performance which unfortunately drops dramatically in the case that the estimate is not close enough.

As already mentioned in the abstract and evident from the above said these techniques can be very aggressive to the useful information included in the signal when working with a bad estimate of the noise signal character. Thus the denoising problem can be reduced[7] to a **searching for an acceptable estimate of the noise**.

III. NOISE PROFILES

The fundamental idea of the presented denoising approach is to have a large set of various **clean**[8] background noises.

A set of clean noises was recorded during the research phase of the project: The set incorporates background noises and acoustic environments that have been expected to appear the most often in the recordings fed into our **LASER**[9] ASR system (details in [2]) when deployed in a reasonable real-life application. The background noises were selected by a common agreement of the involved researchers (i.e. at this stage the process was not data-driven). Table I shows the current contents of the set (on which the method was tested). The recordings were made with the portable Sony MZ-RH1 minidisk recorder and the Sennheiser MKE 2 lavalier microphone in high-quality ATRAC mode. The recorded material was processed in a below depicted way to obtain the corresponding *noise profiles*.

[7]As the rest of the process has been solved for years within the field of the digital signal processing.

[8]A "clean noise" in this context means that the recording contains the noise **only** without any speech.

[9]LASER = LICS Automatic Speech Evaluator/Recognizer (where LICS stands for Laboratory of Intelligent Communication Systems)

A. Noise Profile Computation

The noise profile $\mathbf{P}$ (which is necessary for further denoising) is an $N \times N$ matrix of real numbers:

$$\begin{bmatrix} p_{1,1} & \cdots & p_{1,N} \\ \vdots & \ddots & \vdots \\ p_{N,1} & \cdots & p_{N,N} \end{bmatrix},$$

where N is number of frequency bands (points of a power spectrum) used in the spectral analysis of the signal. The elements are:

$$p_{i,j} = \sum_{m=1}^{M} \frac{O(\frac{S_m(i)}{C_{dr}} - j)}{M} \quad (1)$$

$$O(x) = \begin{cases} 1 \text{ if } |x| < 1 \\ 0 \text{ elsewhere} \end{cases} \quad (2)$$

where M is the total number of analysed frames of the signal, $S_m(i)$ is i-th point of a power spectrum of the m-th frame, and C_{dr} is a constant specified below.

The noise profile matrix $\mathbf{P}$ is obtained by executing the algorithm below (put down in a Pascal-like pseudocode):

```
01: let P(1..N, 1..N) = 0
02: for I = 1 to M do
03:    let X(1..N) = power spectrum of the I-th frame
04:    for J = 1 to N do
05:       let S = X(J) / C_dr
06:       if S > N then S = N endif
07:       inc P(J, [S]) by 1/M
08:    endfor
09: endfor
```

where M is the total number of frames of the processed signal, N is the number of power spectrum points, C_{dr} is a dynamic resolution constant with the value 195.3125 for signed short integers (16-bit wide), and [S] means an integer part of a real number S.

The C_{dr} constant is computed so that the maximal energy in the averaged spectrum of the whole analysed material is projected into the N-th row of the $\mathbf{P}$ matrix (i.e. if the maximal energy found in the spectra is 100000.0 and N = 512, $C_{dr} = 100000/512 = 195.3125$).

By its nature, the noise profile records temporal frequencies of occurrence of spectral energies.

B. Noise Profile Selection

As mentioned in section II a successful denoising strategy comprises a knowledge of the noise signal character. Currently a set of clean noise signals (see table I) is available from which the signal features can be easily obtained. Each of these signals has a noise profile computed using the above described algorithm. When denoising a recorded speech signal it is necessary to compute the noise profile for it too[10]. Then the closest of the prerecorded clean noise signals can be find and used to denoise the recording.

[10]using the same algorithm

The closest noise is chosen by **minimising the distance** between the noise profile matrices and the profile matrix of the signal to be denoised. The used metric to be minimised is

$$d(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sqrt{(a_{i,j}-b_{i,j})^2}, \tag{3}$$

where $\mathbf{A}$ and $\mathbf{B}$ are $N \times N$ profile matrices whose distance is being determined.

IV. DENOISING USING THE SELECTED PROFILE

The clean noise profile which is the closest by the given metric to the recorded utterance profile is the best one to determine the noise features from for denoising (although still sub-optimal).

It is self-evident that the noise in a real recording will not be present in the noise corpus. However, it is not necessary. The keystone of the method can be simply expressed by the sentence: "A filtering using a noise profile derived from the closest noise in the corpus must be more efficient than a filtering using a generic profile." The tests (see below) proved it right.

A. Method 1—FIR pass filtering

A character of a background noise can be expressed (for the subsequent denoising) e.g. by an averaged frame power spectrum $\overline{S}$ computed via FFT[11] from frame power spectrum estimates:

$$\overline{S}(i) = \frac{\sum_{m=1}^{M} S_m(i)}{M}. \tag{4}$$

The MATLAB code below shows how to use the averaged power spectrum $\overline{S}$ to denoise the signal using a FIR pass filter:

```
01: pows = pows ./ max(pows);
02: pows = 1 - pows;
03: freq = 0:1/256:1;
04: coeffs = firpm(128, freq(1:256), pows);
05: filtered = filter(coeffs, 1, raw);
```

The `pows` vector contains the averaged power spectrum $\overline{S}$. It is normalized (line 01) and inverted (line 02) to shape a frequency response of the designed FIR filter. A FIR filter is designed using the *Parks-McClellan optimal equiripple FIR filter design method* (see [4]) at line 04. The filter is then applied onto the noised signal (vector `raw`) at line 05.

No matter how simple this approach is (works in real time even in MATLAB) it leads to acceptable results as shown it table IV—mainly because of using the preselected sub-optimal noise profile.

[11] Fast Fourier Transform; an algorithm of the integral transformation of a signal from time to frequency domain.

TABLE II
NOISED DATA DESCRIPTION

Set	Background noise description
`noised001`	Heavy city bus noise with an indistinct background conversation mixed at 0 dB.
`noised002`	Typical squeaking trolleybus noise with intensive background conversation (clearly intelligible) mixed at 0 dB.
`noised006`	Moderate diesel-powered car noise without any background conversation mixed at -6 dB.

B. Method 2—Spectrum Shaping

The second tested method is based upon transforming the whole signal being denoised into the frequency domain by FFT, manipulating the obtained vector of frame power spectra, and resynthesizing the signal via IFFT[12] back to temporal domain.

The manipulation used in the performed experiment was the following:

(i) **Subtraction** of the averaged power spectrum $\overline{S}$ of the preselected clean noise from a power spectrum of each frame of the denoised signal.
(ii) **Gaussian smoothing** (using 2 neighbours at both sides for 512-point spectrum) of the resulting quotient power spectrum of each frame of the denoised signal.

V. RESULTS

The performance was measured (so far) on 3 large sets of recordings, each containing 400 files. The overall length of the testing signal is 16:47. The noised sets `noisedXXX` were prepared using the GNU/Linux batch audio processing tool **Ecasound** released under GPL from `http://www.eca.cx/ecasound/` (as of August 2008).

The clean noise signals were mixed down with the clean speech recordings from the **LAC**[13] **Chess** corpus (a testing set of the LASER ASR system) at given level—see the detailed description in table II.

At first the ability to choose the right noise profile using the method depicted in III-B was tested. As the programmatic execution of the test was carried out in MATLAB, slightly smaller sets were used[14]. For each noised recording, the correct noise source was known and the method was tested whether it selects the same—the achieved accuracy is shown in table III.

When evaluating the performance of the denoising techniques a baseline was set at first: The whole noised material was fed into the LASER ASR system and its performance was measured (referred to as **Baseline**). Then the noised material was processed using method 1 described in IV-A (referred to as **Method 1**) and method 2 described in IV-B

[12] Inverse FFT
[13] LICS Audio Corpus
[14] When trying to test on all 400 files of each set, MATLAB crashed with "Not enough memory" error report.

TABLE III

NOISE PROFILE SELECTION ACCURACY

Set	Size	Accuracy
noised001	380 files	100.0%
noised002	300 files	100.0%
noised006	300 files	98.0%

TABLE IV

ASR PERFORMACE ON NOISED AND DENOISED SIGNAL

Set	Size	Baseline	Method 1	Method 2
noised001	400 files	39.70%	**46.39%**	**42.61%**
noised002	400 files	59.31%	44.17%	57.46%
noised006	400 files	92.94%	**93.24%**	86.25%

(referred to as **Method 2**). The ASR performance on both the noised and denoised material is summarized in table IV. The performance of the LASER ASR system on clean sets was 97.23%.

VI. CONCLUSIONS

The results show a surprisingly accurate performance of the noise profile identification algorithm which is far beyond preliminary expectations. Even though some of the noise signals in the set were very close one to another[15]—for example the trolleybus noise and the noise of the trolleybus with an auxiliary diesel generator—the method was capable to differentiate them and choose the right one correctly. The nearly 100% accuracy urges for further intensive testing.

On the other hand both two presented denoising techniques proved more or less expected performance—units of percent. An interesting result was achieved by **Method 1** on the noised002 set. The resulting accuracy is extremely low compared to the baseline performance of the ASR system. However, the fact (revealed by listening to the denoised recording) is that the method worked very well: It removed the unwanted speech presumed to be a part of the background noise. Unfortunately it was not the background speech but the (foreground) speech to be recognized. The signals were mixed at 0 dB and thus the spoken utterances were recorded with the same intensity from both background and foreground.

To conclude the paper it can be stated that a surprisingly efficient noise identification technique was developed and a promising denoising strategy that does not show a perfect performance but indicates a direction for further research.

VII. ACKNOWLEDGMENTS

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic grant NPV II-2C06009 COT-SEWing.

[15] At least from a subjective point of view after listening to the recordings.

REFERENCES

[1] R. M. Gray and L. D. Davisson: *An Introduction to Statistical Signal Processing*, Cambridge University Press, Cambridge, United Kingdom, 2004. ISBN 0521838606.

[2] T. Pavelka: *Hybrid Methods of Automatic Speech Recognition* (in press), Ph.D. Thesis, University of West Bohemia, Plzeň, Czech Republic, 2008.

[3] K. H. Knuth: A Bayesian approach to source separation, In: J.-F. Cardoso, C. Jutten and P. Loubaton (eds.), *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation: ICA'99*, Aussios, France, 1999, pp. 283-288.

[4] T. W. Parks and C. S. Burrus: *Digital Filter Design*, John Wiley and Sons Canada, 1987. ISBN 978-0471828969.