Audio Denoising Algorithm by Adaptive Block Thresholding using Short Time Fourier Transform

[1] Shivam Tiwari, [2]Rashika Gupta, [3]Ravimohan
1 Research Scholar, M. Tech, 2Assistant Professor, 3Professor & H.O.D.
Dept. of ECE, SRIT, Jabalpur

Abstract: **In this paper, an adaptive bock algorithm with modified threshold using STFT for denoising of audio signals is proposed. The signal is first segmented into multiple blocks depending upon the minimum mean square criteria in each block, and then thresholding methods are applied for each block. All the obtained blocks after denoising of individual block are then concatenated in order to get the denoised signal. The Short Time Fourier Transform provides more coefficients than the Discrete Wavelet Transform (DWT), representing additional time-frequency details of the signal. STFT gives more degree of freedom in terms of time-frequency resolution of audio signals. When an audio signal corrupted with Additive White Gaussian Noise (AWGN) is processed by using this algorithm, we obtained higher Signal to Noise Ratio (SNR). Hence, the proposed algorithm out-performs the other algorithms. Also the obtained denoised signal with this algorithm is close to the original signal.**

***Keywords*: STFT, Adaptive Block Selection, AWGN, SNR, Thresholding.**

1. Introduction

In the field of denoising the sounds of musical instruments, time frequency based transforms play an important role. They allow us to work with a sound signal from both time and frequency perspectives simultaneously. Such transforms have traditionally been useful in studying the nature of the sound signal, noise, and in facilitating the application of aesthetically interesting and novel modification to specific sound signals [1]. Conventional Fourier analysis pioneered by Fourier in 1807 is a powerful tool to decompose a time-domain signal into separate frequency components and the relative intensities of the individual frequencies are shown in the representation [1]. However, the temporal behavior of the signal's frequency components is unknown in the conventional Fourier analysis. Unlike the conventional Fourier frequency domain, the joint time-frequency (TF) domain provides a convenient platform for signal analysis by involving the dimension of time in the frequency representation of a signal. A straightforward way to acquire localized knowledge about the frequency content of the signal at different times is to perform the Fourier transform over short-time intervals rather than processing the whole signal at once. The resulting TF representation is the short-time Fourier transform (STFT) [2], which remains to date the most widely used method for the analysis of signals whose spectral content varies with time. Recent application examples of the STFT and its variants – e.g. the squared magnitude of the STFT known as the spectrogram – include signal denoising [3],[4], instantaneous frequency estimation [5],[6], and speech recognition [7]. Let's take a look at a real-life example to illustrate the needs for the joint TF representations. Using a bat echolocation sound [8], Fig. 1.1 shows three different plots. By only looking at the time domain plot of the signal at the bottom, we can only see how the intensity or loudness varies with time. On the left of the main plot is the energy density spectrum, that is, the squared magnitude of the Fourier transform of the time signal. It indicates the relative intensity of each frequency components. The power spectrum tells us that the frequency samples mainly range from 50 to 200 samples, but it does not show when these frequencies happen. The main plot is a joint TF plot represented using the contour plot. From it, we can see the frequencies and their relative intensities happened at different time. For instance, the bat signal mainly consists of three nonlinearly FM chirps components overlapping in time.

A noise reduction method based on STFT coefficients contraction [15] basically consists of three steps;

1. Apply STFT to noisy signal as;

$$S.y = S.s + S.z \qquad \text{Eq. (1.1)}$$

Where; y, s, z & W are the resultant noisy audio, clean audio signal, noise signal & the matrix associated to the STFT respectively.

2. Thresholding is done for the obtained transformed coefficients.
3. The desired signal is reconstructed by applying the inverse STFT to the thresholded STFT coefficients.

2. Short Time Fourier Transform (STFT)

2.1 The Continuous STFT

The continuous Short-Time Fourier Transform (STFT) analysis of a signal $x(t)$ can be obtained as [1]:

$$X_{STFT}(t,\omega;h) = \int h^*(t-\tau)x(\tau)e^{-j\omega\tau}d\tau \qquad \text{Eq. (2.1)}$$

Where, $h(t)$ is the analysis window which determines the portion of $x(n)$ being analyzed. The analysis window may be chosen to be real or imaginary, but it is typically chosen to be a real and symmetric function centered at zero, tapering off to zero away from its centre. Hence, at each time instant, (2.1) computes the Fourier transform of a short portion of the signal around t and thus the STFT can be regarded as a local spectrum of the signal $x(\tau)$ around time t. Accordingly, the short-time energy-density spectrum $S(t,\omega;h)$ can be obtained as the squared magnitude of (1) i.e. $S_x(t,\omega;h) = |X_{STFT}(t,\omega;h)|^2$ and is commonly called the spectrogram. When a unit-energy window is used then the total energy of the spectrogram equals that of the signal.

There exists an elemental relationship between the spectrogram and the Wigner distribution (WD);

$$W_x(t,\omega) = \int x\left(t+\frac{\tau}{2}\right) x^*\left(t-\frac{\tau}{2}\right) e^{-j\tau\omega} d\tau \qquad \text{Eq. (2.2)}$$

Namely, the convolution of the WD of the signal with the WD of the STFT window renders the spectrogram, i.e.

$$S_x(t,\omega;h) = \iint W_x(t',\omega')\, W_h\,(t-t',\omega-\omega')\, dt'd\omega' \qquad \text{Eq. (2.3)}$$

The WD is widely recognized for its high TF concentration [2],[3]. On the other hand, it suffers from the presence of cross-term interference which limits its readability, and prevents it from being strictly positive. Although the process in (2.3) is known to eliminate interference and restore positivity, it also smears the signal in the TF plane. So, despite its relative advantages, the spectrogram yields inferior TF signal localization compared to the WD. This deterioration depends on the smoothing kernel $W_h(t,\omega)$ of the two-dimensional convolution operation in (2.3). Since the Gaussian function exhibits the least amount of spread in the TF plane [4], it often is the preferred window type for STFT-based signal analysis [5].

3. Audio Signal Denoising

One of the key signal processing operation deals with removal of noise from signal and it seems to be a major problem. An unwanted signal gets superimposed over a clean and undistorted signal. How can we remove the superimposed signal without deterioration of original clean signal? Several algorithms have been developed for efficient removal of noise in various applications [9]. The method applied to denoise the signal [10] gets more sophisticated if the regularity of noise lessens. When signals pass through equipments and communicating medium, the noise is added naturally which results in signal contamination. "It is difficult to remove this unwanted noise without altering or degrading the original signal. Hence, the basic task in signal processing [9] is to denoise the audio signal with minimum degradation of the original signal. The major cause for pollution in audio signals is humming noise from audio equipments or buzzing and background environment noise" [14]. Hence, attenuation of noise while recovering the underlying signals is the primary objective of audio denoising.

3.1. Thresholding Based Denoising

The threshold is not ideal for musical instrument sound signals because of inappropriate correlation between the MSE & subjective quality & the more realistic presence of correlated noise. In this work a new time frequency dependent threshold estimation method is used. In this method firstly the standard deviation of the noise, σ is calculated for each block. For given σ, threshold for each block is calculated. Noise component removal by "thresholding the transformed coefficients is based on the observation" that in the signals, "energy is mostly concentrated in small number of transformed dimensions. The coefficients of these dimensions are relatively very large compared to noise that has its energy spread over a large number of coefficients. Hence by setting smaller coefficients to be zero, we can optimally eliminate noise while preserving important information of the signal". In transformed domain, noise is characterized by smaller coefficients, while signal energy is concentrated in larger coefficients. This feature is useful for eliminating noise from signal by choosing appropriate threshold. Generally, "the selected threshold is multiplied by the median value of the detail coefficients at some specified level which is called threshold processing".

At each level of decomposition, "the standard deviation of the noisy signal is calculated. The standard deviation σ is given by" [12]:

$$\sigma_j = \frac{median\,(|c_j|)}{0.6745} \qquad \text{Eq. (3.1)}$$

Where C_j are high frequency transformed coefficients at j[th] level of decomposition, which are used to identify the noise components & σ_j is Median Absolute Deviation (MAD) at this level. "This standard deviation can be further used to set the threshold value based on the noise energy at that level. The modified threshold value is given by" [7]:

$$T_h = k * \sigma_j \sqrt{2log\left(L_j log_2 L_j\right)} \qquad \text{Eq. (3.2)}$$

Where T is threshold value, LM is the length of each block of noisy signal & k is the constant whose value is varying between 0-1. For determining the optimum threshold, value of k should be estimated.

The "thresholding function also called shrinkage function is categorized as hard thresholding & soft thresholding function. The hard thresholding function is defined as to retain the transform coefficients which are greater than the threshold λ & sets the rest coefficients to zero". The hard thresholding is defined as;

$$f_h(x) = \begin{cases} x, & |x| \geq \lambda \\ 0, & otherwise \end{cases} \qquad \text{Eq. (3.3)}$$

The choice of threshold λ is dictated by the signal energy & "the standard deviation σ of the noise". If the transform coefficient is greater than λ, then it is retained assuming that it contributes to the original signal; else, it is discarded because, in general, it is observed that high frequency noise contributes significantly to such transform coefficients. The "soft thresholding function shrinks the transform coefficients by λ towards zero". Hence this function is also called as shrinkage function. The soft thresholding function is defined as;

$$f_s(x) = \begin{cases} x-\lambda, & x \geq \lambda \\ 0, & -\lambda < x < \lambda \\ x+\lambda, & x < -\lambda \end{cases} \qquad \text{Eq. (3.4)}$$

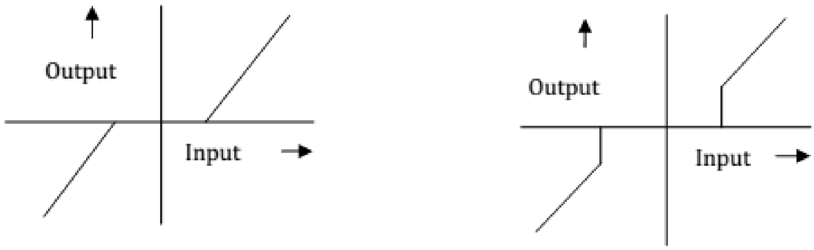

Figure 3.1: (a) Soft thresholding; (b) Hard thresholding

It is shown in figure 3.1 that "hard thresholding function is discontinuous at | x| = λ, due to this discontinuity hard thresholding function results in abrupt artifacts in denoised signal, especially when the noise level is significant. It has been shown" [10], that the soft thresholding gives lesser mean square error in comparison to hard thresholding & hence preferred over hard thresholding; but for sound signals, hard thresholding gives lesser amount of mean square error (MSE).

Thresholding gives amplitude separation. To well separate signal and noise, thresholding is used. The purpose of a filter is for frequency separation and frequency signal restoration. So for amplitude separation thresholding is used [5]. Depending upon the type of noise present in the signal, he thresholding is determined basically in two forms is Soft Thresholding and Hard Thresholding. In Soft thresholding the coefficients which are within the Threshold value are consider as zero and subtract the Threshold value from the coefficients which are above the Threshold value. Depending upon the changes in the noise signal threshold value will change in soft thresholding. In Hard Thresholding , the coefficients which are within the Threshold value are consider as zero and the coefficients which are above the Threshold value remain same and are considered as actual coefficients of the signal. In hard thresholding the threshold value is fixed.

3.2 Block Selection

Most of the musical instrument sound signals are far too long to be processed in their entirety; for example a 8 second sound signal sampled at 11 KHz will contain 11,000 samples. Thus, as with spectral methods of noise reduction, it is necessary to divide the time domain signal in multiple blocks and process the each block individually. The block formation of the signal is shown in the Figure 3.2. The important task is to choose the block length. Berger *et al.* [14] shows that, blocks which are too shorts fail to pick important time structures of the signal. Conversely, blocks which are too long miss cause the algorithm to miss the important transient details in the sound signal. Because of binary splitting nature of STFT, it is good to decompose the signal by selecting appropriate length of each block of the power of two.

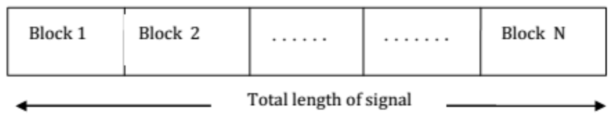

Figure 3.2: Block formation of signal

As discussed previously, the block size chosen must strike a balance between being able to pick up important transient detail in the sound signal, as well as recognizing longer duration, sustained events.

3.3. Threshold Selection

Donoho and Johnstone derived a general optimal universal threshold for the Gaussian white noise under a mean square error (MSE) criterion described in [12]. However this threshold is not ideal for musical instrument sound signals due to poor correlation between the MSE and subjective quality and the more realistic presence of correlated noise. Here we use a new time frequency dependent threshold estimation method. In this method first of all the standard deviation of the noise, σ is calculated for each block. For given σ, we calculate the threshold for each block. Noise component removal by thresholding the STFT coefficients is based on the observation that in musical instrument sound signal, energy is mostly concentrated in small number of STFT dimensions. The coefficients of these dimensions are relatively very large compared to other dimensions or to any other signal like noise that has its energy spread over a large number of coefficients. Hence by setting smaller coefficients to be zero, we can optimally eliminate noise while preserving important information of the signal. In STFT domain noise is characterized by smaller coefficients, while signal energy is concentrated in larger coefficients. This feature is useful for eliminating noise from signal by choosing the appropriate threshold. Generally the selected threshold is multiplied by the median value of the detail coefficients at some specified level which is called threshold processing.

3.4 Choice of Thresholding Level λ:

Given a choice of block size and the residual noise probability level δ that one tolerates, the thresholding level λ .For each block width and length, λ is estimated using "Monte Carlo simulation". The partition of macro blocks in to blocks of different sizes is as shown below:

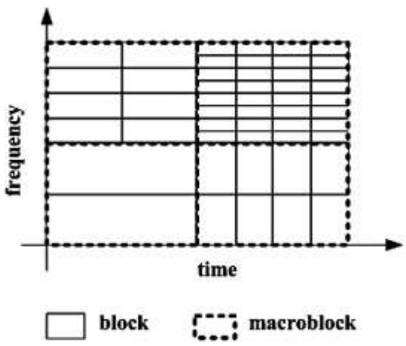

Figure 3.3: Partition of macro blocks

The adaptive block thresholding chooses the sizes by minimizing an estimate of the risk. The risk cannot be calculated since is unknown, but it can be estimated with Stein risk estimate. The adaptive block thresholding groups coefficients in blocks whose sizes are adjusted to minimize the Stein risk estimate and it attenuates coefficients in those blocks [10]. For audio signal denoising, an adaptive block thresholding non-diagonal estimator is described that automatically adjusts all parameters. It relies on the ability to compute an estimate of the risk, with no prior stochastic audio signal model, which makes this approach particularly robust. Thus an adaptive audio block thresholding algorithm that adapts all parameters to the time-frequency regularity of the audio signal. The adaptation is performed by minimizing a Stein unbiased risk estimator calculated from the data. The resulting algorithm is robust to variations of signal structures such as short transients and long harmonics. The coefficients (soft/hard thresholding}. The adaptive block thresholding chooses the Block sizes by minimizing an estimate of the risk.

4. Proposed Audio Denoising Algorithm

The proposed STFT based block denoising algorithm for reduction of white Gaussian noise is explained in the following steps:

1. Choose a sound signal of suitable length.
2. Add "White Gaussian Noise" to the original signal depending upon the standard deviation N.
3. Divide the noisy signal into blocks of different length and depending upon the length of the signal in time domain; preferably, number of samples, N, 2M where M is an integer.
4. Calculate mean square error (MSE) for each block.
5. Optimal block is the one resulting in minimum mean square error.
6. Compute the "Short Time Fourier Transform (STFT)" of one block of the noisy signal at level 1.
7. Estimate the standard deviation of the noise using (8) and determine the threshold value using (9), then apply the different thresholding techniques for time and level dependent STFT coefficients using (6) and (7).
8. Take inverse "Short Time Fourier Transform (STFT)" of the coefficients obtained through step 7, which has reduced noise.
9. Calculate "mean square error (MSE), peak signal to noise ratio (PSNR)" for de-noised signal.
10. Repeat steps 4 to step 8 for other level of decomposition.
11. Concatenate all the blocks of the de-noised signals obtained through step 9 and do averaging operation for MSE and SNR of the sound signal.

Detailed Description of Proposed Denoising Algorithm Steps:

i Determine the SNR of the musical noise signal.

Signal to noise ratio value calculate = $\frac{Time\ Window}{1000}$ x $\frac{Signal\ Power}{Noise\ Power}$.

ii Apply Hanning window

STFT Transformed components are divided into windows as:

Window size = $(\frac{Time\ Window}{1000}) * Sampling\ frequency$

If window size is even Even window = (window size + 1)

If half size window calculate = $(\frac{window\ size-1}{2})$.

No of windows calculate = floor $(\frac{Lenght}{window\ size})$*2

iii Apply half overlapped window

1. Y=Overlap Add (X,A,W,S);
2. Y is the signal reconstructed signal from its spectrogram.
3. X is a matrix with each column being the STFT (FFT) of a segment of signal.
4. A is the phase angle of the spectrum which should have the same dimension as X.
5. If it is not given the phase angle of X is used which in the case of real values is zero (assuming that it's the magnitude).
6. W is the window length of time domain segments if not given the length is assumed to be twice as long as STFT (FFT) window size.
7. S is the shift length of the segmentation process for example in the case of non-overlapping signals it is equal to W and in the case of 50% overlap is equal to W/2. If not given W/2 is used.

iv Find STFT coefficients

1. Find out the STFT coefficient

2. STFT coefficient = zero's (window size, no of window – 1)
3. We calculate STFT coefficient small window
4. Apply FFT and calculate small windows.

v Determine The Block Thresholding

1. Block Thresholding can be apply (noisy signal, time window, frequency sampling, and sigma noise).
2. Block attenuation with bi-dimension (time & frequency).
3. Block size selected by using section, SURE algorithm.
4. Each block considered macro block. All macro blocks are same size.
5. Macro block size = Lmax * Wmax.
6. We calculate the maximum block length & width.
7. Find the lambda value of each block.
8. Each macro block divided small blocks same size.
9. Calculate length & width of each block find the lambda value.

vi Applying Thresholding Each Block.

1. We apply hard Thresholding each block
2. Hard Thresholding calculate Ɛ above value.
3. Ɛ below value can be considered error.

vii Apply Inverse STFT.

1. Inverse STFT converted time frequency domain signal to time domain signal.
2. Window size = $\frac{Time\ Window}{1000}) * Sampling\ frequency$
3. if window size is odd

Odd window = (window size + 1)

Half size window calculates = $\frac{window\ size-1}{2}$ Length.

4. No of windows calculate = floor ($\frac{Lenght}{window\ size}$)*2

viii Compare the SNR of the musical noise signal and the denoised signal.

1. Apply inverse STFT and calculate.
2. STFT coefficient of different blocks are added.
3. Get the reconstructed signal.
4. Error can be calculate = original signal – reconstructed signal
5. If the error is small. The reconstructed signal is better.

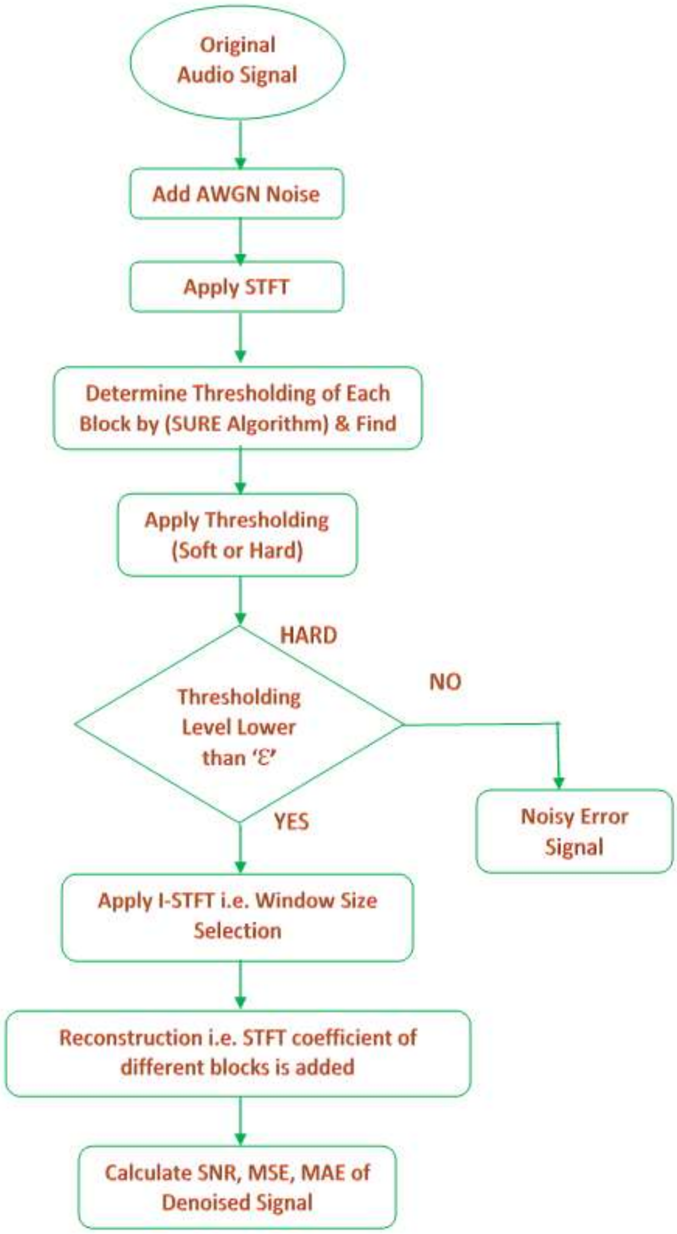

Figure 4.1: Proposed STFT based Adaptive Block Thresholding Audio Denoising Algorithm

5. Results and Discussions

The denoising algorithm developed in the previous section is applied to the sound samples of the various audio signals. Here in this work Mozart.wav is sampled at 11,000 samples per second. For experimental purpose the we have simulated Mozart.wav 5 dB, 10dB, 15 dB, 20 dB & 25 dB. All noisy signals are denoised by our proposed algorithm gives better results than previous methods.

For comparing the performance and measurement of quality of denoising, the Signal to Noise Ratio (SNR) is determined between the original signal S_i and the signal denoised S_d, by our algorithm.

$$SNR = 10log_{10}\left(\frac{S_{max}}{MSE}\right)^2 \qquad \text{Eq. (5.1)}$$

Where $S_{\max}$ is the maximum value of the signal and is given by,

$$S_{max} = \max(\max(S_i), \max(S_d)) \qquad \text{Eq. (5.2)}$$

And MSE is mean square error, given by:

$$MSE = \frac{1}{N}\sum_{l=1}^{N}[S_d(l) - S_i(l)]^2 \qquad \text{Eq. (5.3)}$$

Where N is the length of the signal.

5.1 Spectrogram Analysis

5.1.1 Spectrogram of Clean Audio

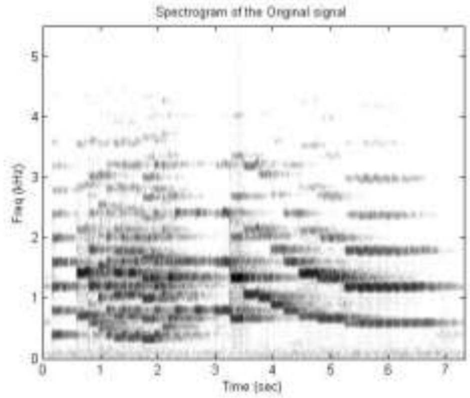

Figure 5.1: Spectrogram of Clean Audio (Mozart.wav)

5.1.2 Spectrogram of Noisy Audio (at 5dB)

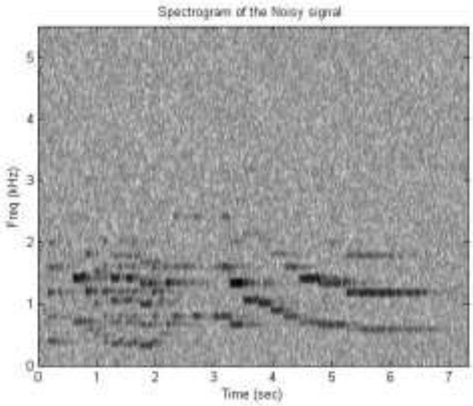

Figure 5.2: Spectrogram of Noisy Audio (Mozart.wav at -5dB AWGN with σ=0.047)

5.1.3 Spectrogram of Denoised Audio

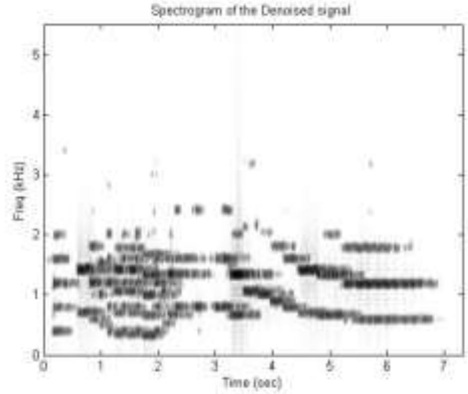

Figure 5.3: Spectrogram of Denoised Audio (Mozart.wav)

5.3 Result Summary

Performance comparison of Block Thresholding [1], [5], Block Thresholding (BT) with soft thresholding wavelet [4], Block Thresholding (BT) with hard thresholding wavelet [4], Minimum Mean Square Error Log Spectral Amplitude Estimation algorithm (MMSE-LSA) [5], Minimum Mean Square Error Log Spectral Amplitude Estimation algorithm by using Decision Direct method (MMSE-LSA-DD) [5] of Mozart signal for different SNR values is shown in the below table 5.1.The additive white Gaussian noise variance is taken at different σ, for obtaining different dB noisy signal.

Signal & Noise SNR	Noise Variance (σ)	MMSE-LSA-DD [5]	MMSE-LSA [5]	BT with Hard Thresholding Wavelet [4]	BT with Soft Thresholding Wavelet [4]	Block Thresholding [1],[5]	Proposed STFT based Adaptive Block Thresholding
		SNR (dB)	SNR (dB)	SNR (dB)	SNR (dB)	SNR (dB)	SNR (dB)
Mozart 5dB	0.047	7.625	7.625	9.84	11.57	14.90	**15.46**
Mozart 10dB	0.026	-	12.625	12.76	15.02	18.31	**19.09**
Mozart 15dB	0.015	-	17.727	15.97	18.75	22.03	**23.05**
Mozart 20dB	0.008	-	22.825	19.48	22.64	25.14	**26.29**
Mozart 25dB	0.004	-	28.785	23.62	27.43	30.29	**31.25**

Table 5.1: Performance Comparison of Previous Work with Proposed Work

It is observed from Table 5.1 that the SNR values are dependent upon, type of thresholding and the level of decomposition. Soft thresholds are better than hard thresholds for denoising the sound signals for [4]. The selection of level of decomposition plays a significant role, and should be optimal for best denoising results. Our proposed work outperforms all existing techniques.

6. Conclusion

Adaptive block thresholding has been widely used in denoising the sounds of musical instruments and then the other denoising techniques. In this paper, STFT is used for denoising sound signal corrupted with additive white Gaussian noise. First, sound signal is divided into multiple blocks depending upon the optimal block size for each signal. Denoising of signal is performed with these optimal block sizes in time-frequency domain by thresholding the time-frequency coefficients. When each block is denoised, all the blocks are concatenated to form the final denoised signal. It is also observed that when modified threshold is used, the SNR values are increased. Higher thresholds remove the noise well but some parts of the original signal are also removed because it is not possible to remove the noise without affecting the original signal. But using STFT having higher time-frequency resolution the denoised signal is almost same as original signal. This proposed algorithm gives highest SNR compared to other algorithm after denoising.

REFERENCES

[1] Ilker Bayram, "Employing phase information for audio denoising", IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), 2014.

[2] Gautam Bhattacharya, Philippe Depalle, "Sparse denoising of audio by greedy time-frequency shrinkage", IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), 2014.

[3] Richard E. Turner, Maneesh Sahani, "Time-Frequency Analysis as Probabilistic Inference", IEEE transaction on signal processing, Volume-62, Number-23, 2014.

[4] S. S. Joshi and Dr. S. M. Mukane, "Comparative Analysis of Thresholding Techniques using Discrete Wavelet Transform", International Journal of Electronics Communication and Computer Engineering, Volume 5, Issue (4) July, 2014.

[5] K.P. Obulesu and P. Uday Kumar, "Implementation of Time Frequency Block Thresholding Algorithm in Audio Noise Reduction", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 2, Issue 7, July 2013.

[6] Jagadale, B. N., "Audio signal processing using wavelet transform," Journal of Computer and Mathematical Sciences, Vol. 3, No. 6, pp. 557- 663, 2012.

[7] Jain, S. N., and Rai, C., "Blind source separation and ICA techniques: a review," International Journal of Engineering Science and Technology, Vol. 4, No. 4, pp. 1490-1503, 2012.

[8] Nehe, N. S., and Holambe, R. S., "DWT and LPC based feature extraction methods for isolated word recognition," EURASIP Journal of Audio, Speech and Music Processing, Springer, pp. 1-7, 2012.

[9] Cohen, R., "Signal denoising using wavelets," Project Report, Israel Institute of Technology, 2012.

[10] Alam, M., Islam, M. I., and Amin, M. R., "Performance comparison of STFT, WT, LMS and RLS adaptive algorithms in denoising of speech signals," International Journal of Engineering and Technology, Vol. 3, No. 3, pp. 235-238, 2011.

[11] Chen, D., Maclachlan, S., and Kilmer, M., "Iterative parameter choice and multigrid methods for anisotropic diffusion denoising," SIAM Journal of Scientific Computing, Vol. 33, No. 5, pp. 2972-2994, 2011.

[12] Basumallick, N., and Narasimhan, S. V., "A discrete cosine adaptive harmonic wavelet packet and its application to signal compression," Journal of Signal and Information Processing, Vol. 1, pp. 63-67, 2010.

[13] Abid, K., Ouni, K., and Ellouze, N., "A new psychoacoustic model for MPEG1 layer 3 coder using a dynamic gammachirp wavelet," IEEE Proceedings of International Conference on Signal Processing and Information Technology, pp. 123-128, 2009.

[14] Castells, F., Laguna, P., Srnmo, L., and Bollmann, A., "Principle component analysis in ECG signal processing," EURASIP Journal of Applied Signal Processing, pp. 1-21, 2007.

[15] Bahoura, M., and Rouat, J., "Wavelet speech enhancement based on time-scale adaptation," Speech Communications, Vol. 48, No. 12, pp. 1620-1637, 2006.

[16] Bakirci, U., and Kucuk, U., "The compression of musical instrument signals by wavelet transform," IEEE Proceedings of International Conference on Signal Processing and Communications Applications, pp. 493-495, 2004.

[17] Chevalier, P., Albera, L., Comon, P., and Ferreol, A., "Comparative performance analysis of eight blind source separation methods on radio communication signals," IEEE International Joint Conference on Neural Networks, 2004.

[18] Ching, T., and Wang, H. C., "Enhancement of single channel speech based on masking property and wavelet transform," Speech Communications, Vol. 41, No. 2, pp. 409-427, 2003.

[19] Asano, F., Ikeda, S., Ogawa, M., Asoh, H., and Kitawaki, N., "Combined approach of array processing and independent component analysis for blind separation of acoustic signals," IEEE Transactions on Speech & Audio Processing, Vol. 11, No. 3, 2003.

[20] Alam, J. F., and Walker, J. S., "Time frequency analysis of musical instruments," Proceedings in Society of Industrial and Applied Mathematics, Vol. 44, No. 3, pp. 457-476, 2002.

[21] Antoniadis, A., Bigot, J., and Sapatinas, T., "Wavelet estimators in nonparametric regression: A comparative simulation study," Journal of Statistical Software, Vol. 6, No. 6, pp. 1-83, 2001.

[22] Dapena, A., Bugallo, M. F., and Castedo, L., "Separation of convolutive mixtures of temporally-white signals: A novel frequency-domain approach," Proceedings of International Conference on Independent Component Analysis Blind Source Separation, pp. 315-320, 2001.

[23] Batista, L. V., Melcher, E., and Carvalho, L. C., "Compression of ECG signals by optimized quantization of discrete cosine transform coefficients," Journal of Medical Engineering 2001.

