Fused Models for Noise Reduction in Speech Processing

Nathaniel Ayewah
Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
ayewah@engr.smu.edu

Peter-Michael Seidel
Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
seidel@engr.smu.edu

***Abstract*-We evaluate and compare two approaches for noise reduction in speech processing. One model is a recently developed implementation of a cochlea model that is solved in the time domain and that accounts for the influence of hair cells outside the inner ear membrane. The other approach involves statistical de-noising techniques based on wavelets. Our results indicate that the cochlea model achieves a stronger performance in removing noise, but requires significantly more computational effort. We propose the implementation of a fused model combining features of the cochlea model with wavelet-based techniques to achieve improved noise reduction properties at smaller computational costs.**

I. Introduction

The performance of current speech processing algorithms degrades drastically in the presence of noise. Successful processing of speech signals in noisy environments is of importance in numerous applications, including: medical applications (e.g., hearing aids and cochlea implants), speech recognition, telephony (e.g., hands-free telephones), and military applications (e.g., transmission of speech and surveillance). Our focus is the development of speech processing methods that are amenable to ultra-low power environments such as hearing aids, cochlea implants, and mobile phones.

In this project, we compare and fuse two existing de-noising speech processing methods – one based on a cochlea model and the other based on wavelets – to achieve improved voice signal quality at low computational cost. The cochlea model simulates mechanical properties of the inner ear. The wavelet model provides time-frequency localization of the signal, making it possible to filter out noise.

The proposed hybrid model builds on the qualities of the cochlea model and the computational performance of the wavelet models. It observes the response of the cochlea model to different wavelets and efficiently approximates the output of the cochlea model based on these observations.

II. OHC Model

A. The Structure of the Cochlea

The cochlea is a spiral shaped structure located at the inmost section of the ear (Fig. 1). It contains two fluid-filled chambers separated by an elastic partition which contains the *basilar membrane* (BM). Sounds traveling through the chambers cause the BM to vibrate. This vibration stimulates auditory nerve fibers which carry signals to the brain [1].

There are around 15,000 hair cells attached to the BM (in the human ear). They are divided into two groups based on their functionality. One group, the *inner hair cells*, are connected to the nerve fibers. They transfer the mechanical motion of the BM into neural activity in the nerve fibers. The rest of the hair cells are called *outer hair cells* (OHC). They act as local-amplifiers that influence the mechanical response of the membrane to produce "high sensitivity and sharp tuning" [1]. This action allows the cochlea to adapt to the dynamic range of the incoming signal. Hearing-impairment has been associated with the loss of outer hair cells [2].

B. Modeling the Cochlea

Cochlea models are useful for predicting the behavior of the cochlea. They simulate the vibration of the BM using equations that apply basic physical principles such as the conservation of mass and the dynamics of deformable bodies. Reference [2] provides details of an implementation of a cochlea model which specifically models the mechanical response of the BM taking into account the action of OHCs. We refer to this model from here on as the *OHC Model*.

The OHC Model is based on differential equations derived by considering the pressure on the BM as a result of the stimulating speech signal. Generally the pressure difference between the two chambers shown in Fig. 1 is

$$P = P_T - P_V, \tag{1}$$

where P_T and P_V are the pressures in the *scala tympani* and

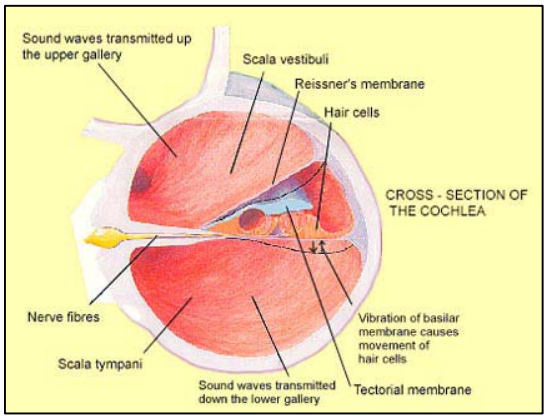

Fig. 1. Cross Section of a Cochlea [2]

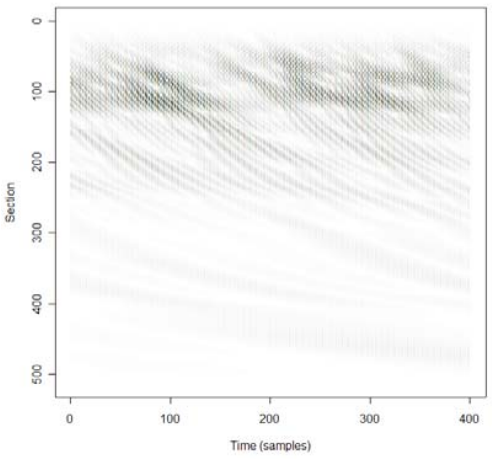

Fig. 2. Visualization of LBTMV matrix

scala vestibulli respectively. The OHC Model introduces the behavior of the outer hair cells to modify the pressure on the BM:

$$P_{BM} = P + P_{OHC}. \tag{2}$$

The important output from the model is a matrix – $\xi_{BM}(x, t)$ – which represents the displacement of different locations x on the BM over time t. This is used to calculate the velocities along the membrane which can serve as a representation of the speech signal stimulus. We refer to this as a *location-time basilar membrane velocity* representation (LTBMV-representation). Integrating the velocities along the membrane reconstructs the original speech signal with reduced noise.

Fig. 2 shows a visualization of the matrix of displacements. It shows that different sections of the BM resonate to different frequencies. The lower section numbers represent the *base* of the BM while the higher section numbers represent the *apex* of the BM. The signal arrives at the base from the outer ear and travels toward the apex. Higher frequencies resonate near the base while lower frequencies produce maximal displacement nearer the apex [1]. The outer hair cells help to reduce noise by essentially damping certain frequencies based on the dynamic range of the signal.

C. OHC Model Performance

The OHC model uses a variable step trapezoidal method to solve the differential equations. This requires significant computational effort to execute multiple iterations for each time-step, improving the accuracy in each iteration. The OHC takes about 1000 seconds to process a 1-second speech signal. In the process, it goes through over 6 million iterations and performs over 200 million binary operations. Later, we will show that we can reduce this effort significantly by using wavelet analysis to approximate the LTBMV matrix.

III. Wavelet Analysis

A. Features of Wavelets

A wavelet is a "small wave" with its energy concentrated in time [3]. Wavelets are used as basis functions for representing or decomposing signals, just as sinusoids are used as basis

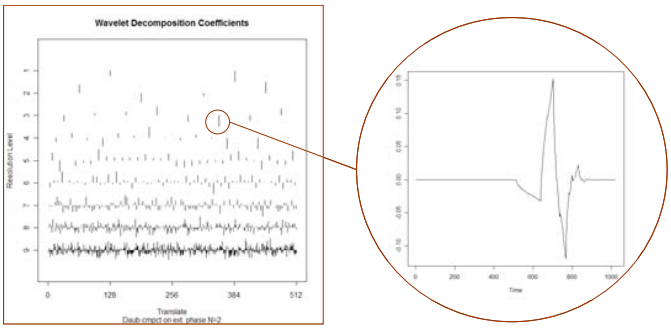

Fig. 3. A wavelet coefficient can be arranged on a level and position based on how its corresponding wavelet is scaled and shifted. Higher resolution levels capture finer details while lower levels indicate broad trends.

functions in Fourier analysis. Fourier analysis is useful for studying the frequency content of a signal. It results in a one-dimensional array of coefficients which is *localized* in terms of frequency. This analysis allows to represent a long periodic signal using only a few frequency coefficient values.

Sinusoids have infinite energy and hence do not allow for time localization in analyzing signals. Wavelets, on the other hand, have finite energy concentrated around a center and hence help uncover time-varying phenomena in the signal. Wavelet analysis results in a two-dimensional array of coefficients which is localized in terms of time and frequency. The original signal is reconstructed precisely by summing shifted and scaled versions the original basis wavelet (Fig. 3) weighted by the wavelet coefficients..

B. Applications of Wavelets

Probably the most poignant observation in wavelet analysis is that most of the energy of the signal is compacted into a few coefficients. This leads to applications in signal compression and denoising [4]. In both cases, small low energy coefficients are removed using a *thresholding* method leaving a compressed version of the signal. This *lossy* compression technique has been used in speech and image compression (e.g. the JPEG-2000 standard).

Our interest is in denoising signals. When a signal with *additive* noise is analyzed using wavelets, most of the noise is relegated to small low energy coefficients [4]. The reduction of these low energy coefficients using a thresholding method leads to reduced noise in the signal.

There are generally two thresholding methods: hard and soft thresholding. In both cases, the first step is to select a threshold value, T_n. In hard thresholding, all coefficients with magnitude below T_n are simply set to zero. In soft thresholding, each coefficient, c, is changed to

$$\text{sgn}(c) \times \max(|c| - t, 0), \tag{3}$$

where sgn(c) is the sign of the coefficient [5]. In general, the threshold value, T_n, is chosen to be a multiple of the standard deviation of the coefficients. Parameter choices are usually made based on assumptions on the type of noise involved and on the percentage of the energy being removed.

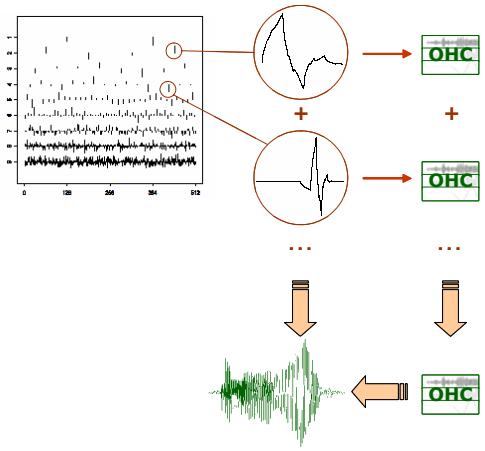

Fig. 4. Adding the matrices generated from individual wavelets results in a final matrix that can be used to reconstruct the output of the OHC Model.

The effectiveness of thresholding depends on how well the wavelet compresses the energy of the original signal. Different wavelets are more or less suitable for different types of signals. References [3] and [4] discuss a variety of choices.

C. *Wavelet Performance*

In our experiments, we use an implementation called *wavethresh* described in [5] and created for the R-statistical environment [6]. Wavethresh implements the efficient *Pyramid Algorithm* described by S. Mallat [7] for the computation of the wavelet coefficients with linear time complexity. For comparison with the OHC Model, the wavelet model de-noises the same 1-second signal described in Section II in less than 1 second.

IV. Fused Model

A. *Method Formulation*

We formulate our method by studying the response of the OHC model when single normalized basis wavelet functions are chosen as input signals. An obvious observation is the way the frequency and time localization features of a wavelet are captured in the LTBMV matrix (Fig. 5-b). A more interesting observation is that the linearity property of wavelets seems to be captured in the matrix (Fig. 4). In other words, scaling and adding two wavelets corresponds to scaling and adding their respective matrices. (Shifting does not correspond as accurately, but we use it later to achieve some optimizations).

This observation leads to a simple formulation. As a preprocessing step, we propose to create a *characteristic* matrix for every coefficient position. (This can be done by using a single non-zero wavelet coefficient in an inverse wavelet transform (reconstruction), and then providing the result (time representation of a single normalized wavelet) to the OHC Model.) Then, during signal processing, a given signal is decomposed using wavelet analysis and the resulting coefficients are used to scale their corresponding characteristic matrices. Finally, the scaled characteristic matrices are added to produce the final LTBMV-representation of the de-noised signal. Initial experiments demonstrated that this method replicates the output of the OHC Model with very little error.

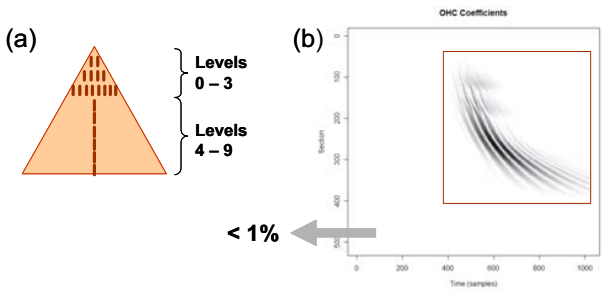

Fig. 5. Observations lead to optimized formulation in which high resolution levels only have one matrix which is shifted (a) and a subset of the matrix is used with small values removed (b).

B. *Method Shortcomings and Modifications*

The problem with the formulation described above is that it requires a large amount of space. For example, a signal with 1024 samples generates a 1023×512 matrix which requires about 4 MB. Since the characteristic matrix must be generated from a wavelet in the same resolution (length) as the intended signal, each characteristic matrix is also 4 MB. The wavelet decomposition of the signal yields 1023 coefficients. As a result, our formulation requires 4 GB of storage space to hold all the characteristic matrices! This massive amount of data complicates implementation and makes a direct implementation inappropriate for real-time, low power environments.

We are proposing to deal with the challenges of the direct implementation based on two observations. The first is that shifting a wavelet corresponds approximately to shifting the characteristic matrix. This approximation is less accurate at low resolution levels where the wavelet is longer and generates a response through most of the matrix. But at high resolution levels, where the wavelet is very brief and generates a very short response, this approximation is very accurate. We modify our method by adding a new feature – the *shifting levels*. In each of these levels, a single matrix corresponding to the coefficient at the midpoint of the level is generated and shifted. This allows us to reduce the number of matrices in our 1024-sample example from 1023 to 21 with shifting levels 4 – 9 (Fig. 5-a).

The second observation is related to the first one. Considering that higher resolution level coefficients yield short responses in the LTBMV matrices, we observe that most of the matrix contains insignificant values that are less than 1% of the largest value in the matrix. We ignore these values and store a smaller matrix that contains more significant values (Fig. 5). This reduces the size of the matrices in our 1024-sample example from 1023×512 to 666×411 (2.1 MB) at level 4 and 95×105 (78 KB) at level 9.

These two modifications (shifting and thresholding) allow the reduction of the storage requirement for our 1024-sample example from 4 GB to 64 MB – a 63-fold improvement! Of

course for longer signals larger characteristic matrices would be needed. This raises another problem: how can our formulation be changed to work with arbitrary length signals? (The length of the signal dictates the size of the characteristic matrices.)

Experiments indicated that simply concatenating the results of 1024-sample segments to create one large segment shows poor results. This is because, as is shown in Fig. 2, the response of a signal in the matrix "travels" down the BM from the base to the apex. Breaking a signal into disjoint segments does not reflect this flow. However, we observe that breaking the signal into *overlapping* segments better accounts for this phenomenon. This final modification is illustrated in Fig. 6.

C. Implementing the Fused Model

The results in the next section demonstrate that the fused model formulation is comparable to the OHC Model in terms of accuracy, but requires considerably less computational effort. One issue that arises in implementing the method is how to determine the parameters and their impact on the model's performance. Some of our parameter choices were arbitrary or as a result of observation. More work will need to be done to determine guidelines for optimal choices.

Many of the parameters have to be set during the preprocessing step and cannot be changed in real time. These include decisions on the use of the wavelet filter for creating characteristic matrices, the segment lengths, the choice of shifting levels, the thresholding methodology for the characteristic matrices and all parameters of the OHC Model (including the sampling rate of the signal). Parameters that can be chosen at run time include for example the overlap and the wavelet thresholding method (if the wavelet decompositions for the noisy signal are to be thresholded).

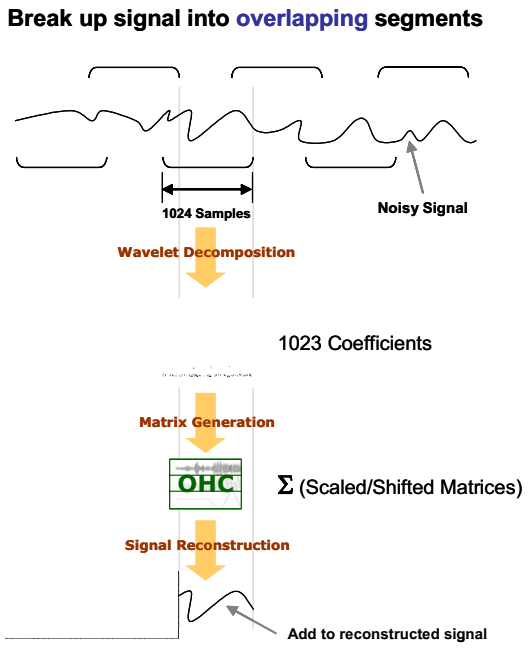

Fig. 6. In the final formulation, the signal is divided into overlapping segments. The non-overlapping portion of each reconstructed signal is concatenated to the reconstruction derived from previous segments.

V. Results

A. Performance Measures

The primary measure for evaluating our model outputs is the Signal-to-Noise Ratio (*SNR*) measured in decibels (*dB*) and defined as:

$$SNR = 10\log_{10}\left(\frac{P_S}{P_N}\right), \tag{4}$$

where P_S and P_N are a measure of the power in the signal and the noise respectively. In the experiments that follow, we focus on the increase in *SNR* (SNR↑):

$$\text{SNR}\uparrow = SNR(s_{denoised}) - SNR(s_{noisy}), \tag{5}$$

where s_{noisy} is the original noisy signal and $s_{denoised}$ is the output of one of the models. We also consider the root mean square error (*RMS*):

$$RMS = \sqrt{\frac{E_N}{n}}, \tag{6}$$

where E_N is the energy of the noise in a given signal. (The noise in a signal is derived by subtracting the original *noise free* signal from the given signal.) To appreciate how much noise is being removed, we introduce a *Noise Reduction Ratio* (NRR) based on the *RMS* error:

$$\text{NRR} = RMS(s_{noisy}) / RMS(s_{denoised}), \tag{7}$$

For example if the RMS error is reduced from 0.2 to 0.1, then NRR is 2 and half of the noise has been removed.

We also observe the running times of these models. Other evaluation criteria which are not included in this report include:

- perceived audio quality in psychological experiments,
- latency between when a signal enters a model and when an output can be perceived, and
- size and power consumption of hardware implementation.

B. Experiment Parameters and Results

We experimented on two speech signals described in Table I. The signals were contaminated with two kinds of noise. *White noise* contains all the frequencies a person can hear in approximately equal amounts. It sounds like it has more high frequency content because higher octaves have more frequencies than lower octaves. *Pink noise* compensates for this by reducing the volume of frequencies at higher octaves.

In our experiments, we use a Daubechies wavelet provided by wavethresh – "DaubLeAsymm", "filter 9" – with "soft" thresholding of all coefficients below 85th percentile [5]. (This wavelet gave the best results among all the wavelets provided by wavethresh in initial experiments.) The Fused model uses this wavelet to generate the characteristic matrices and to decompose the signals to be processed. We run the Fused model twice. In the first run we use all the wavelet

decomposition coefficients for the given signal to calculate the final LTBMV-matrix. In the second run we threshold the coefficients (using the method used in the wavelet model).

Tables II and III summarize the results. The best results occur when the wavelet decomposition of the signal is thresholded before running the Fused Method. Fig. 7 and Fig. 8 illustrate that the Fused Model (without thresholding) generates results that are very similar to the output of the OHC model. But, as Table III shows, the Fused Model is up to 23 times faster than the OHC Model, although the implementation has not yet been optimized and is run within the R-stat interpreter environment.

VI. Applications and Conclusions

Noise reduction tools have many applications in speech processing as mentioned in the introduction. One important application is in hearing aids. Hearing aids are the primary tool for assisting individuals with hearing loss due to a damaged cochlea. But since hearing aids amplify the sound without regard to the level (volume) of the original signal, background noise also gets amplified. Improving the SNR is important to make hearing aids more comfortable. It has been shown that improving the SNR by 1 dB can improve intelligibility by 7% to 19% [8].

The Fused model can be used in concert with other techniques such as binaural processing and speech recognition to provide enhanced speech processing. However, more research is needed to decide the parameters that deliver optimal performance and results. Research is also needed to evaluate the complexities of implementing this model in hardware and performing real time speech processing.

Table I

Summary of Signals Used in Experiments

	Duration	Sample Rate	Samples
Short Signal	476 ms	44.1 KHz	20,993
Long Signal	2.374 sec	44.1 KHz	104,704

Table II

Results of 3 Experiments

	Experiment 1[a] Short Signal, White Noise		Experiment 2[a] Short Signal, Pink Noise		Experiment 3[a] Long Signal, White Noise	
	SNR↑	NRR	SNR↑	NRR	SNR↑	NRR
OHC	8.78	2.75	3.84	1.56	5.71	1.93
Wavelet	7.51	2.39	0.19	1.02	5.17	1.89
Fused (no thresholding)	8.29	2.60	5.04	1.79	5.56	1.90
Fused (with thresholding)	**12.50**	**4.26**	**5.26**	**1.83**	**7.67**	**2.51**

[a]Each experiment is conducted 5 times with varying amounts of noise. The average SNR↑ and NRR is recorded.

Table III

Model Running Times

	Wavelet	OHC	Fused
Short Signal	< 1 sec	509 sec	22 sec
Long Signal	2 sec	2438 sec	106 sec

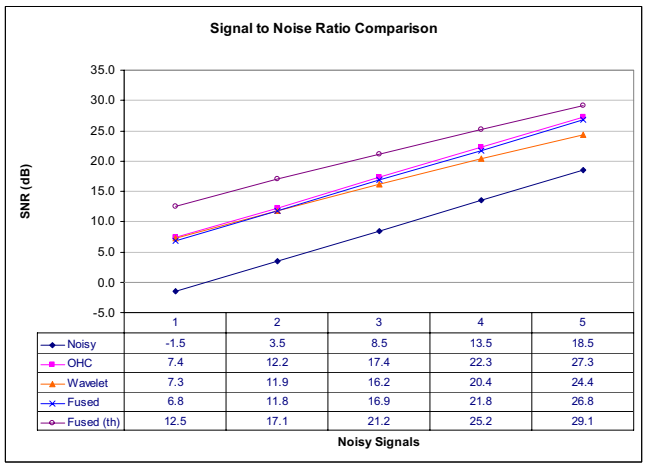

Fig. 8. Experiment 1: Detailed SNR Results

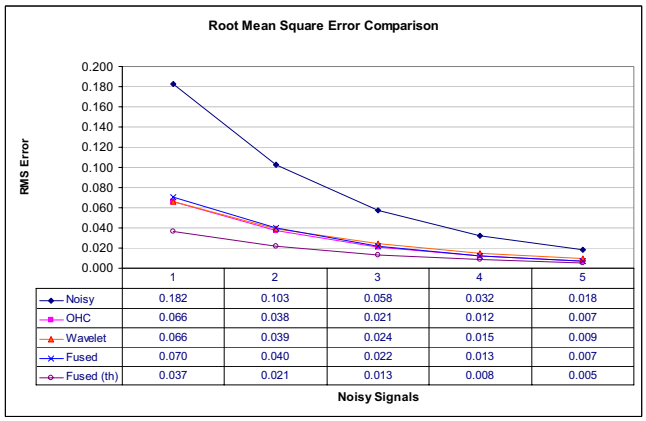

Fig. 8. Experiment 1: Detailed RMS Error Results

Acknowledgment

The authors would like to thank A. Cohen and M. Furst – Department of Electrical Engineering-Systems at Tel-Aviv University, Tel-Aviv, Israel – for providing access to their implementation of the OHC Model.

References

[1] B.C.J. Moore, *An Introduction to the Psychology of Hearing*, Academic Press, London, 2003, pp. 1-53.

[2] A. Cohen and M. Furst, "Cochlear Model with Outer Hair Cells", Technical Report, Department of Electrical Engineering-Systems, Tel-Aviv University, 2003.

[3] C. Burrus, R. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, New Jersey, 1998.

[4] J. Walker, *A Primer on Wavelets and their Scientific Applications*, CRC Press, Boca Raton, FL, 1999.

[5] G.P. Nason, A. Kovac, and M. Maechler, *The wavethresh Package*, http://cran.us.r-project.org/, 2003.

[6] K. Hornik, *The R FAQ*, http://www.ci.tuwien.ac.at/~hornik/R/, ISBN 3-901167-51-X, 2003.

[7] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, July, 1989.

[8] B.C.J. Moore, *Cochlear Hearing Loss*, Whurr Publishers, London, 1998.