DATA DENOISING IN ANALOG AND DIGITAL DOMAINS

by

SayedMasoud Hashemi Amroabadi
MASc., Isfahan University of Technology, Isfahan, Iran, 2008
BASc., Isfahan University of Technology, Isfahan, Iran, 2005

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering
Toronto, Ontario, Canada, 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis.
I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.
Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.
Signature

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Data Denoising in Analog and Digital Domains

© SayedMasoud Hashemi Amroabadi, 2010

Masters of Applied Science (MASc)

Signal Processing and Communications

Ryerson University

In this thesis, we develop various methods for the purpose of data denoising. We propose a method for Mean Square Error (MSE) estimation in Soft Thresholding. The MSE estimator is based on Minimum Noiseless Data Length (MNDL). Our simulation results show that this MSE estimate is a valuable comparison measure for different soft thresholding methods. Two denoising methods are proposed for analog domain: Mean Square Error EstiMation (MSEEM) which minimizes the worst case MSE estimate, and Noise Invalidation Denoising (NIDe) method which is based on the newly proposed idea of noise signature. While MSEEM shown to be the optimum denoising method for non-sparse signals, NIDe approach outperforms the other well known denoising methods in presence of colored noise. In digital domain we address two interesting problems: 1) simultaneous denoising and quantization method, 2) denoising a digital signal in digital domain. For problem one, we propose a new method that generalizes the idea of dead zone estimation to a multi-level noise removal. An example of this method is shown for hyperspectral image denoising and compression. A digital domain denoising approach pioneers in answering the second problem with only one prior knowledge on the desired signal, that it is digital. The method provides the optimum reconstruction levels in the MSE sense.

One of the critical steps of denoising process is the noise variance estimation. As a part of this thesis, we propose a novel noise variance estimation method for BayesShrink that outperforms conventional MAD-based noise variance estimation. Although BayesShrink is one of the most efficient denoising methods, no analytical analysis is available for it. Here, we study Bayes estimators for General Gaussian Distributed (GGD) data and provide the theoretical justification for BayesShrink. This study enables us to generalize the BayesShrink threshold to Generalized BayesShrink which outperforms the BayesShrink itself.

Acknowledgments

I would like to thank all people who have helped and inspired me during my study.

Foremost, I would like to express my sincere gratitude to my advisor Dr. Soosan Beheshti for the continuous support of my study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my MASc. study.

Besides my supervisor, I would like to thank my family: my lovely parents Fatemeh and Mohammad, for supporting me spiritually throughout my life and their non-stop love.

Last but not least, I would like to thank my brothers Mahdi and Meisam, and my sister-in-law Sheida for their supports and love.

Contents

List of Figures

List of Tables

Notations

- $\bar{y}(i)$: Noiseless data
- $y(i)$: Noisy data
- $w(i)$: $i.i.d$ Gaussian Noise
- $\bar{\theta}(i)$: Noiseless wavelet coefficients
- $\theta(i)$: Noisy wavelet coefficients
- $v(i)$: Gaussian Noise wavelet coefficients
- $\hat{\theta}(i)$: Estimated wavelet coefficient
- $\hat{y}$: Estimated noiseless data from noisy data
- $\sigma_{\bar{y}}$: Standard deviation of noiseless data
- σ_w : Noise standard deviation
- σ_y : Noisy data standard deviation
- T : Threshold value
- m : Number of coefficients larger than threshold value T which are kept in thresholding
- $\psi_T(\theta)$: Hard Thresholding function with T as its threshold value
- $\eta_T(\theta)$: Soft Thresholding function with T as its threshold value
- $\bar{\theta}_T$: Wavelet coefficients corresponding to the kept part of the noisy wavelet coefficients in thresholding methods
- Δ_T : Wavelet coefficients corresponding to the removed part of the noisy wavelet coefficients in thresholding methods
- A_T : Wavelet basis part corresponding to the kept part of the coefficients in thresholding methods
- B_T : Wavelet basis part corresponding to the removed part of the coefficients in thresholding methods
- A' : Transposed version of matrix A
- $||.||_2^2$: Second norm of the matrix

- $z_{T,H}$: Reconstruction error in Hard Threshold, $z_{T,H} = \frac{1}{N}||\bar{y} - \hat{y}||_2^2$
- $z_{T,S}$: Reconstruction error in Soft Threshold, $z_{T,S} = \frac{1}{N}||\bar{y} - \hat{y}||_2^2$
- $x_{T,H}$: Data error caused by Hard thresholding, $x_{T,H} = \frac{1}{N}||y - \hat{y}||_2^2$
- $\phi(x)$: CDF of Gaussian distribution
- $DL(\theta|\bar{\theta})$: description length of noiseless data
- Q_i : Quantization level in ith level
- y^Q : Noisy quantized data
- $R_{..}$: Autocorrelation
- $.^N$: the superscript N represent the length of a vector, i.e., $y^N = [y(1), \cdots, y(N)]'$
- $< a, b >$: Inner product of real vectors a and b which can be calculated by $< a, b >= a'b$.
- MSE : Mean Square Error (2.6)
- NMSE : Normalized MSE (2.7)
- MAD : Median Absolute Deviation, noise variance estimation method
- MNDL : Minimum Noiseless Description Length, is a MSE estimation method
- SPIHT : Set Partitioning in Hierarchical Trees, is a bit alignment and compression methods
- PSNR : Peak SNR, here is used for SNR of the outputs of the algorithms when we input is image (2.8)
- ISNR : Input SNR, SNR of input data

Chapter 1

Introduction

A fundamental problem in statistical signal processing is estimating signal from its noisy version. In this thesis we focus on this problem and propose denoising algorithms for both analog and digital domains. In classical signal processing, it is typical to assume the signal is low-pass and the noise is not. Thus, the noise can be reduced with a low-pass linear filter. Since linear time invariant (LTI) filtering is equivalent to multiplication in Frequency domain (Fourier Transform), all LTI techniques can be called "Fourier-Based" denoising. Low-pass filtering is extensively studied in many classical signal processing textbook, [1, 2, 3]. However, many signals have useful high-pass features like edges in the images which are removed or blurred by using low-pass filters.

Wiener filtering is another classical method used for denoising the signal corrupted by noise. Wiener filtering is based on the knowledge of the spectral properties of the original signal and the noise, to design an LTI filter that its output is close to the original signal in MSE sense [4, 5]. However, the spectral properties of the signal and/or noise are not always available.

Wavelet-based techniques offer an alternative to noise removal methods. Wavelet denoising can provide low-pass filtering to reduce noise, while the useful high-pass features of the signal can be preserved. Wavelet denoising is performed by taking the wavelet transform of the noisy signal and mapping the details with coefficients below a certain threshold to zero. An inverse wavelet transform is applied to the threshold signal to achieve the denoised signal. In wavelet transform, the edges and the useful high-pass features have significant coefficients and will not be removed in the thresholding process.

The pioneer research on wavelet shrinkage is done by Donoho and Johnstone [6, 7, 8] in which they introduced VisuShrink and SureShrink. VisuShrink uses Soft thresholding and universal threshold. Universal threshold is only a function of data length and noise variance. In most cases the noise variance is not known. In these situations, the noise variance is usually estimated by Median Abolute absolute Deviation (MAD) method in which median value of the high frequency subbands are used to estimate the noise variance. The other method proposed by Donoho and Johnstone, SureShrink, minimizes Steien's unbiased Risk Estimator (SURE). These two methods are working well for sparse signals such as the signals which are piecewise polynomial. One of the well-known works in wavelet thresholding is done by Chang *et al.* [9, 10] in which BayesShrink method is proposed. BayesShrink is originally proposed for natural images in which the subband wavelet coefficients can be modeled with General Gaussian Distribution (GGD) and tries to minimize the Bayes Risk. GGD is accepted as a standard distribution for subband wavelet coefficients of natural images [11, 12, 13, 14]. Although BayesShrink has many advantages, it does not have an analytical proof.

Another important problem in statistical signal processing is Mean Square Error (MSE) estimation. In wavelet denoising and shrinkage, MSE estimation is only addressed by Beheshti and Dahlhe in Minimum Noiseless Data Length (MNDL) [15]. MNDL is a best bases selection which provides upper and lower bounds for Noiseless Data Length (NDL) in Hard thresholding methods. For a given noise variance, minimizing NDL is equivalent to minimizing MSE. Thus, the upper bound, provided by MNDL, can be used to minimize the worst case MSE. Therefore, MNDL can also be used as a wavelet shrinkage method.

This thesis is organized as follows:

Chapter 2 contains problem formulation and background information including a brief review on wavelet transform and existing wavelet shrinkage methods. In Chapter 3, we extend the MSE estimation method proposed in MNDL [15] for soft threshold denoising. Using the provided upper bound and minimizing the worst case MSE, we propose a Soft Threshold wavelet denoising method called MSE EstiMation (MSEEM).

Noise Invalidation Denoising (NIDe) is introduced in Chapter 4. NIDe uses a noise signature to validate the signal and removes the coefficients which have the noisy behavior. Noise signature was proposed by Nikvand and Beheshti in [16]. In [16] Noise Invalidation in Soft Thresholding (NIST) is proposed that uses the absolute sorted value of noise wavelet coefficients as the noise signature. NIDe has two major differences with NIST: 1) an analytical signature with limited mean and variance is proposed, and 2) masking process is used instead of thresholding. Masking process removes all the coefficients with noise behavior which can be in different positions. While thresholding process removes the coefficients lower than a certain threshold value.

In Chapter 5, we propose two denoising methods that deal with digital data. One of the proposed methods accepts the noisy signals as its input and simultaneously denoises and quantizes it. This method takes care of the noisy input signal in two stages. In the first stage, it uses a multistage denoising method based on MNDL to detect the noise dominant part of the signal and at the same time it denoises and quantizes this part. Therefore, the remaining part of the input is noiseless dominant and one of the existing quantization methods can be used to quantize this part. The important advantage of this method is to prevent the resources to process the input noise which wastes the time and resources. Using this denoising/quantization algorithm, a compression method is proposed for Hyperspectral images and it is shown that the proposed method outperforms separate denoising and quantization approach. The other proposed method in Chapter 5 denoises the noisy quantized data. This method finds the reconstruction levels of the noisy quantized data. This method uses only the noisy data and does not need any side information, such as the number of levels or the value of quantization levels, to reconstruct the signal. When the input SNR is high, the reconstruction levels found by the proposed method are the same as the quantization levels. As a result, in low SNR cases, the reconstruction levels can be different in numbers and values from the quantization levels of the original data.

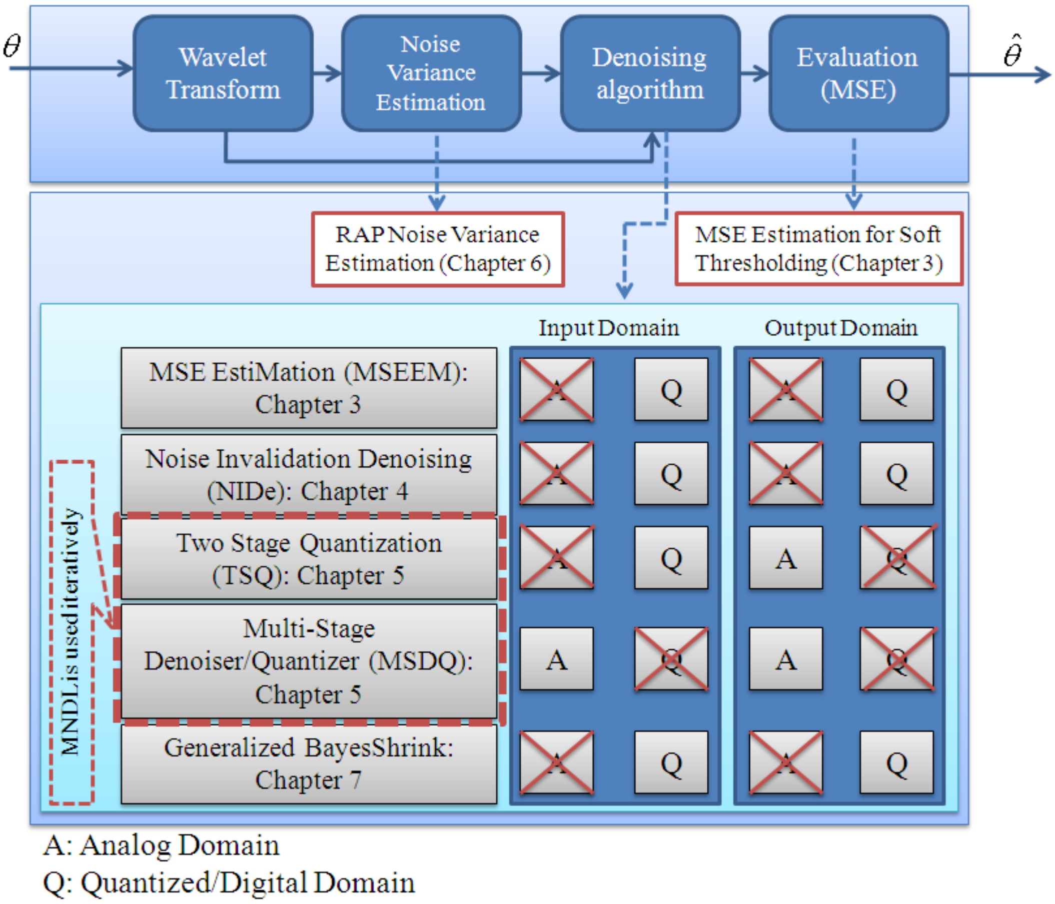
In Chapter 6 we address the noise variance estimation in BayesShrink. In this chapter we show that MAD-based noise variance estimation over estimates the noise variance in the

Table 1.1: Contribution of each chapter.

Chapter 3	A novle method is proposed in this chapter to estimate the MSE in soft thresholding methods. The proposed method only uses the noisy signal to estimate MSE.
Chapter 4	A universal denoising method is proposed which uses a noise signature to denoise the signal and provides a mathematical proof for limited mean and small variance of noise signature.
Chapter 5	In this chapter TSQ and MSDQ methods are proposed. TSQ generalizes the dead zone area concept in uniform quantizers and simultaneously denoises and quantizes the noisy signals. MSDQ pioneers the quantized/digital data denoising methods.
Chapter 6	A noise variance estimation is proposed for BayesShrink denoising method which outperfomrs MAD noise variance estimation in BayesShrink and improves the performance of BayesShrink.
Chapter 7	A novle denoising method is proposed based on Bayesian estimation of noisy GGD signals. In addition a mathematical justification is provided for BayesShrink.

cases with large wavelet coefficients in high frequency subbands. Then we use the statistical properties of noise, like Gaussian Noise autocorrelation, in combination with BayesShrink to estimate the noise variance. In Chapter 7, we study BayesShrink approach from a new angle. By drawing the Bayesian estimation of the noiseless wavelet coefficients form the noisy coefficients corrupted with Gaussian noise, we show that BayesShrink is a linear estimation of this Bayesian estimation function. This study provides tha theory behind the Bayesian approaches for Generalized Gaussian Distributed (GGD) signals and provides the role of BayesShrink in the Bayesian estimataion approaches. We propose a Bayesian estimate in form of a threshold that can improve the BayesShrink performance.

Figure 1.1 depicts the relationship among the chapters and the contribution in each chapetr is mentioned in Table 1.1.


θ
Wavelet Transform
Noise Variance Estimation
Denoising algorithm
Evaluation (MSE)
$\hat{\theta}$
RAP Noise Variance Estimation (Chapter 6)
MSE Estimation for Soft Thresholding (Chapter 3)
MNDL is used iteratively
A: Analog Domain
Q: Quantized/Digital Domain

	Input Domain		Output Domain	
MSE EstiMation (MSEEM): Chapter 3	~~A~~	Q	~~A~~	Q
Noise Invalidation Denoising (NIDe): Chapter 4	~~A~~	Q	~~A~~	Q
Two Stage Quantization (TSQ): Chapter 5	~~A~~	Q	A	~~Q~~
Multi-Stage Denoiser/Quantizer (MSDQ): Chapter 5	A	~~Q~~	A	~~Q~~
Generalized BayesShrink: Chapter 7	~~A~~	Q	~~A~~	Q

Figure 1.1: The relationship among the chapters of the thesis.

Chapter 2

Problem Formulation and Background

2.1 Problem Formulation

A large portion of digital signal processing is devoted to signal denoising. The basic idea behind this thesis is the estimation of noiseless signal from a noisy signal.

Suppose that for a noiseless signal, $\bar{y}$, we observe y, which is samples of the signal that have been contaminated with noise. Assuming an additive noise model, the noisy signal is the noiseless signal plus noise:

$$y(i) = \bar{y}(i) + w(i) \qquad i = 1, ..., N \tag{2.1}$$

where $y^N = \{y(i), i = 1, ..., N\}$ is the observed data vector, $\bar{y}^N = \{\bar{y}(i), i = 1, ..., N\}$ is the noiseless data and the noise, $w^N = \{w(i), i = 1, ..., N\}$, is independent and identically distributed (i.i.d) from a Gaussian distribution with zero mean and unknown standard deviation σ_w. In the notations the superscript N represents the length of the vector.

Our goal is to find an estimation of $\bar{y}$ which is shown by $\hat{y}$. The first step in denoising is to use a transform which represents the signal in a better form. Any orthogonal transform like Fourier transform, Short Term Fourier Transform (STFT) [17] and Wavelet Transform (WT) [18, 19, 20] can be used.

If $\bar{\theta}(i)$ denotes the noiseless transform coefficients, $\theta(i)$ denotes the noisy coefficients, and $v(i)$ denotes the noise coefficients. These coefficients can be expressed in terms of a desired orthonormal basis such that:

$$\theta(i) =< s_i, y^N >, \quad \bar{\theta}(i) =< s_i, \bar{y}^N >, \quad v(i) =< s_i, w^N > \tag{2.2}$$

where s_i is an element of orthonormal basis $S = [s_1, ..., s_N]$. Therefore the following equations hold.

$$\bar{y}^N = \sum_{i=1}^{N} \bar{\theta}(i) s_i, \quad y^N = \sum_{i=1}^{N} \theta(i) s_i \tag{2.3}$$

and the following equation describes the relationship between them.

$$\bar{\theta}(i) = \theta(i) + v(i) \tag{2.4}$$

Orthonormal transforms do not change the variances.

$$\sigma_w = \sigma_v, \quad \sigma_{\bar{y}} = \sigma_{\bar{\theta}}, \quad \sigma_y = \sigma_\theta \tag{2.5}$$

Because of the lack of statistical information about the noiseless signal ($\bar{y}$) the estimated parameters ($\hat{y}$) are chosen to optimize a criterion based on the observed signal (y). A common measure of the quality of an estimator is Mean-Square Error (MSE). MSE shows the Euclidean distance between the estimation and the noiseless signal in average. For example if we apply a threshold value, T, on a signal and map the $N - m$ smallest samples, which are smaller than T, to zero and keep the m larger ones, MSE is defined as mean of z_T. Where $z_T = ||\bar{y} - \hat{y}||^2$ is the reconstruction error.

$$MSE = \mathrm{E}(z_T) = \mathrm{E}(||\bar{y} - \hat{y}||_2^2) \tag{2.6}$$

Beside MSE, we use Normalized MSE (NMSE) and PSNR to evaluate the results. NMSE is defined as follows:

$$NMSE = \frac{\sum_i (\bar{\theta}_i - \hat{\theta}_i)^2}{\sum_j \bar{\theta}_j^2} \tag{2.7}$$

which normalizes MSE with the signal power. PSNR for an image with B bits per pixel resolution in dB can be calculated using,

$$PSNR = 10 \log_{10} \frac{(2^B - 1)^2}{MSE} \tag{2.8}$$

A popular technique for denoising is wavelet shrinkage or thresholding. A wavelet shrinkage method is achieved by taking the wavelet transform of the observed data vector and applying a function that shrinks the wavelet coefficients toward zero. Thresholding methods set a fraction of coefficients exactly to zero. The denoised signal is recovered by inverting the processed coefficients. Wavelet shrinkage and thresholding methods are effective because wavelet bases give a sparse representation for most signals, meaning that only a small fraction of large wavelet coefficients are needed to adequately represent the signal while the reminder can be set to zero with minimal effect. When noise is added to a signal, the small wavelet coefficients are buried in the noise. Applying a threshold maps a fraction of the coefficients to zero, ideally isolating these small coefficients. Hence the wavelet representation of some signals is more sparse than other ones, the fraction of the small non-essential coefficients varies in different signals. Therefore, the ideal fraction of removed coefficients differs from signal to signal, even if the level of noise in the signals is the same.

Here we focus on thresholding methods which keep m coefficients and map $N-m$ reminders to zero which can be the $N-m$ smallest ones or any other coefficients with noise like behavior.

Since in this thesis Orthogonal Wavelet Transform and specifically Discrete Wavelet Transform (DWT) is used to represent the signal, Wavelet Transform is briefly introduced in next section.

2.2 Brief Overview on Wavelet Transform

Although our proposed methods in the thesis can use any orthogonal transform, we use Discrete Wavelet Transform (DWT) because of its high performance and advantages shown by Donoho and Johnstone [6, 8]. In this section, wavelet transform is beefily reviewed.

2.2.1 The Continuous Wavelet Transform

Continuous Wavelet Transform (CWT) can be computed using the following equation,

$$\gamma(s,\tau) = \int f(t)\psi^{*}_{s,\tau}(t)dt \tag{2.9}$$

where $*$ denotes complex conjugation, s is scale factor and τ is translation factor. This equation shows how a function $f(t)$ is decomposed into a set of basis functions $\psi^{*}_{s,\tau}(t)$ called the wavelets. The wavelets are generated from a single basic wavelet $\psi(t)$, called mother wavelet, by scaling and translation:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s}) \tag{2.10}$$

The inverse wavelet transform is given by:

$$f(t) = \int\int \gamma(s,\tau)\psi_{s,\tau}(t)d\tau ds \tag{2.11}$$

It is important to note that in (2.9), (2.10) and (2.11) the wavelet basis functions are not specified. This is a difference between the wavelet transform and the Fourier transform, or other transforms. The theory of wavelet transforms deals with the general properties of the wavelets and wavelet transforms only. The most important properties of wavelets are the admissibility and the regularity conditions and these are the properties which gave wavelets their name. It can be shown that square integrable functions $\psi(t)$ satisfying the *admissibility condition*,

$$\int \frac{|\Psi(\omega)|^2}{|\omega|}d\omega < +\infty \tag{2.12}$$

in which $\Psi(\omega)$ stands for the Fourier transform of $\psi(t)$. The admissibility condition implies that the Fourier transform of $\psi(t)$ vanishes at the zero frequency, $|\Psi(\omega)|^2|_{\omega=0} = 0$. This means that wavelets must have a band-pass like spectrum.

A zero at the zero frequency also means that the average value of the wavelet in the time domain must be zero,

$$\int \psi(t)dt = 0 \tag{2.13}$$

Therefore, it must be oscillatory. In other words, $\psi(t)$ must be a wave.

2.2.2 The Discrete Wavelet Transform

The wavelet transform as described so far has the following properties that make it difficult to use directly in the form of (2.9). The first is the redundancy of the CWT. In (2.9) the wavelet transform is calculated by continuously shifting a continuously scalable function over a signal and calculating the correlation between the two. These scaled functions are not orthogonal basis and the obtained wavelet coefficients are highly redundant.

Even without the redundancy of the CWT we still have an infinite number of wavelets in the wavelet transform and it is better to reduce it to a manageable number. This is the second problem we have. The third problem is that for most functions the wavelet transforms have no analytical solutions and they can be calculated only numerically. Discrete Wavelet Transform solves these problems.

Wavelet can be seen as a band-pass filter, then a series of dilated wavelets can be seen as a band-pass filter bank like Figure 2.1. If we look at the ratio between the center frequency of a wavelet spectrum and the width of this spectrum we will see that it is the same for all wavelets.

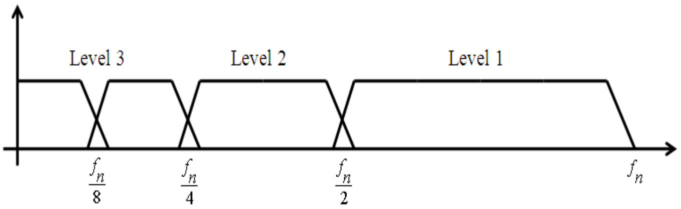


Figure 2.1: Wavelet spectra resulting from scaling of the mother wavelet in the time domain.

Discrete Wavelet Coefficients can be obtained through a filter bank, like the filter bank shown in Figure 2.2. In this filter bank, one of the filters is low pass filter corresponds to scaling function, $h[n]$, and has smoothing effect and the other one is high pass filter corresponds to the wavelet function, $g[n]$, and enhances the details [21]. The filter bank makes a tree structure in the coefficients. Each node of the tree represents a wavelet coefficient and each level of the tree represents a scale of detail of the wavelet coefficient while the root is the coarsest scale. A property of the wavelet coefficients is correlation in within and across scale [22]. This means that large coefficients are more likely to be adjacent to large coefficient both at the same level as in the details.

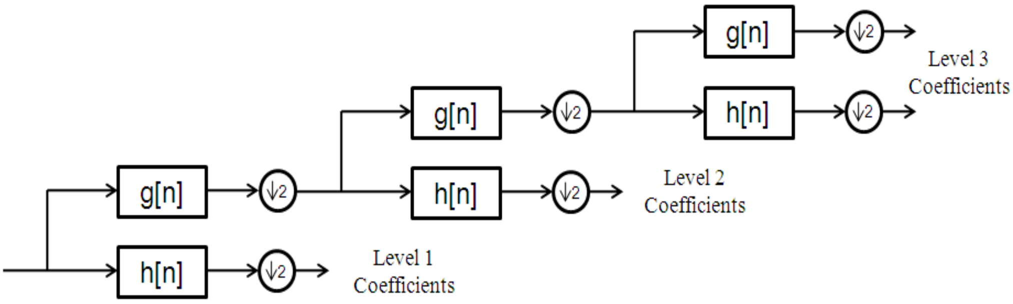


Figure 2.2: A three level wavelet filterbank.

2.2.3 Orthogonality of Wavelet transform

Orthogonal transforms are an important class of linear transforms. In general, a linear transform is orthogonal if it has a matrix representation W with $W^{-1} = W'$, where W' denotes the transpose of T.

Two properties of orthogonal transforms are used repeatedly in this work:

a) Orthogonal transform are Euclidean-distance preserving. If T is orthogonal, $||W(x-y)|| = ||x-y||$ for any two vectors x and y. This property is useful for denoising, since it means that the distance between two vectors is the same in transform domain as the original space. Consequently, a quantity such as an error between a true signal and an estimate can be evaluated in both of the domains.

b) Orthogonal transformations of a white noise are white. A zero-mean random vector $\eta \in \mathbb{R}^{\mathbb{N}}$ is a white noise if its autocorrelation matrix $E[\eta\eta'] = \sigma_\eta^2 I_N$ where σ_η is the variance of the noise. If W is orthogonal, $\nu = W\eta$ is also a white noise because $E[\nu\nu'] = E[W\eta\eta' W'] = WE[\eta\eta']W' = \sigma_\eta^2 I_N$.

This property is important for removing additive white noise because it allows us to consider the noise in the transform domain with the same statistics and properties.

A discrete wavelet transform can be made orthogonal by imposing certain restrictions on the filters. Specifically, it can be shown that a wavelet transform is orthogonal if and only if the filter transfer functions satisfy

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 1$$

and

$$H_1(z) = -z^{2^k+1}H_0(-z-1)$$

for some integer k. In the remainder of this work, we will restrict our attention to orthogonal wavelet transforms.

2.2.4 Denoising Signals with Wavelet Thresholding

When the signal ,$\bar{y}(i)$, is corrupted with an additive gaussian noise, $w(i)$, we would like to minimize the effect of noise in the noisy observed signal, $y(i)$.

The basic steps to denoise a signal with a Wavelet-based method are as follows:

1. Take samples $(y(1), y(2), ..., y(N))$ from the observed signal.

2. Compute the discrete wavelet transform of the observed noisy samples, $\theta(i) = \bar{\theta}(i) + v(i)$.

3. Apply a threshold or shrinkage procedure on the wavelet coefficients to find the best estimation of noiseless wavelet coefficients, $\hat{\theta}(i)$. In this thesis we focus on hard and soft threshold methods.

4. Take the inverse wavelet transform of the processed coefficients to get the denoised signal.

2.3 Hard Thresholding

Hard thresholding with threshold T is defined as follows.

$$\psi_T(\theta) = x.\mathbf{1}\{|\theta| > T\} \tag{2.14}$$

which can be written as follows for each coefficient.

$$\hat{\theta}_T(i) = \begin{cases} 0 & \text{if } \theta(i) < T, \\ \theta(i) & \text{if otherwise.} \end{cases} \tag{2.15}$$

Hard thresholding is a "keep or kill" procedure which keeps m and kills $N-m$ coefficients, where N is the number of wavelet coefficients. The transfer function of the Hard thresholding is shown in Figure 2.3.

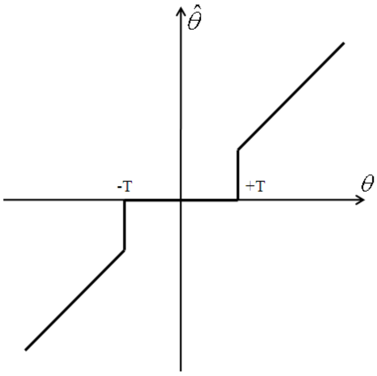


Figure 2.3: Hard Thresholding transfer function.

The only thresholding method which uses hard thresholding is Universal thresholding method [23]. Although Minimum Noiseless Description Length (MNDL) denoising [15] also uses hard thresholding, we introduce it in MSE estimation section, since it is basically an MSE estimation algorithm.

2.3.1 Universal Threshold

Donoho and Johnstone in [6] proposed a "Universal threshold". This threshold value is applicable for sparse signals (Besov space) and in particular for the piecewise polynomial signals in which most of the energy of signal is concentrated in a few number of large wavelet coefficients.

Universal threshold is based on the theorem proved in [23]. This theorem says that if z be a sample from distribution $N(0, I_n)$, then

$$\pi_n \equiv Prob\{||z||_{l_n^\infty} \leq \sqrt{2\log N}\} \to 1 \quad \text{as } n \to \infty \tag{2.16}$$

where N is the number of the wavelet coefficients of the samples. This theorem implies that for the *i.i.d* Gaussian white noise, in the limit the noise is almost surely bounded by the threshold $\lambda = \sigma_\omega\sqrt{2\log n}$. An advantage of this threshold is that it is easy to compute. The only parameter in this threshold is the noise variance. In [6] median estimator in high frequency subband is proposed by Donoho and Johnstone to estimate the noise variance.

$$\hat{\sigma}_{MAD} = \frac{Median(|\theta|)}{0.6745}, \theta \in \ subband \ \ HH_1 \tag{2.17}$$

this method is called Median Absolute Deviation (MAD) estimate and has become the standard for noise variance estimation [18]. In [24] some examples, especially with large data sets, are given in which σ_ω is over estimated and the Universal threshold makes the signal over-smooth by removing too many coefficients. In the given examples there are significant high frequency coefficients in the signal. Thus, the estimated noise variance using MAD is very large.

2.4 Soft Thresholding

Although hard threshold removes the noise effectively, it makes artifacts as a result of the Gibbs oscillation near discontinuities. An improvement over the the wavelet hard thresholding is the wavelet soft thresholding scheme.

Soft thresholding with threshold T is defined as follows.

$$\eta_T(\theta) = \hat{\theta}_T = \operatorname{sign}(\theta)\max(0, |\theta| - T) \tag{2.18}$$

Soft thresholding, shrinks coefficients above the threshold value. While, hard thresholding may seem to be natural but the continuity of soft thresholding has some advantages. For example, some of the visual affects made by hard thresholding are decreased by using soft thresholding. Soft thresholding transfer function is shown in Figure 2.4. The most important wavelet thresholding method using soft threshold are, VisuShrink [6], SureShrink [8], and BayesShrink [10]. In addition here we introduce Noise Invalidation in Soft Thresholding (NIST) as well [16, 25].

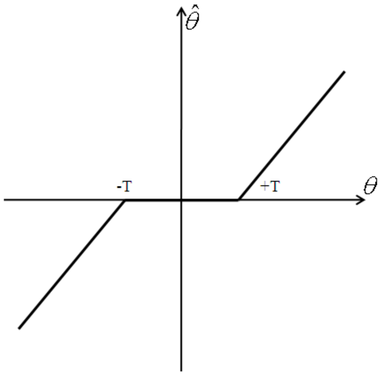


Figure 2.4: Soft Thresholding transfer function.

2.4.1 VisuShrink

VisuShrink [6] uses Universal threshold in soft thresholding to remove the noise. For signals with large number of elements such as images, VisuShrink yields an oversmoothed estimate of the noiseless signal, an example can be seen in figure 2.5. It is because the Universal Threshold (UT) is derived in the constraint that with high probability the estimate should be at least as smooth as the signal. Thus, the UT tends to be high for signals with high number of elements. Therefore, this method is not working well for the signals with discontinuity.

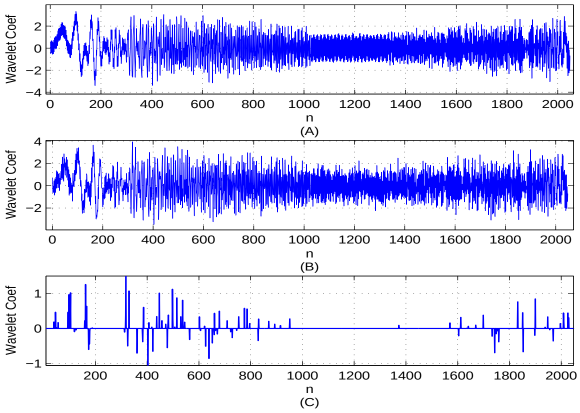


Figure 2.5: (A) Wavelet coefficients of the noiseless signal, (B) Wavelet transform of the noisy signal with SNR of 10 and length of 2048, (C) Thresholded coefficients by VisuShrink.

2.4.2 SureShrink

In [8] the idea behind the universal threshold and VisuShrink is extended to a level-dependent thresholding method called SureShrink. Let $\bar{y} = \{\bar{y}_i, i = 1, ..., d)\}$ be a vector of length of d, and $y = \{y_i, i = 1, ..., d\}$ be the observations with distribution of $N(\bar{y}, \sigma_\omega)$. SureShrink minimizes the Steien's unbiased Risk Estimator (SURE) which is a method for estimating the error $||\hat{y} - y||_2^2$ in an unbiased way, $\hat{y}$ is the estimation of $\bar{y}$ using soft thresholding. SURE for a threshold t and observed coefficients y is defined as follows

$$SURE(t; y) = d - 2 \times \#\{i : |y_i| < t\} + \sum_{i=1}^{d} \min(|y_i|, t)^2 \tag{2.19}$$

To find the threshold t the SURE will be minimized.

$$t^* = \arg\min_t SURE(t; y) \tag{2.20}$$

SureShrink has a good performance in extreme sparse cases. To make it more useful in the other cases as well, a hybrid method is proposed [8]. The idea behind the Hybrid method is that in each subband SURE is applied unless it is determined that the coefficients at the subband are negligible, in which case a Universal threshold is applied. Let η be the soft thresholding operator, t_{UT} be the Universal threshold and t_{SURE} be the SURE threshold value, then the Hybrid method can be formulated as follows,

$$\hat{\theta}_{hybrid}(i) = \begin{cases} \eta(t_{UT}; \theta(i)) & \text{if } s_d^2 \leq \gamma_d, \\ \eta(t_{SURE}; \theta(i)) & \text{if otherwise.} \end{cases} \tag{2.21}$$

where

$$s_d^2 = \frac{\sum_i (x_i^2 - 1)}{d} \qquad \gamma_d = \frac{\log_2^{\frac{3}{2}}(d)}{\sqrt{d}} \tag{2.22}$$

and d is number of coefficients in the subband.

2.4.3 BayesShrink

Chang *et al.* in [10] proposed a method to minimize the Bayes Risk (expected value of the mean squared error), which is given by

$$\tau(T) = E(\hat{y} - \bar{y})^2 = E_{\bar{y}} E_{y|\bar{y}} (\hat{y} - \bar{y})^2 \tag{2.23}$$

in which $\hat{y} = \eta_T(y)$, $y|\bar{y} \sim N(\bar{y}, \sigma_\omega)$ and $\bar{y}$ follows the General Gaussian Distribution (GGD), $\bar{Y} \sim GG_{\sigma_{\bar{y}},\beta}$. GGD can be shown as

$$GG_{\sigma_{\bar{y}},\beta} = C(\sigma_{\bar{y}}, \beta) \exp -[\alpha(\sigma_{\bar{y}}, \beta)|y|]^\beta, \tag{2.24}$$

$$\alpha(\sigma_{\bar{y}}, \beta) = \sigma_{\bar{y}}^{-1} [\frac{\Gamma(\frac{3}{\beta})}{\Gamma(\frac{1}{\beta})}]^{\frac{1}{2}}$$

and

$$C(\sigma_{\bar{y}}, \beta) = \frac{\beta.\alpha(\sigma_{\bar{Y}}, \beta)}{2\Gamma(\frac{1}{\beta})}$$

in which $\Gamma(x)$ is the Gamma function and is defined as $\Gamma(x) = \int_0^\infty e^{-u} u^{t-1} du$. Thus, to find the optimal threshold T^* the Bayesian Risk should be minimized. The threshold found in [10] is a function of $\sigma_{\bar{y}}$ and σ_w.

$$T^*(\sigma_{\bar{y}}, \sigma_w) = \arg\min_T \tau(T) \rightarrow T_{BS}(\sigma_{\bar{y}}) = \frac{\sigma_\omega^2}{\sigma_{\bar{y}}} \tag{2.25}$$

where σ_ω^2 is the noise variance and is estimated using MAD given in (2.17). $\sigma_{\bar{y}}^2$ is the variance of the noiseless signal which can be estimated from $\sigma_y^2 = \sigma_{\bar{y}}^2 + \hat{\sigma}_\omega^2$.

In BayesShrink, $\frac{T_B}{\sigma_\omega}$, the normalized threshold value, is inversely proportional to $\sigma_{\bar{y}}$ and proportional to σ_ω. When $\frac{\sigma_\omega}{\sigma_{\bar{y}}} \ll 1$, the signal is much stronger than the noise, $\frac{T_B}{\sigma_\omega}$ is chosen to be small to keep most of the coefficients; when $\frac{\sigma_\omega}{\sigma_{\bar{y}}} \gg 1$, the noise is dominant so the normalized threshold is chosen to be large to remove most of the coefficients.

2.4.4 Noise Invalidation in Soft Thresholding (NIST)

In most of the proposed denoising methods, a prior distribution is assumed for noiseless signal and based on it a shrinkage or thresholding algorithm is designed to estimate the noiseless signal. Nikvand and Beheshti in [16] proposed a denoising method based on soft thresholding which has a different point of view to the denoising problem. In this method instead of modeling the noiseless signal which is not available, a signature is provided for the noise and that signature is used to validate if the behavior of the coefficients is similar to noise or not.

The signature proposed in [16] is sorted absolute value of the Gaussian noise coefficients, Γ_m. If γ_m be a sample of Γ_m, it can be shown by:

$$\gamma_m = |\theta_{sort}[m]| \tag{2.26}$$

where θ_{sort} is the set of ascending sorted version of coefficients. Using this notation, in the denoising process $NIST$ tries to minimize $\alpha[m]$ which is:

$$\alpha[m] = \frac{\gamma_m - E(\Gamma_m)}{\lambda\sqrt{var(\Gamma_m)}} \tag{2.27}$$

Consequently, the number of kept coefficients is m^* and $N - m^*$ is the number of coefficients following the noise behavior, where N is the number of all coefficients. The number of kept coefficients can be found by:

$$m^* = N - \arg\min_m(\alpha[m] \geq 1) \tag{2.28}$$

removed coefficients are the smallest $N - m^*$ coefficients.

2.5 Bayesian Shrinkage Methods

There are other wavelet shrinkage methods which are using shrinkage functions to decrease the effect of noise in the coefficients. Most of these algorithms are based on Bayesian estimators. Since this group are not using any thresholding method, they are out of our focus in the thesis. To become familiar with them, we just briefly introduce some of these methods. In Bayesian wavelet shrinkage methods, a prior probability distribution is assumed for all the wavelet coefficients. The difference between the proposed methods is specially in this prior distribution. In [26], the assumed prior is a mixture of two Gaussian distributions,

$$d_{j,k}|\pi_{j,k} \sim (1-\pi_{j,k})N(0, c_j^2\tau_j^2) + \pi_{j,k}N(0, \tau_j^2) \tag{2.29}$$

where $j = 0, ..., J-1$, $k = 0, ..., 2^j - 1$ and J is the level of decomposition in wavelet transform. In [27], the following prior is used,

$$d_{j,k} \sim (1-\pi_{j,k})N(0, \tau_j^2) + \pi_{j,k}\delta(0) \tag{2.30}$$

in which $\pi_{j,k}$ is the probability that the coefficient is zero which has a Bernoulli distribution, $P(\pi_{j,k} = 0) = 1 - P(\pi_{j,k} = 1) = p_j$. In [27], the shrinkage function is calculated by finding the posterior median of the coefficients while in [26] the posterior mean is used. There are two main difficulties in these Bayesian methods: 1) difficulty in estimating the parameters and 2) inaccuracy caused by modelling each coefficient individually which fails to model the correlation between the wavelet coefficients.

In [28, 29], the first difficulty is approached by using simpler models for wavelet coefficients. To handle the second difficulty, Gaussian and non-Gaussian mixture densities are used in combination of Hidden Markov Models (HMM) and Hidden Markov Trees (HMT) to model the dependencies of the coefficients [22, 30, 31].

2.6 MSE Estimation in Hard Thresholding

MSE is one of the tools that can be used to evaluate the performance of the denoising methods. In almost all the parameter estimation problems we try to minimize the MSE.

However, as it is shown in (2.6), to calculate MSE, noiseless data must be available. Since we do not have access to the noiseless data, to minimize the MSE an MSE estimation and Minimization algorithm should be used.

In theory it is not difficult to minimize the MSE. For example it is well known that the posterior mean, $E(y|\bar{y})$, is an MSE minimizing method. However, in practical problems the complexity of the posterior mean is very high. Thus, different methods are proposed to find the minimum MSE (MMSE) with a computationally efficient algorithm such as Maximum a Posterior (MAP). Using the suboptimal solutions, we need to have a criterion to be able to compare the MSE of the method used in practice with the MMSE. Unfortunately, it is very hard to compute the the MMSE in many cases. To overcome this problem some methods are proposed to find bounds for MMSE in different estimation problems to be used behalf of the actual MMSE.

In [32] a lower bound is provided for error estimation of diffusion filters which is based on covariance inequality and the bound proposed in [33, 34] and Cramer-Rao bound [35, 36]. Other types of bounds are proposed in [37, 38, 39] which are based on the bounds proposed in [40] to estimate an unknown parameter α form the observed signal, $r(t) = s(t, \alpha)n(t)$ where $s(t)$ is a known signal used for communication, (*i.e. pulse-frequency modulation (PFM)*). However, all of the available methods are working in particular conditions and applications.

MSE estimation in wavelet denoising methods is addressed only in Minimum Noiseless Data Length (MNDL) [15]. MNDL is a best basis selection algorithm which provides upper and lower bounds for MSE in general. However, MNDL can be considered as an MSE estimator in especial case when noise variance, σ_w^2, is known and a kill and keep denoising process like hard threshold is used.

In MSE estimation of denoising methods, unknown parameter is the noiseless data, $\bar{y}$, which is corrupted by an additive white Gaussian noise w and the unknown parameter must be estimated with the known noisy data, y. The relationship between the known and unknown parameters are given in (2.1). The observed noisy data is projected to orthogonal Wavelet bases. The relationship between the associated coefficients is given in (2.4).

As it can be seen in figure 2.3, hard threshold method keeps the m largest coefficients and kills the remaining $N-m$ wavelet coefficients less than the threshold value. Thus, we can divide the wavelet coefficients into two parts: 1) coefficients killed in hard thresholding, and 2) kept coefficients [15]. These two groups have different effects on error (MSE). If we denote the noiseless coefficient corresponded to removed coefficients by

$$\Delta_T = \begin{bmatrix} \bar{\theta}(m+1) \\ ... \\ \bar{\theta}(N) \end{bmatrix} \tag{2.31}$$

with length of $N-m$ and coefficients corresponding to the kept ones by

$$\bar{\theta}_T = \begin{bmatrix} \bar{\theta}(1) \\ ... \\ \bar{\theta}(m) \end{bmatrix} \tag{2.32}$$

with length of m. The noiseless coefficients can be shown by:

$$\bar{\theta} = \begin{bmatrix} \bar{\theta}_T \\ \\ \Delta_T \end{bmatrix} \tag{2.33}$$

Let us denote W as the wavelet transform matrix, using this notation we have $\bar{\theta} = W\bar{y}$. Thus, $y^N = W^{-1}\bar{\theta} + w^N$ and since we are using orthonormal wavelet transform $W^{-1} = W'$ where W' is transposed of W. Similarly, W' can be divided into two parts:

$$W' = \begin{bmatrix} A_T & B_T \end{bmatrix} \tag{2.34}$$

In which A_T is the part of the matrix which associates the kept coefficients and B_T is corresponding to the removed coefficients. Using these notations we have:

$$y^N = \begin{bmatrix} A_T & B_T \end{bmatrix} \begin{bmatrix} \bar{\theta}_T \\ \\ \Delta_T \end{bmatrix} + w^N \tag{2.35}$$

in which w is white Gaussian noise. Consequently, the estimated coefficients vector in hard thresholding is as follows.

$$\hat{\theta}_{T,H} = \begin{bmatrix} A'_T y^N \\ \\ 0_{(N-m\times 1)} \end{bmatrix} = \begin{bmatrix} \bar{\theta}_T + A'_T w^N \\ \\ 0_{(N-m\times 1)} \end{bmatrix} \tag{2.36}$$

Using this equation, the reconstruction error would be:

$$z_{T,H} = \frac{1}{N}||\bar{\theta} - \hat{\theta}_{T,H}||_2^2 = \frac{1}{N}||A'_T w^N||_2^2 + \frac{1}{N}||\Delta_T||_2^2 \tag{2.37}$$

Note that $A'_T w$ represents the first m elements of the projected noise, $v = W \times w$. As a result, we have the following equation for the first term in (2.37),

$$\frac{1}{N}||A'_T w^N||_2^2 = \frac{1}{N}\sum_{i=1}^{m} v^2(i) \tag{2.38}$$

and since v has a Gaussian distribution, this is a sample of an mth order Chi-square distribution. Therefore, the reconstruction error in hard thresholding is a sample of a chi-square random variable with the following mean and variance:

$$\begin{aligned} \mathrm{MSE_H} = \mathrm{E}(Z_{T,H}) &= \frac{m}{N}\sigma_w^2 + \frac{1}{N}||\Delta_T||_2^2 & (2.39)\\ \mathrm{var}(Z_{T,H}) &= \frac{2m}{N^2}(\sigma_w^2)^2. & (2.40)\end{aligned}$$

In the next step a lower and an upper probabilistic bound for $||\Delta_T||_2^2$ are provided [15]. To find these bounds data error, $x_{T,H}$, is used:

$$x_{T,H} = \frac{1}{N}||y^N - \hat{y}_T||_2^2 = \frac{1}{N}||\theta - \hat{\theta}_T||_2^2 \tag{2.41}$$

Similar to $Z_{T,H}$, it is shown in [15] that $x_{T,H}$ follows Chi-Square distribution with the following mean and variance:

$$\begin{aligned} \mathrm{E}(X_{T,H}) &= (1 - \frac{m}{N})\sigma_w^2 + \frac{1}{N}||\Delta_T||_2^2 & (2.42)\\ \mathrm{var}(X_{T,H}) &= \frac{2}{N}(1 - \frac{m}{N})(\sigma_w^2)^2 + \frac{4\sigma_w^2}{N^2}||\Delta_T||_2^2 & (2.43)\end{aligned}$$

To find the bounds, the sorted absolute value of the noisy wavelet coefficients are used as threshold values, T. For each T, m represents the number of the kept nonzero coefficients. Thus, for each threshold value $x_{T,H}$ can be calculated and using the value of $x_{T,H}$, Δ_T is estimated.

Finally, $Z_{T,H}$ is bounded to the following upper and lower bounds:

$$\overline{Z_{T,H}(\phi(\beta), y, \phi(\alpha))} = \frac{m}{N}\sigma_w^2 + U_{T,H}(y,\alpha) + \beta\frac{\sqrt{2m}}{N}\sigma_w^2 \tag{2.44}$$

$$\underline{Z_{T,H}(\phi(\beta), y, \phi(\alpha))} = \frac{m}{N}\sigma_w^2 + L_{T,H}(y,\alpha) - \beta\frac{\sqrt{2m}}{N}\sigma_w^2 \tag{2.45}$$

where

$$U_{T,H}(y, \phi(\alpha)) = x_{T,H} - m_w + \frac{2\alpha^2\sigma_w^2}{N} + K_T(\alpha) \tag{2.46}$$

$$L_{T,H}(y, \phi(\alpha)) = x_{T,H} - m_w + \frac{2\alpha^2\sigma_w^2}{N} - K_T(\alpha)$$

and K_T is,

$$K_T(\alpha) = 2\alpha\frac{\sigma_w}{\sqrt{N}}\sqrt{\frac{\alpha^2\sigma_W^2}{N} + x_{T,H} - \frac{1}{2}m_w} \tag{2.47}$$

$$m_w = (1 - \frac{m}{N})\sigma_W^2 \tag{2.48}$$

in these equations $\phi(\alpha) = \int_{-\alpha}^{\alpha} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}dx$ and $\phi(\beta) = \int_{-\beta}^{\beta} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}dx$ provide confidence interval for x_T and z_T around their mean value using Central Limit Theorem (CLT).

$$Pr\{|Z_{T,H} - \mathrm{E}(Z_{T,H})| \leq \beta\sqrt{varZ_{T,H}}\} = \phi(\beta) \tag{2.49}$$

$$Pr\{|X_{T,H} - \mathrm{E}(X_{T,H})| \leq \alpha\sqrt{varX_{T,H}}\} = \phi(\alpha) \tag{2.50}$$

It is shown in [15] that the position of minimum in the Bounds is very close to the minimum of $Z_{T,H}$. Thus, absolute value of the coefficient at that position can be chosen as the threshold value in hard threshold to find the best estimation of noiseless data with minimum MSE.

2.7 Concluding Remarks

In this chapter we provided an overview of wavelet transform and one of the most current areas in statistical signal processing called Wavelet Shrinkage. Some pioneer wavelet denoising and shrinking methods were introduced. The common fact about the introduced methods is that, all of them are using the sparsity property of the wavelet transform which is applicable for signals in Besov space such as piecewise polynomial functions. Some of the introduced algorithms use extra assumptions, like BayesShrink, which assumes a heavy tailed GGD distribution for wavelet coefficients. Some of them are proposed for more general cases such as MNDL based and NIST denoising which are not using any special assumption about the signal.

The methods were presented in three groups: 1) Hard Thresholding, 2) Soft Thresholding, and 3) MSE estimation methods. Universal Threshold was the only methods in hard thresholding category. Soft Thresholding methods introduced here were: VisuShrink, SureShrink, BayesShrink and NIST. The third category, MSE estimation methods, only contained a special case of Minimum Noiseless Data Length (MNDL) method. MNDL can be used to estimate the MSE of denoising methods which are using hard threshold.

Additional denoising methods were introduced in section 2.5, which are using nonlinear shrinkage functions to estimate the denoise signal without providing any threshold value. Since in this thesis we focus on the methods that kill some coefficients and keep the remaining part, including masking and thresholding, these additional methods are out of our focus.

Chapter 3

MSE Estimation in Soft Thresholding and MSE EstiMation (MSEEM) Denoising

A criterion is needed to compare the results of a parameter estimator, for example denoising defined in (2.1) and (2.4). One of the best know criteria is MSE. The MSE (2.6) is dependent on the unknown parameters. In Section 2.6, we reviewed an MSE estimation algorithm that is useful in hard threshold denoising applications. The introduced method is a special case of MNDL in which noise variance is known and fixed. Here we extend the use of this MSE estimate method to be used in soft threshold denoising methods.

3.1 Reconstruction Error and Data Error in Soft Thresholding

MSE (2.6) is mean value of the reconstruction error. To estimate MSE in soft thresholding methods, we must be able to estimate z_T. Here, we introduce a method to estimate z_T in soft thresholding methods. Similar to the hard thresholding we divide the noiseless coefficients ($\bar{\theta}$) and wavelet matrix (W) into two parts which are shown in (2.33) and (2.34). Using these notations, the noisy data can be calculated from (2.35).

In Soft Threshold, given in (2.18), the coefficients smaller than the threshold value, T, are mapped to zero and the larger coefficients are shrinked by T. Consequently, the noiseless estimated coefficients are as follows.

$$\hat{\theta}_{T,S} = \begin{bmatrix} A'_T y^N - T\text{sgn}(A'_T y^N) \\ \\ 0_{(N-m\times 1)} \end{bmatrix} \tag{3.1}$$

$$= \begin{bmatrix} \bar{\theta}_T + A'_T w^N - T\text{sgn}(A'_T y^N) \\ \\ 0_{(N-m\times 1)} \end{bmatrix} \tag{3.2}$$

In which the Δ_T part, which contains the coefficients smaller than T, are removed. The remaining part is $\bar{\theta}_T$ corrupted with additive Gaussian noise coefficients, $A'_T w^N$, and decreased by the threshold value, T. Thus, the reconstruction error in this case is:

$$z_{T,S} = \frac{1}{N}||\bar{\theta} - \hat{\theta}_{T,S}||_2^2 = \frac{1}{N}||A'_T w^N - T\text{sgn}(A'_T y^N)||_2^2 + \frac{1}{N}||\Delta_T||_2^2 \tag{3.3}$$

Since $A'_T w^N$ contains the first m [1] elements of the projected noise, the reconstruction error in (3.3) is:

$$z_{T,S} = \frac{1}{N}\sum_{i=1}^{m}(v(i) + e(i))^2 + \frac{1}{N}||\Delta_T||_2^2 \tag{3.4}$$

in which $v(i)$s are noise wavelet coefficients and for $e(i)$ we have:

$$\sum_{i=1}^{m} e^2(i) = ||T\text{sgn}(A'_T y^N)||_2^2 = m \times T^2 \tag{3.5}$$

A sum of form $\sum_{i=1}^{m}(v(i) + e(i))^2$ is a noncentral chi-square of order m with mean and variance

$$\text{E}(\sum_{i=1}^{m}(v(i) + e(i))^2) = m\sigma_w^2 + \sum_{i=1}^{m} e^2(i) = m\sigma_w^2 + mT^2 \tag{3.6}$$

$$\text{var}(\sum_{i=1}^{m}(v(i) + e(i))^2) = 2m(\sigma_w^2)^2 + 4\sigma_w^2 mT^2 \tag{3.7}$$

Using (3.5) and (3.6) the mean and variance of $z_{T,S}$ are as follows:

[1] number of kept coefficients in soft thresholding

$$\begin{aligned} \text{MSE}_S = \text{E}(Z_{T,S}) &= \frac{m}{N}(\sigma_w^2 + mT^2) + \frac{1}{N}||\Delta_T||_2^2 \quad (3.8) \\ \text{var}(Z_{T,S}) &= \frac{2m}{N^2}(\sigma_w^2)^2 + \frac{4m}{N^2}(\sigma_w^2 T^2). \quad (3.9) \end{aligned}$$

As it can be seen in (3.8) and (3.9), the mean and variance of $z_{T,S}$ are only functions of the data length, the threshold value, the additive noise variance, and $\frac{1}{N}||\Delta_T||_2^2$. The threshold value and data length are available. The noise variance is needed in majority of the existing denoising approaches and methods such as median absolute deviation (MAD) are used for estimating this value. The main challenge here is to estimate $\frac{1}{N}||\Delta_T||_2^2$ by *only* using the observed noisy data. To provide an estimate we use the method which is proposed in [15]. The procedure is as follows: The absolute value of the noisy wavelet coefficients are used as the threshold values. At each iteration one of the threshold values is used in Soft thresholding method to estimate the noiseless wavelet coefficients. Then, using the same threshold value to find its data error, given in (2.41), we are able to find the bounds.

$x_{T,H}$ is the distance between the coefficients of the observed noisy data and coefficients of a hard thresholded version of the same data. Using (2.36), this error can be written as follows,

$$x_{T,H} = \frac{1}{N}|| \begin{bmatrix} 0_{(m\times 1)} \\ \Delta_T + B_T' w^N \end{bmatrix} ||_2^2, \quad (3.10)$$

Since $B_T' w^N$ contains the smallest $N-m$ [2] elements of the projected noise, the data error in (3.10) is:

$$x_{T,H} = \frac{1}{N}\sum_{i=m+1}^{N}(v(i))^2 + \frac{1}{N}||\Delta_T||_2^2 \quad (3.11)$$

A sum of form $\sum_{i=m+1}^{N}(v(i))^2$ is a chi-square of order $N-m$ with mean and variance

$$\text{E}(\sum_{i=m+1}^{N}(v(i))^2) = (N-m)\sigma_w^2 \quad (3.12)$$

$$\text{var}(\sum_{i=m+1}^{N}(v(i))^2) = 2(N-m)(\sigma_w^2)^2 \quad (3.13)$$

[2] number of removed coefficients in soft thresholding

Using (3.11) and (3.12) the mean and variance of $x_{T,H}$ are as follows:

$$\begin{aligned} E(X_{T,H}) &= (1-\frac{m}{N})\sigma_w^2 + \frac{1}{N}||\Delta_T||_2^2 && (3.14) \\ \mathrm{var}(X_{T,H}) &= \frac{2}{N}(1-\frac{m}{N})(\sigma_w^2)^2 + \frac{4\sigma_w^2}{N^2}||\Delta_T||_2^2. && (3.15) \end{aligned}$$

Consequently, from (3.14) we can estimate $\frac{1}{N}||\Delta_T||_2^2$ by:

$$\frac{1}{N}||\Delta_T||_2^2 \approx x_{T,H} - (1-\frac{m}{N}\sigma_w^2) \qquad (3.16)$$

or more accurately, we can find probabilistic bounds for $\frac{1}{N}||\Delta_T||_2^2$ using Central Limit Theorem (CLT) applied on $X_{T,H}$. From CLT and using (3.14) and (3.15) we have the following equations,

$$Pr\{|X_{T,H} - E(X_{T,H})| \leq \alpha\sqrt{\mathrm{var}(X_{T,H})}\} = \phi(\alpha) \qquad (3.17)$$

where $\phi(\alpha)$ is the Cumulative Density Function (CDF) of Gaussian distribution. Before starting, let us denote m_w as $(1-\frac{m}{N})$, $\bar{x}_{T,H}$ as $x_{T,H} - m_w\sigma_w^2$, m_δ as $\frac{1}{N}||\Delta_T||_2^2$, and v_w as $\frac{2}{N}(1-\frac{m}{N})\sigma_w^4$.

Using these notations and (3.17):

$$m_\delta - \alpha\sqrt{\frac{4\sigma_w^2}{N}m_\delta + v_w} \leq \bar{x}_{T,H} \leq m_\delta + \alpha\sqrt{\frac{4\sigma_w^2}{N}m_\delta + v_w} \qquad (3.18)$$

From this equation we can find the lower and the upper bound of $m_\delta = \frac{1}{N}||\Delta_T||_2^2$. To find the lower bound, $\bar{x}_{T,H} - m_\delta \leq \alpha\sqrt{\frac{4\sigma_w^2}{N}m_\delta + v_w}$ and to find the upper bound $m_\delta - \bar{x}_{T,H} \geq \alpha\sqrt{\frac{4\sigma_w^2}{N}m_\delta + v_w}$ should be solved. Thus, we can find the bounds by solving:

$$(\bar{x}_{T,H} - m_\delta)^2 = \alpha^2(\frac{4\sigma_w^2}{N}m_\delta + v_w) \qquad (3.19)$$

which gives the following upper, $U_{T,S}$, and lower, $L_{T,S}$, bounds:

$$U_{T,S}(\phi(\alpha)) = x_{T,H} - m_w + \frac{2\alpha^2\sigma_w^2}{N} + K_T(\alpha) \qquad (3.20)$$

$$L_{T,S}(\phi(\alpha)) = \min\{0, x_{T,H} - m_w + \frac{2\alpha^2\sigma_w^2}{N} - K_T(\alpha)\} \qquad (3.21)$$

and K_T is given in equation (2.47). The important thing here is how to select α. Selected α should satisfy the following condition:

$$\alpha \geq \frac{N}{\sqrt{2(N-m)}} \left(1 - \frac{m}{N} - \frac{x_{T,H}}{\sigma_w^2} \right) \tag{3.22}$$

to be able to solve the equation in (3.19).

3.2 Probabilistic Bounds for MSE in Soft Thresholding

The final step is to find the bounds of MSE which is the mean of Z_T. Using (3.8) and the bounds provided for $||\Delta_T||_2^2$ in (3.20) and (3.21) we have the following bounds for MSE in soft thresholding, MSE_S.

$$\frac{m}{N}(\sigma_w^2 + mT^2) + L_{T,S}(\phi(\alpha)) \leq MSE_S \leq \frac{m}{N}(\sigma_w^2 + mT^2) + U_{T,S}(\phi(\alpha)) \tag{3.23}$$

3.3 MSE EstiMation (MSEEM) Thresholding

The proposed method can be used as a denoising method as well. It provides an upper bound for MSE which can be considered as the worst case error. The threshold value that makes the upper bound minimum, can be used as an estimation for the best threshold value that minimizes the worst case MSE.

$$T_{MSEEM} = \arg_T \min\{\frac{m}{N}(\sigma_w^2 + mT^2) + U_{T,S}(\phi(\alpha))\} \tag{3.24}$$

Since we compare the noisy coefficients with the threshold value to keep or remove them, the threshold values can be selected from the noisy wavelet coefficients. Thus, we use the absolute value of the wavelet coefficients in soft threshold and find the estimated MSE for each threshold. The threshold value that minimizes the MSE will be chosen as the best threshold value.

3.4 Simulation Results

In this section we provide examples of MSE estimation for soft thresholding methods. In all these experiments we set the confidence probability of our estimations to be $p = 0.9$. The estimates are the probabilistic worse case estimates, that is the provided upper bound in (3.23). The test signals are Blocks, Bumps, QuadChirp and Mishmash introduced in [8] that cover a range of sparse and non-space signals. Length of the tested signals is $2^{11} = 2048$ and the test is done for 100 runs. In addition a swallow-like signal which is sum of sin and cosine waves passed through a Gaussian filter, introduced in [41] , is used to evaluate the results in a real application. Figures 3.1 and 3.2 show the test signals.

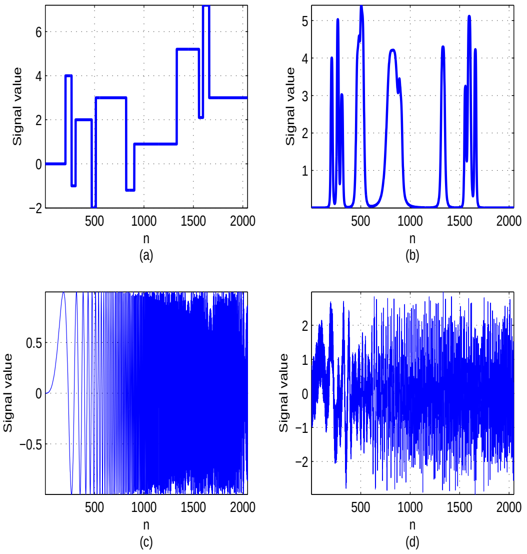


Figure 3.1: Signals used to test the proposed algorithm: (a) Blocks, (b) Bumps, (c) QuadChirp and (d) MishMash.

In figure 3.3 the bounds provided by MNDL for the test signals with SNR of 5 is shown. In this figure, $\phi(\alpha)$ and $\phi(\beta)$ are 0.9. Decreasing these values will give tighter bounds which are shown in figure 3.4 in which $\phi(\alpha)$ and $\phi(\beta)$ are 0.5.

In figure 3.5 the bounds provided for Swallow-like signal is shown when SNR is 5 and

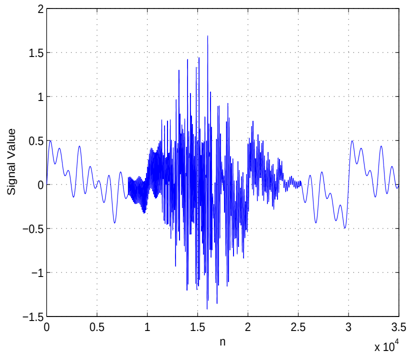
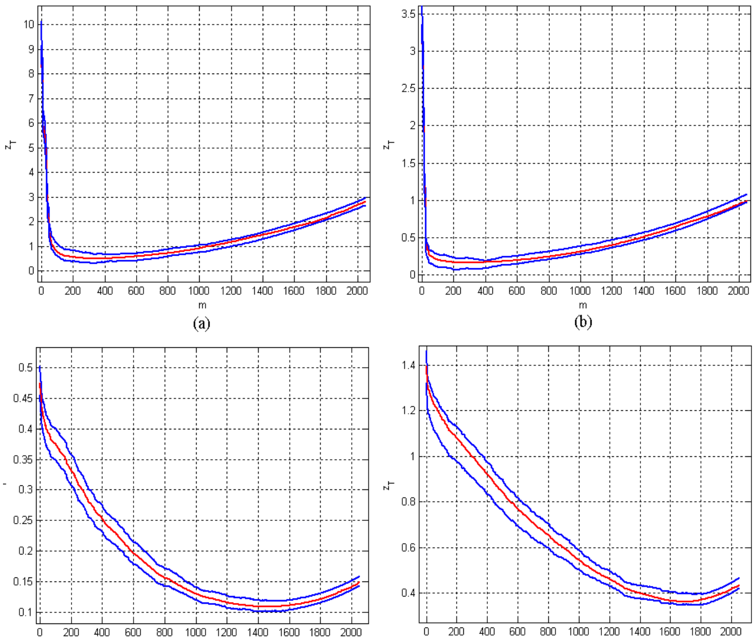


Figure 3.2: Swallow signal used to test the proposed algorithm.



(a)

(b)

(c)

(d)

Figure 3.3: $z_{T,S}$ and its upper and lower bounds provided by MNDL when Soft threshold is applied with different threshold values which are absolute value of the wavelet coefficients. The algorithm is applied on the test signals with SNR=5 and $\phi(\alpha) = \phi(\beta) = 0.9$. (a) Blocks, (b) Bumps, (c) QuadChirp, and (d) MishMash.

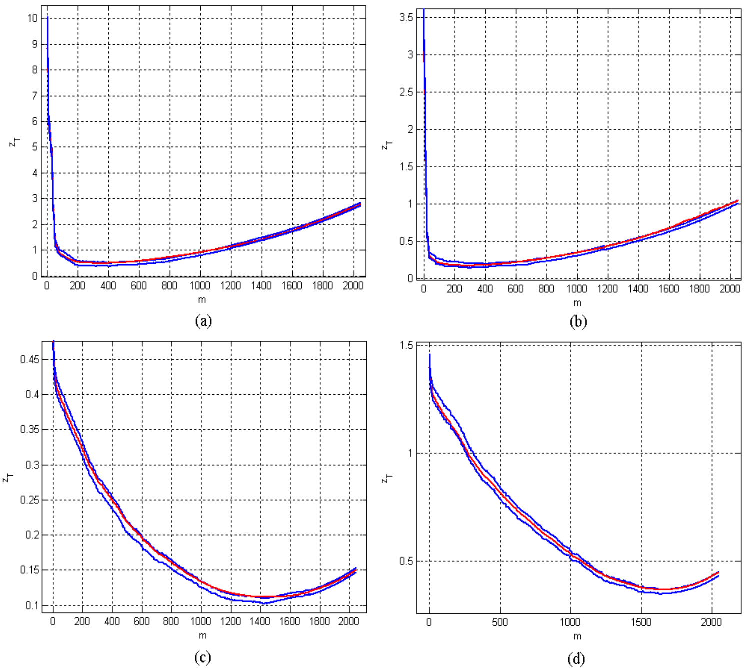


Figure 3.4: $z_{T,S}$ and its upper and lower bounds provided by MNDL when Soft threshold is applied with different threshold values which are absolute value of the wavelet coefficients. The algorithm is applied on the test signals with SNR=5 and $\phi(\alpha) = \phi(\beta) = 0.5$. (a) Blocks, (b) Bumps, (c) QuadChirp, and (d) MishMash.

$\phi(\alpha)$ and $\phi(\beta)$ are 0.9.

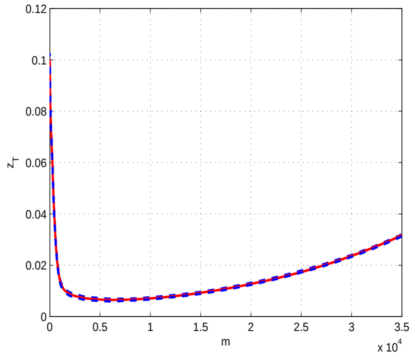


Figure 3.5: $z_{T,S}$ bounds for Swallow-like signal when SNR is 5.

To test the performance of the algorithm to estimate the MSE in different denoising methods which are using Soft threshold, tables 3.1 and 3.2 are given. In table 3.1, VisuShrink is applied on the test signals and MSE of VisuShrink in 100 runs is calculated. This value is compared with the MSE that is estimated with the proposed algorithm. The values in the table are from the upper bound found by proposed algorithm to have the worst case estimation. In table 3.2, SureShrink is used and its MSE is compared with the estimated MSE in that threshold value.

As it can be seen in Tables 3.1 and 3.2, the estimated value and the real MSE are very close which shows the strength of the proposed method in estimating MSE.

Table 3.3, compares the minimum of the upper bound found by the proposed method with the real minimum MSE which can be reached using Soft Threshold and the MSE estimated by the proposed method in [42]. In [42] a data-driven thresholding method based on MNDL is proposed. Similar to MNDL, MSE estimation is used in this method to find the threshold value that makes the MSE minimum. As it can be seen in Table 3.3 the estimated Minimum MSE is very close to the minimum possible MSE in soft threshold. Thus, MSEEM is very close to optimal denoising method. To test the robustness of the proposed method, Table 3.4 is given. In this table the mean and variance of the m values found by the proposed

Table 3.1: Comparison of the real MSE of VisuShrink and the MSE estimated by the proposed method. Top: Real MSE of VisuShrink, Bottom: Estimated MSE.

Signal	SNR		
	0	5	10
Blocks	1.7425 1.8282	1.1198 1.2595	0.7334 0.7602
Bumps	0.7828 0.9676	0.5517 0.6638	0.3955 0.4425
QuadChirp	0.4680 0.5471	0.4695 0.5222	0.4704 0.4886
MishMash	1.4196 1.5475	1.4226 1.5131	1.4253 1.4393
swallow-like	0.0731 0.0755	0.0581 0.0592	0.0426 0.0434

Table 3.2: Comparison of the real MSE of SureShrink with the MSE that is estimated with the proposed method. Top: Real MSE of SureShrink, Bottom: Estimated MSE.

Signal	SNR		
	0	5	10
Blocks	7.5544 7.6274	1.5737 1.5964	0.1116 0.2117
Bumps	2.0306 2.0584	0.1250 0.1903	0.0732 0.0987
QuadChirp	0.3714 0.4253	0.3654 0.3584	0.3550 0.3562
MishMash	0.8806 0.8594	0.6637 0.6958	0.8848 0.9384
swallow-like	0.0138 0.0233	0.0123 0.0131	0.0105 0.0116

Table 3.3: Comparison of the mimum of the upperbound of the proposed method, MSE of method proposed in [42], and the possible minimum MSE using soft threshold. Top: MSE of denoised signal with MSEEM, MSE of the proposed method in [42], and Bottom: Minimum possible MSE.

Signal	SNR		
	0	5	10
Blocks	1.3641	0.6812	0.1807
	1.587	0.7821	0.2289
	1.3383	0.6717	0.1806
Bumps	0.5706	0.2397	0.0895
	0.5829	0.2399	0.0893
	0.5509	0.2335	0.0892
QuadChirp	0.2551	0.1035	0.0412
	0.2612	0.1116	0.0893
	0.2553	0.1021	0.0411
MishMash	0.8177	0.3730	0.1474
	0.8490	0.4325	0.1828
	0.8128	0.3720	0.1473
swallow-like	0.0160	0.0065	0.0027
	0.0187	0.0077	0.0031
	0.0159	0.0065	0.0027

method are shown in different cases. m is the position at which the minimum of the upper bound happens. In addition to show the strength of the proposed method, the mean value of m is compared with the optimum number of coefficients that must be kept to make the MSE minimum. This number can be found by plotting real MSE for all the coefficients and choose the coefficient that makes the MSE minimum.

In Table 3.5 we compare the MSE resulted from MSEEM, BayesShrink and NIST [16, 25]. NIST keeps the low frequency wavelet coefficients unchanged and applies the soft threshold on other coefficients. To make MSEEM comparable with NIST, MSEEM is changed to work the same. Therefore, for table 3.5 we keep the low frequency coefficients and apply MSEEM on the remaining part. BayesShrink is completely subband dependent and finds a specific threshold for each subband.

As it can be seen in Table 3.5, MSEEM outperforms NIST in most cases. The only case in which NIST is working better than MSEEM is low SNR Blocks. However, since

Table 3.4: Mean and variance of the estimated m which is the position of the minimim of the upper bound and optimum m which makees the real MSE minimum. Top: mean of estimated m, Middle: variance of estimated m, and Buttom: real optimum m value.

Signal	SNR		
	0	5	10
Blocks	183.6	213.5	258.8
	23.2	36.9	39.4
	150	192	265
Bumps	168.9	245.6	304.1
	21.1	33.5	41.3
	147	201	299
QuadChirp	488.3	737.5	859.6
	65.1	45.1	26.4
	442	721	845
MishMash	523.8	830.0	963.6
	28.8	42.9	13.6
	508	831	947

Table 3.5: Comparison of MSE in MSEEM, NIST, and BayesShrink. Top: MSEEM, Middle: NIDe, and Bottom: BayesShrink.

Signal	SNR		
	0	5	10
Blocks	0.9295	0.3191	0.1421
	5.352	0.3529	0.1397
	0.596	0.2689	0.1394
Bumps	0.3582	0.1677	0.0716
	0.4280	0.2015	0.0784
	0.2641	0.1187	0.0526
QuadChirp	0.2661	0.1147	0.0420
	0.3933	0.1436	0.0486
	0.2542	0.1091	0.0410
MishMash	0.8346	0.3751	0.1271
	1.2174	0.4353	0.1391
	0.8300	0.3771	0.1337
swallow-like	0.0134	0.0071	0.0025
	0.0196	0.0150	0.0074
	0.0132	0.0035	0.0016

BayesShrink is subband dependent, it is working better than MSEEM in almost all the cases except MishMash.

Since MSEEM does not work very well when the number of coefficients is low, it is not very good in subband dependent form. BayesShrink is tuned for the signals with general Gaussian distribution. MishMash is not following GGD, so MSE of BayesShrink is higher than MSEEM in this case.

3.5 Concluding Remarks

In this chapter we proposed an MSE estimation based on MNDL for Soft Threshold denoising methods. An upper and a lower bound were proposed for MSE in soft thresholding denoising cases. The estimation was compared with the actual MSE of different methods and the simulations showed that the MSE estimates are close to the true values. Using the MSE bound and finding the minimum of the bound which was very close to the real minimum MSE, an MSE EstiMation (MSEEM) soft threshold denoising algorithm was proposed. MSE of MSEEM was compared with the minimum possible MSE that can be reached by soft threshold and it was shown that these MSEs are very close.

Chapter 4

Noise Invalidation Denoising (NIDe)

Most of the wavelet shrinkage methods are based on rejecting those wavelet coefficients that are smaller than a certain value and keeping the remaining coefficients. Thus, the problem of removing noise from a set of observed data is transformed into finding a proper threshold for the data coefficients. The pioneer shrinkage methods, such as VisuShrink and SureShrink, propose thresholds that are functions of the noise variance and the data length [6, 8]. Over the past fifteen years, several thresholding approaches such as [8, 10, 43] have been developed. These methods provide optimum thresholds by focusing on certain properties of the noise-free signal, and they are proposed for particular applications, mostly for the purpose of image denoising. Unlike these approaches, the method presented in this chapter focuses only on the properties of the additive noise. By relying on the noise statistics, the method defines a probabilistic region of confidence for the noise coefficients. Consequently, to find the best estimation of noiseless signal, $\hat{\theta}$, it validates the observed coefficients, given in (2.4), that are out of the noise confidence region and contain noiseless dominant parts.

A similar method is provided in [16]. In this paper, Noise Invalidation in Soft Thresholding (NIST) is proposed which uses a noise signature. The noise signature is the mean value of the sorted absolute value of N Gaussian zero mean noise samples. Here we introduce Noise Invalidation Denoising (NIDe) which uses an Empirical estimation of CDF of the sorted absolute value of the noise samples to define the signature [25]. We show that this signature has a finite small variance and mean which makes it an ideal signature to be used in invalidation process. NIDe is generalized version of NIST and provides the theory behind the signatures. In addition, our presented approach is a general-purpose method that does not need to exploit any particular property of the noise-free data itself.

For a general-purpose threshold, instead of concentrating on the properties of the remaining coefficients after denoising, it is logical to focus on the dismissed coefficients. These coefficients are discarded because they are attributed to the noise or very noisy coefficients. It is rational to equivalently state that these coefficients are discarded since they behave similarly to a set of coefficients that can be generated by an associated Gaussian distribution of the additive noise. In the following section we present one of the signatures of a set that is generated by this Gaussian distribution.

The provided denoising method has two major differences from the existing methods: 1) it focuses on the removed coefficients instead of the noiseless coefficients; and 2) it uses masking instead of hard or soft thresholding.

The Noise Invalidation Denoising method (NIDe) does not use thresholding methods. NIDe considers a probabilistic bound around the noise signature to validate the coefficients and removes the coefficients in the bound and keeps the coefficients out of the bound. Therefore, it is not using any threshold value but provides a mask to keep the coefficients out of the bound.

Wavelet denoising methods, rely heavily on the value of the additive noise variance. Our proposed method also requires the estimate of noise variance, and we use the MAD method for this estimation [6].

4.1 Additive Noise Signature

A general noise signature is defined as follows. Consider the additive noise random variable $V = [v_1, v_2, ..., v_N]'$ with zero mean and finite variance. Define the signature function for any value z as $g(z, V)$ such that the mean and variance of this function over V are finite values:

$$E(g(z,V)) = G_E(z) \tag{4.1}$$

$$var(g(z,V)) = G_{var}(z) \tag{4.2}$$

The signature for samples of a random process of length N, $V^N = [V_1, ..., V_N]'$, with iid members that have the same distribution of V is defined as:

$$g(z, v^N) = \frac{1}{N} \sum_{i=1}^{N} g(z, v_i) \tag{4.3}$$

It follows that the expected value and variance of the signature are:

$$E(g(z, v^N)) = \frac{1}{N} \sum_{i=1}^{N} E(g(z, v_i)) = G_E(z) \tag{4.4}$$

$$var(g(z, v^N)) = \frac{1}{N^2} \sum_{i=1}^{N} var(g(z, v_i)) = \frac{1}{N} G_{var}(z) \tag{4.5}$$

For a large data length, while the mean is a finite fixed value, the variance becomes smaller. The use of such signatures in invalidation of the additive noise is explored with an example in the following section.

4.1.1 Signature Example: Absolute Noise Sorting (ANS)

Consider a noise signature of following form

$$g(z, v_i) = \begin{cases} 1 & \text{if } |v_i| \leq z, \\ 0 & \text{if } |v_i| > z. \end{cases} \tag{4.6}$$

In this case, for the signature of the iid random process V^N in (4.3) we have

$$E(g(z, V_i)) = 1 \times Pr(|V_i| \leq z) + 0 \times Pr(|V_i| > z) = Pr(|V_i| \leq z) = F(z) \tag{4.7}$$

where $F(.)$ is the cdf of absolute value of the additive noise. For variance we have

$$E(g^2(z,V_i)) = 1 \times Pr(|V_i| \leq z) + 0 \times Pr(|V_i| > z) = F(z), \tag{4.8}$$

$$var(g(z,V_i)) = E(g^2(z,V_i)) - E^2(g(z,V_i)) = F(z) - F^2(z) = F(z)(1-F(z)) \tag{4.9}$$

If we show the CDF of standard Gaussian distribution by $\phi(.)$,

$$F(z) = 2\phi(\frac{z}{\sigma}) - 1 \tag{4.10}$$

For each z, the value of $g(z,v^N) = m/N$ where m is the number of samples of v^N with absolute values smaller than z. Equivalently, when sorting v^N, the mth value is the largest v_i that is smaller than z. Therefore, when sorting the v_is, the index is $Ng(z,v^N) = m$. Figure 4.1 illustrates the effect of sorting and the role of the small variance in providing a noise signature. The figure shows the behavior of 100 samples of Gaussian noise with unit variance and length 2048. As the top figure shows, with a very high probability, the values of this data are bounded between $\pm 3\sigma$.

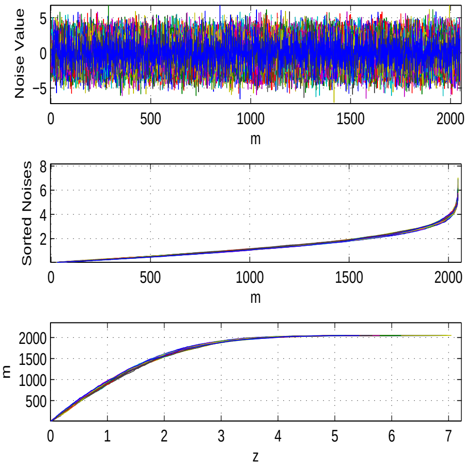


Figure 4.1: Top figure: 100 runs of a zero mean Gaussian distribution with unit variance and length 2048. Middle: The same 100 runs of the above figure sorted based on their absolute values. Bottom: This is the middle figure with its vertical and horizontal axes swapped ($m = Ng(z,v^N)$).

However, if we sort the same data based on its absolute value in the middle figure, the values collapse in a much denser area. Such behavior can be explained by the ANS signature as follows. The bottom figure shows the result of swapping the horizontal and vertical axes of the middle figure. Here the horizontal axis is z and the vertical shows 100 samples of $Ng(z, v^N)$ where $N = 2048$. As it is expected, these values are around mean $NF(z)$ with variance $F(z)(1 - F(z))$. This will allow us to define proper confidence regions, with a high probability p, around the noise signature. Due to the structure of these distributions, these regions are considerably smaller than the corresponding confidence regions of the Gaussian distribution of the additive noise itself.

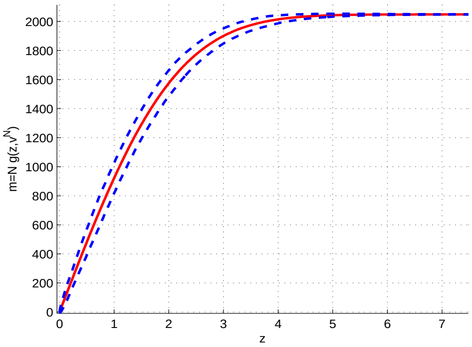


Figure 4.2: Solid line: Mean of the noise $g(z, V^N)$. Dashed lines are upper and lower bounds with confidence probability 0.999997.

Therefore, for each z and for a high confidence probability p, we can find $L_N(z)$ and $U_N(z)$ around the mean value of $F(z)$ such that

$$Pr\{L_N(z) \leq g(z, v_N) \leq U_N(z)\} = p \tag{4.11}$$

For example, Figure 4.2 shows the bounds on $g(z, V^N)$ for confidence probability $p = 0.999997$ and with $\sigma = 2.5$.

4.1.2 Confidence Region and Gaussian Estimate

While it is straightforward to make a table of values of the boundaries shown in Figure 4.2, it is possible to use Gaussian estimates for the distributions of $g(z, v^N)$ for large enough values of N as $g(z, v^N)$ in (4.3) is average of N independent variables. Using the Central Limit Theorem for this distribution, we have

$$Pr\{\frac{|g(z, v^N) - F(z)|}{\lambda\sqrt{\frac{1}{N}F(z)(1 - F(z))}} \leq 1\} \approx erf(\frac{\lambda}{\sqrt{2}}) \tag{4.12}$$

where $erf(x) = \frac{1}{\pi}\int_0^\infty e^{-t^2}dt$. This estimates the boundaries in (4.11) to be

$$L_N(z) = F(z) - \lambda\sqrt{\frac{1}{N}F(z)(1 - F(z))} \tag{4.13}$$

$$U_N(z) = F(z) + \lambda\sqrt{\frac{1}{N}F(z)(1 - F(z))} \tag{4.14}$$

The choice of λ should be such that the probability is close to one and at the same time the boundary is not very loose. In statistics the three-sigma rule, or empirical rule, states that for a normal distribution, almost all values lie within three standard deviations of the mean. For a better quality measure, the six sigma approach increases the standard deviation to 4.5 (equivalently $p = 0.999997$). Consequently, we suggest choosing λ such that $3 \leq \lambda \leq 5$. Interestingly, our experimental observation shows that the threshold associated with $\lambda = 4.5$ provides the optimum threshold with respect to MSE in 90% of cases.

4.2 Noise Invalidation with Absolute Coefficient Sorting (ACS)

The coefficients of our observed data is in form of $\theta_i = v_i + \bar{\theta}_i$ (2.4) which has the same structure as the noise except its mean which is the noiseless coefficient $\bar{\theta}_i$. If $H(z, \bar{\theta}_i)$ be defined as

$$H(z, \bar{\theta}_i) = \phi(\frac{z - \bar{\theta}_i}{\sigma_w}) + \phi(\frac{z + \bar{\theta}_i}{\sigma_w}) - 1 \tag{4.15}$$

From (4.7) and (4.9), mean and variance of θ_i will be:

$$E(g(z,\theta_i)) = Pr(|v_i + \bar{\theta}_i| \leq z) = Pr(-z - \bar{\theta}_i \leq v_i \leq z - \bar{\theta}_i) = H(z,\bar{\theta}_i) \tag{4.16}$$

$$var(g(z,\theta_i)) = H(z,\bar{\theta}_i)(1 - H(z,\bar{\theta}_i)) \tag{4.17}$$

Figures 4.3 and 4.4 show typical behaviors of $H(z,\bar{\theta}_i)$ for various $\bar{\theta}_i$s and change of noise variance. Note that $g(z,\theta_i)$ when the mean $\bar{\theta}_i$ is zero is the same as $g(z,v_i)$. Sorting the coefficients in this case is analogous to calculation of

$$g(z,\theta^N) = \frac{1}{N}\sum_{i=1}^{N} g(z,\theta_i) \tag{4.18}$$

which has the following mean and variance.

$$E(g(z,\theta^N)) = \frac{1}{N}\sum_{i=1}^{N} H(z,\bar{\theta}_i) \tag{4.19}$$

$$var(g(z,\theta^N)) = \frac{1}{N^2}\sum_{i=1}^{N} H(z,\bar{\theta}_i)(1 - H(z,\bar{\theta}_i)) \tag{4.20}$$

Since the value of $H(z,\bar{\theta}_i)$ in (4.15) is bounded between zero and one, the variance of this value is much less than its mean for large values of N. Therefore, a dense area will cover the sorted data with a high probability. This area becomes distinguished from the area covered from the sorted noise only signal as the value of z grows and as the nonzero coefficients become effective. This performance is illustrated in Figure 4.5 which shows the area covered by the sorted noisy data for Blocks signal when SNR is 5. The figure also shows the behavior of the sorted noise only data. As it can be seen with probability 0.99997 there is no overlap between the sorted noise and sorted noisy data after a certain value of z.

4.2.1 Noise Invalidation in Application

Using the noise sorting signature, it is possible to invalidate the noisy coefficients with a high confidence. Figure 4.6 shows the application of the method. The confidence region for noise

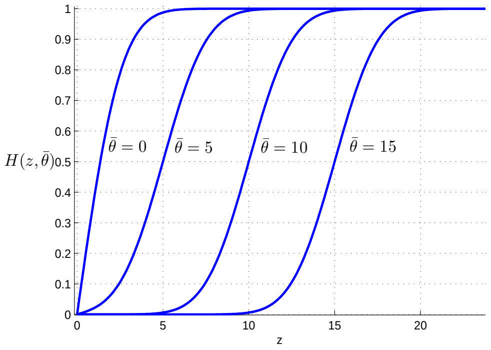


Figure 4.3: Expected value of $g(z, \theta)$ for various values of $\bar{\theta}$ when the additive noise variance is σ = 4.

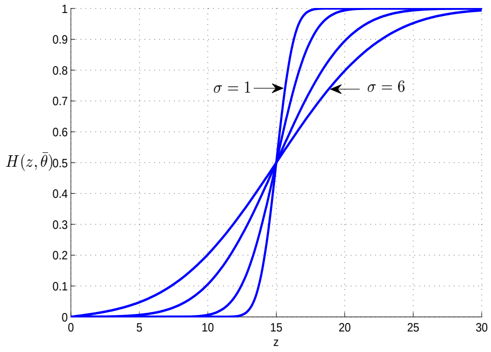


Figure 4.4: Expected value of $g(z, \theta)$ when $\bar{\theta}$=15 and the noise standard deviations σ = 1, 2, 4 and 6.

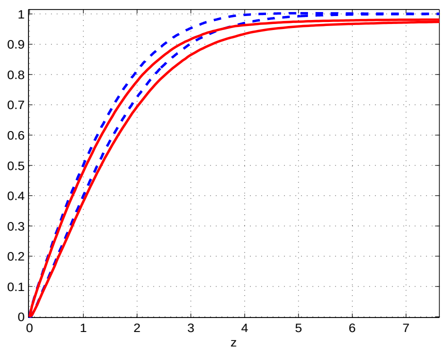


Figure 4.5: The area between the solid lines is the confidence region of sorted absolute values of the noisy data coefficients of Blocks signal (SNR=5) with probability 0.999997. The area between the dashed lines is the noise confidence region with probability 0.999997.

only data is available upon knowing or estimating the noise variance. As the sorted absolute noisy data leaves the noise confidence region, it assures that the coefficients are becoming more effective than the noisy part of the data. The best estimation of the noiseless part will be the coefficients out of the confidency bound.

$$\hat{\theta} = \{\theta : \forall z \in \theta, L_N(z) \nleq g(z,\theta) \nleq U_N(z)\} \tag{4.21}$$

which means to keep the coefficients fallen out of the bound and remove the coefficients inside the confidence bound.

4.2.2 Colored Noise in Absolute Coefficient Sorting (ACS)

The signature of the colored noise can be defined the same as the iid case. If we denote the autocorrelation between the zero mean Gaussian v_is by $R_{vv}(m)$,

$$g(z, v^N)) = \frac{1}{N}\sum_{i=1}^{N} g(z, v_i) = [\frac{1}{N}\ \frac{1}{N}\ \cdots\ \frac{1}{N}] \begin{bmatrix} g(z, v_1) \\ g(z, v_2) \\ \vdots \\ g(z, v_N) \end{bmatrix} \tag{4.22}$$

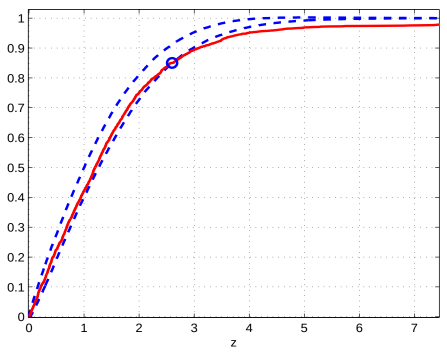


Figure 4.6: Solid line is the sorted absolute values of the observed data coefficients crossing upper bound of the noise confidence region at $z = 2.6$ when the observed data is noisy Blocks signal (SNR=5). The area between the dashed lines is the noise confidence region with probability 0.999997.

For the expected value of this function we have

$$E(g(z, V^N))) = \frac{1}{N}\sum_{i=1}^{N} E(g(z, V_i)) = F(z) \tag{4.23}$$

which is similar to that of the iid additive noise. However, for the variance, since the following holds

$$var(g(z, V^N)) = var(\frac{1}{N}\sum_{i=1}^{N} g(z, V_i)) =$$

$$[\frac{1}{N}\ \frac{1}{N}\ \cdots\ \frac{1}{N}] \begin{bmatrix} var(g(z,V_1)) & cov(g(z,V_1)g(z,V_2) & \cdots & cov(g(z,V_1)g(z,V_N)) \\ cov(g(z,V_1)g(z,V_2)) & var(g(z,V_2)) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ cov(g(z,V_1)g(z,V_N)) & \cdots & \cdots & var(g(z,V_N))) \end{bmatrix} \begin{bmatrix} \frac{1}{N} \\ \frac{1}{N} \\ \vdots \\ \frac{1}{N} \end{bmatrix}$$

we have

$$\begin{aligned} var(g(z, V^N))) = \frac{1}{N}F(z)(1 - F(z)) + \\ \frac{1}{N^2}\sum_{i=1,j=1,i\neq j}^{N} cov(g(z, V_i)g(z, V_j)) \end{aligned} \tag{4.24}$$

where elements of the second term are

$$\begin{aligned} &cov(g(z,V_i)g(z,V_j)) = \\ &E(g(z,V_i)(g(z,V_j)) - E(g(z,V_i))E(g(z,V_j)) \end{aligned} \tag{4.25}$$

in which the second term is simply $F^2(z)$ and the first term is $Pr(|V_i| \leq z \;\&\; |V_j| \leq z)$. For the first term, the joint distribution of V_i and V_j is a Gaussian distribution with zero mean and variance

$$E(\begin{bmatrix} V_i \\ V_j \end{bmatrix}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; var(\begin{bmatrix} V_i \\ V_j \end{bmatrix}) = \sigma^2 \begin{bmatrix} 1 & \rho_{ij} \\ \rho_{ij} & 1 \end{bmatrix} \tag{4.26}$$

where

$$\rho_{ij} = \frac{R_{vv}(i-j)}{R_{vv}(0)} \tag{4.27}$$

with $R_{vv}(0) = \sigma^2$. The decomposition of the covariance matrix is as follows

$$\begin{aligned} &\begin{bmatrix} 1 & \rho_{ij} \\ \rho_{ij} & 1 \end{bmatrix} = \\ &\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1+\rho_{ij} & 0 \\ 0 & 1-\rho_{ij} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{aligned} \tag{4.28}$$

Therefore, by the following transformation

$$\begin{bmatrix} x_i \\ x_j \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} v_i \\ v_j \end{bmatrix} \tag{4.29}$$

the two x_i and x_j random variables are independent and with variances $\sigma^2(1+\rho_{ij})$ and $\sigma^2(1-\rho_{ij})$. As a result, for the first term of the covariance

$$E(g(z,V_i)(g(z,V_j)) = Pr(|V_i| \leq z \;\&\; |V_j| \leq z) \tag{4.30}$$

$$\leq Pr(|X_i| \leq \sqrt{2}z \;\&\; |X_j| \leq \sqrt{2}z) \tag{4.31}$$

$$= F(\frac{\sqrt{2}z}{\sqrt{1+\rho_{ij}}})F(\frac{\sqrt{2}z}{\sqrt{1-\rho_{ij}}}) \tag{4.32}$$

Figure 4.7 show the area considered for the calculation of this probability.

With similar analogy, for the signal in presence of the noisy data, we have

$$E(g(z,\Theta^N))) = \frac{1}{N}\sum_{i=1}^{N} H(z,\bar{\theta}_i) \tag{4.33}$$

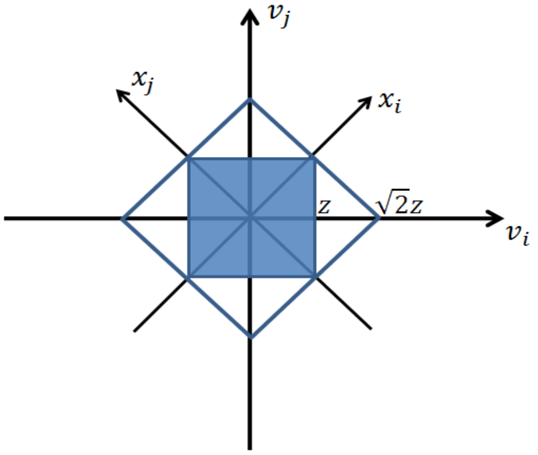


Figure 4.7: The desired area for calculation of the probabilities in (4.30) and (4.32).

where $H(z,\bar{\theta}_i)$ was defined in (4.15). For the variance of the sorted absolute value of the noisy data, similarly (4.9) holds. Therefore, structure of the variance is similar to (4.24):

$$\begin{aligned} var(g(z,\Theta^N))) = \frac{1}{N^2}\sum_{i=1}^{N}(H(z,\bar{\theta}_i) - H^2(z,\bar{\theta}_i)) + \\ \frac{1}{N^2}\sum_{i=1,j=1,i\neq j}^{N} cov(g(z,\theta_i)g(z,\theta_j)) \end{aligned} \tag{4.34}$$

For the covariance in the second term

$$\begin{aligned} cov(g(z,\Theta_i)g(z,\Theta_j)) = \\ E(g(z,\Theta_i)(g(z,\Theta_j)) - E(g(z,\Theta_i))E(g(z,\Theta_j)) \end{aligned} \tag{4.35}$$

we use (4.33) to calculate $E(g(z,\Theta_i))E((g(z,\Theta_j))$ and have

$$\begin{aligned} E(g(z,\Theta_i)(g(z,\Theta_j)) = Pr(|V_i+\bar{\theta}_i| \leq z \,\&\, |V_j+\bar{\theta}_j| \leq z) \\ \leq Pr(\sqrt{2}(-z-\frac{\bar{\theta}_i+\bar{\theta}_j}{2}) \leq X_i \leq \sqrt{2}(z-\frac{\bar{\theta}_i+\bar{\theta}_j}{2}) \qquad (4.36) \\ \& \sqrt{2}(-z-\frac{\bar{\theta}_i-\bar{\theta}_j}{2}) \leq X_j \leq \sqrt{2}(z-\frac{\bar{\theta}_i-\bar{\theta}_j}{2})) \qquad (4.37) \\ = H(\frac{\sqrt{2}z}{\sqrt{1+\rho_{ij}}},\frac{\bar{\theta}_i+\bar{\theta}_j}{\sqrt{2(1+\rho_{ij})}})H(\frac{\sqrt{2}z}{\sqrt{1-\rho_{ij}}},\frac{\bar{\theta}_i-\bar{\theta}_j}{\sqrt{2(1-\rho_{ij})}}) \qquad (4.38) \end{aligned}$$

Thus, if the additive noise is colored, the expected value of $g(z, V^N)$ and $g(z, \Theta^N)$ will remain the same as these expected values for the white noise in (4.10) and (4.20). For the variance of the sorted noise we have

$$var(g(z, V^N))) = \frac{1}{N} F(z)(1 - F(z)) + \frac{1}{N^2} \sum_{i=1,j=1,i \neq j}^{N} cov(g(z, V_i) g(z, V_j)) \tag{4.39}$$

$$\leq \frac{1}{N} F(z)(1 - F(z)) + \frac{1}{N^2} \sum_{i=1,j=1,i \neq j}^{N} [F(\frac{z}{\sqrt{1 + \rho_{ij}}}) F(\frac{z}{\sqrt{1 - \rho_{ij}}}) - F^2(z)] \tag{4.40}$$

As the variance indicates, the wider is the autocorrelation of the noise process with itself, the wider is the signature region of the noise and noisy data and therefore, as it is expected, it may become more difficult to distinguish the data from the noise.

4.3 Simulation Results

We perform our denoising methods on white and colored noisy versions of six standard signals, Blocks, Mishmash, Bumps, and QuadChirp which are the test signals introduced in [6]. Five level decomposition with Haar wavelet is chosen for this experiment. The confidence probability of the methods for noise invalidation region is 0.999997 and test is done for 100 runs. Figure 4.8 shows the six signals and their coefficients. As this figure confirms the test signals represent a wide range of possible coefficient structures. For example Figure 4.9 shows the coefficient distribution of some of these signals. while signals such as Blocks have very few nonzero coefficients and many coefficients close to zero, signals such as MishMash have more uniformly distributed coefficients. Blocks and MishMash signals represent two extreme structures, while QuadChirp have a combined structure of both of these signals. We compare the proposed method with VisuShrink, SureShrink, and MSEEM which are more general-purpose thresholding approaches. On the other hand, Sure-LET and BayesShrink are image denoising methods that are performing well with the one dimensional signals. We also consider these method comparison. We compare the performance of the methods based on their normalized mean square error (NMSE) given in (2.7).

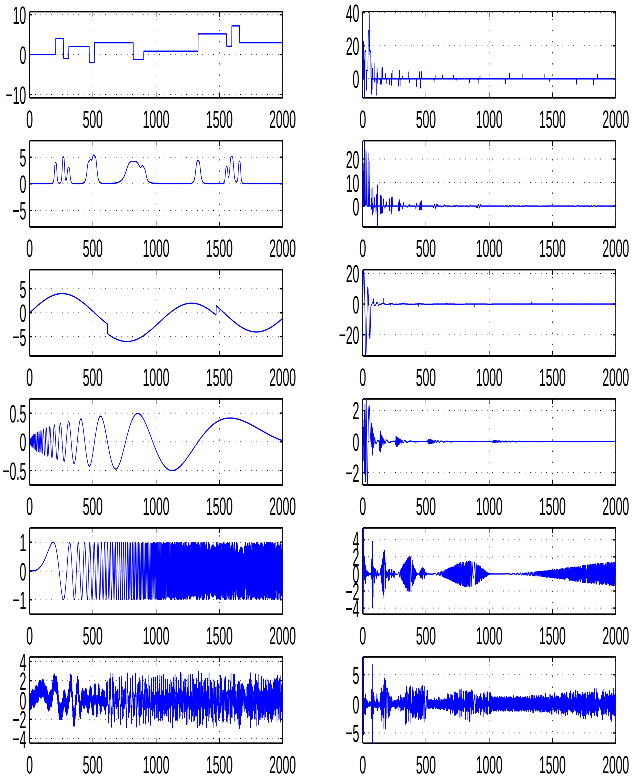


Figure 4.8: From top to bottom: Blocks, Bumps, HeavySin, Doppler, QuadChirp, and MishMash. Left figures are the signals and right figures are their corresponding wavelet coefficients.

Tables 4.1 and 4.2 provide the MSE of the compared methods. The methods are compared for a range of SNRs both in presence of white and colored noise. It is important to note that for this number of runs, the MSEs for all the methods have very small standard deviation that are much smaller than the MSE itself.

As Table 4.1 shows, three methods NIDe, BayesShrink and Sure-Let are comparable in presence of additive white noise, with NIDe outperforming the other two methods for the more sparse signals 90% of times for even a wider range of SNR including SNRs between

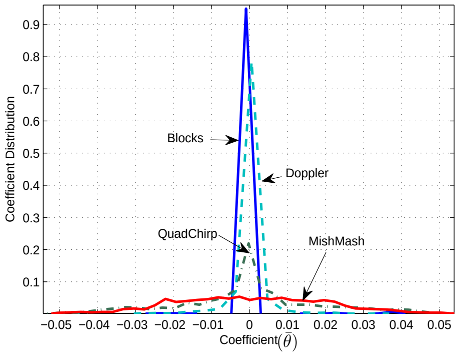


Figure 4.9: Coefficient distribution for Blocks, Doppler, Quadchirp and Mishmash.

1 and 14 that are shown in the table. For the non-sparse ones such as Quad-Chirp and Mishmash, however, Sure-let performs slightly better than the other two methods.

Figures 4.10 show the denoised versions of Blocks with these methods. As the figures show and Table 4.1 confirms, the denoised data with NIDe and BayesShrink in presence of a white noise are comparable while the other methods have larger MSE. Table 4.2 compares the performance of the methods when the signals are corrupted by a colored noise. Autocorrelation of this colored noise is depicted in Figure 4.11. This table shows the strength of NIDe in colore noise cases in comparison with the other mentioned methods.

In presence of colored noise, NIDe does not use any whitening filter to denoise the signal. In colored case NIDe uses the same method as it uses in white noise case and provides a noise signature for the colored noise and uses it to validate the signal. Design of the whitening filter, using it to whiten the signal and noise, and use of its inverse to project the signal to the original domain increases the error. Thus, beside the strength of NIDe in denoising, using no whitening filter is another reason that NIDe has better performance in colored noise case.

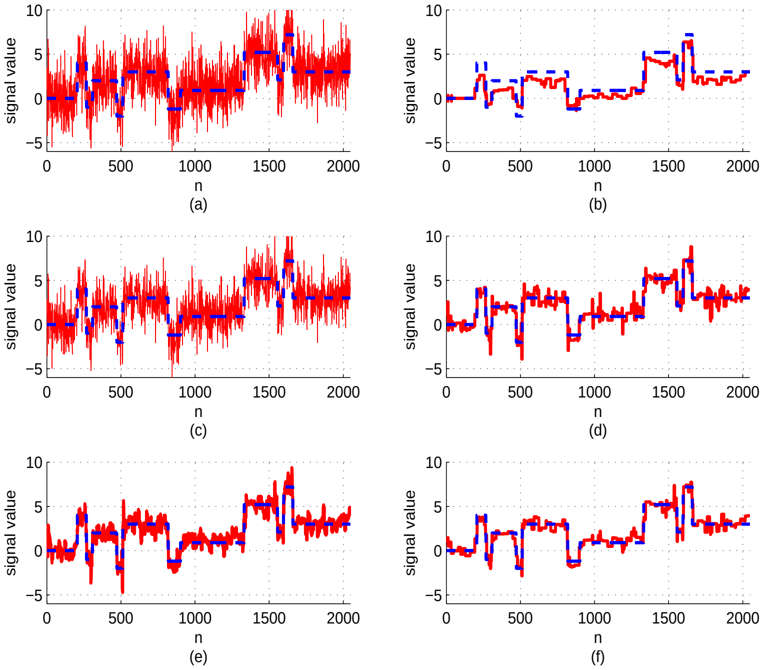


Figure 4.10: Dashed Blue line is the noiseless Blocks. (a) Noisy Blocks with SNR=4, (b) denoised by VisuShrink, (c) denoised by SureShrink, (d) denoised by BayesShrink, (e) denoised by Sure-Let, (f) denoised by NIDe.

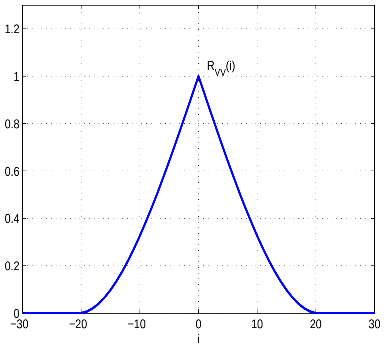


Figure 4.11: Autocorrelation of the colored additive noise.

4.4 Concluding Remarks

A denoising approach based on direct invalidation of the coefficients was proposed. The invalidation process used a signature of the additive noise in the form of a probabilistic confidence region. The signature was defined based on the statistical properties of the additive noise and was such that its standard deviation was much smaller than its mean. In this work we provided one example of such signature which illustrated itself in simply sorting the coefficients. This signature was tested by both colored and white noises and its strength in both the cases was shown. It was shown that NIDe has a robust performance in all the cases including colored and white noise with different SNR. When the additive noise was white, NIDe outperformed the other mentioned methods in sparse cases. However, when the noise was colored, NIDe outperfomred all the other mentioned methods.

Table 4.1: Normalized Reconstruction MSE for the Thresholding Methods. For the white additive noise. Averaged over 100 runs

	Visu	SURE	Bayes	SURE	MSEEM	NIDe
			Shrink	-LET		
Blocks						
SNR=1	0.168	0.657	0.111	0.124	0.091	**0.066**
SNR=4	0.124	0.259	0.058	0.065	0.059	**0.042**
SNR=8	0.086	0.032	0.026	0.027	0.028	**0.020**
SNR=10	0.072	0.015	0.015	0.017	0.018	**0.013**
SNR=14	0.048	0.007	0.007	0.007	0.011	**0.005**
Bumps						
SNR=1	0.155	0.427	0.098	0.122	0.132	**0.091**
SNR=4	0.127	0.065	0.073	0.063	0.084	**0.070**
SNR=8	0.096	0.026	**0.023**	0.027	0.044	0.025
SNR=10	0.083	0.023	0.019	0.018	0.030	**0.017**
SNR=14	0.062	0.017	0.012	**0.009**	0.014	**0.009**
HeavySin						
SNR=1	0.137	0.670	0.029	0.115	0.047	**0.028**
SNR=4	0.096	0.268	**0.017**	0.057	0.030	**0.017**
SNR=8	0.063	0.032	0.010	0.023	0.018	**0.009**
SNR=10	0.050	**0.007**	0.016	0.015	0.014	**0.007**
SNR=14	0.035	0.004	**0.003**	0.006	0.005	**0.003**
Doppler						
SNR=1	0.798	0.098	**0.069**	0.126	0.125	0.078
SNR=4	0.659	0.085	0.085	0.067	0.087	**0.076**
SNR=8	0.493	0.078	0.036	0.030	0.049	**0.032**
SNR=10	0.424	0.076	0.029	0.020	0.036	**0.024**
SNR=14	0.308	0.071	0.012	0.010	0.015	**0.009**
QuadChirp						
SNR=1	0.931	0.782	0.466	**0.447**	0.467	0.637
SNR=4	0.903	0.771	0.284	**0.276**	0.278	0.357
SNR=8	0.864	0.757	**0.129**	0.131	0.129	0.151
SNR=10	0.841	0.753	**0.086**	**0.086**	0.085	0.096
SNR=14	0.780	0.750	0.038	0.037	0.036	**0.035**
MishMash						
SNR=1	0.926	0.523	0.517	**0.462**	0.497	0.693
SNR=4	0.906	0.430	0.318	**0.286**	0.306	0.374
SNR=8	0.865	0.432	0.146	**0.136**	0.140	0.156
SNR=10	0.836	0.623	0.095	**0.089**	0.090	0.099
SNR=14	0.752	0.641	0.040	**0.039**	0.041	**0.039**

Table 4.2: Normalized Reconstruction MSE for the Thresholding Methods with the colored additive noise with autocorrelation in Figure 4.11. Averaged over 100 runs

	Visu	SURE	Bayes	SURE	MSEEM	NIDe
			Shrink	-LET		
Blocks						
SNR=1	0.406	0.695	0.562	6.212	0.754	**0.385**
SNR=4	0.212	0.334	0.286	3.107	0.572	**0.210**
SNR=8	0.098	0.124	0.121	1.214	0.365	**0.092**
SNR=10	0.063	0.074	0.079	0.753	0.151	**0.060**
SNR=14	0.030	0.035	0.037	0.295	0.042	**0.024**
Bumps						
SNR=1	0.455	0.648	0.559	6.443	0.827	**0.427**
SNR=4	0.279	0.305	0.293	3.220	0.620	**0.245**
SNR=8	0.105	0.115	0.125	1.273	0.298	**0.098**
SNR=10	**0.072**	0.075	0.081	0.773	0.131	**0.072**
SNR=14	0.035	0.038	0.035	0.299	0.052	**0.030**
HeavySin						
SNR=1	0.402	0.712	0.531	6.527	0.711	**0.334**
SNR=4	0.209	0.328	0.270	3.256	0.423	**0.173**
SNR=8	0.085	0.117	0.105	1.280	0.184	**0.071**
SNR=10	0.057	0.069	0.068	0.803	0.096	**0.047**
SNR=14	0.027	0.023	0.028	0.317	0.054	**0.020**
Doppler						
SNR=1	0.446	**0.399**	0.569	6.431	0.850	0.401
SNR=4	0.330	0.236	0.293	3.201	0.571	**0.235**
SNR=8	0.231	0.138	0.127	1.207	0.317	**0.110**
SNR=10	0.187	0.113	0.082	0.76	0.275	**0.075**
SNR=14	0.133	0.089	0.037	0.038	0.167	**0.030**
QuadChirp						
SNR=1	0.964	1.183	1.139	1.298	1.102	**0.594**
SNR=4	0.978	0.965	0.489	0.536	0.962	**0.326**
SNR=8	0.986	0.833	0.164	0.186	0.471	**0.137**
SNR=10	0.990	0.801	0.100	0.117	0.235	**0.088**
SNR=14	0.980	0.771	0.039	0.051	0.082	**0.037**
MishMash						
SNR=1	0.974	1.186	1.205	1.246	0.821	**0.607**
SNR=4	0.956	0.889	0.523	0.523	0.598	**0.336**
SNR=8	0.961	0.727	0.174	0.180	0.240	**0.142**
SNR=10	0.963	0.699	0.105	0.109	0.128	**0.091**
SNR=14	0.963	0.675	0.042	0.041	0.071	**0.031**

Chapter 5

Retrieving Signal from its Noisy Version in Digital Domain

In this chapter we focus on denoising problem in digital domain. We study the denoising problem in digital domain in two parts. In the first part, the input is noisy continues signal and we want to simultaneously denoise and quantize the noisy input. It is known that uniform quantization with a dead-zone is very close to the optimum quantization [44]. In this part, we propose a new method that generalizes the idea of dead-zone estimation to a multi-level noise removal. The second part deals with the noisy quantized/digital signal and we aim to find the best reconstruction levels to make the output as close as possible to the noiseless signal. This method pioneers in denoising a noisy quantized/digital data.

When the input of the quantization system is noisy, quantization of that noisy signal is wasting the resources to quantize noise. In the first part of this chapter we study this problem and propose a solution for it. To solve the noisy signal quantization problem, the input signal is denoised and quantized at the same time by a Two Stage Denoising (TSD) method. In the first stage, TSD detects the noise dominant part of the input signal and uses a multistage denoising/quantizing method based on MNDL to simultaneously denoise and quantize the noise dominant part. In the second stage the noiseless dominant part is quantized using one of the existing quantization methods [45].

In the second part, the input is a quantized/digital signal. Here we propose a method to find the best levels to reconstruct the noisy quantized signal. It is shown that in high SNR cases, the reconstruction levels are the same as quantization levels. However, when the input SNR is low, since the statistical properties of the signal is changed, the algorithm finds the levels in a way to minimize the reconstruction error. In this case, the reconstruction levels found by the proposed method could be different from the original levels either in number of levels or in amplitude.

Both of the proposed algorithms in this chapter are using MNDL as their core. Thus, in the next section MNDL is briefly introduced.

5.1 Minimum Noiseless Description Length(MNDL)

We use the MNDL method for the purpose of denoising. MNDL method chooses the optimum number of coefficients such that the noiseless code length is minimized [15, 46]. If use a threshold value T to keep the first m largest absolute values of the coefficients and set the rest to zero, we denote the result by θ_T. It is shown in [15] that the description length of the noiseless (desired) data with this θ_T is

$$DL(\theta_T, \bar{\theta}) = \frac{-\log(f(\theta_i|\bar{\theta}_i))}{N} = \log\sqrt{2\pi\sigma_w^2} + \frac{\log e}{2\sigma_w^2 N} z_T \tag{5.1}$$

where $z_T = ||\bar{\theta} - \theta_T||_2^2$ is the reconstruction error that is the error between the noiseless coefficients and θ_T. Since the additive noise is zero mean Gaussian, the conditional probability density function of the ith elements of the noisy coefficient is,

$$f(\theta_i|\bar{\theta}_i) = \frac{1}{\sqrt{2\pi}\sigma_w} e^{\frac{-(\theta_i - \bar{\theta}_i)^2}{2\sigma_w^2}} \tag{5.2}$$

The MNDL approach provides a probabilistic worst-case estimate for this error as a function of m, $\widetilde{DL}$ and chooses the m^* for which the DL is minimized:

$$\theta_{m^*} = \arg_{\theta_T} \min \widetilde{DL}((\theta_T, \bar{\theta})) \tag{5.3}$$

It can be seen in (5.1), when noise variance σ_w^2 is known and fixed, minimizing data length is equivalent to minimize z_T. Thus (5.3) can be written as:

$$\theta_{m^*} = \arg_{\theta_T} \min z_T(\bar{\theta}, \theta_T) \tag{5.4}$$

The result of this stage of denoising keeps m^* first largest absolute values of the coefficients and set the rest of the coefficients to zero.

5.2 Quantization of Noisy Signals

In real applications, the signals are usually corrupted with noise. Using a lossless compression method for a noisy data is not efficient, as the method uses the resources for compression of the noise along with the data. Therefore, when the input is a noisy signal, lossy compression methods are used. One of the most important stages in lossy compressions is quantization. Here we propose an effective quantization method for noisy signals.

In this problem the input signal is a noisy one shown in (2.1) and (2.4). The algorithm should find a quantized signal which is close to the noiseless signal as much as possible. Thus in this problem, the noiseless estimation ($\hat{y}$) is a quantized signal. To compress the wavelet transform of the image, it may sound logical to first denoise the wavelet coefficients with the available denoising techniques, and then compress the data with the available compression methods. However, the wavelet thresholding methods (most suitable for denoising in this scenario) can not denoise the image efficiently and a considerable amount of the additive noise is still remained in the image after this type of thresholding. In our proposed method, the denoising removes considerable amount of the noise effects through quantization. Using MNDL approach [15], a multilevel denoising approach provides a quantization value at each stage for the coefficients with noise dominant components. The first stage stops producing new quantization levels once it recognizes that the remaining coefficients are noiseless dominant. This realization is though reconstruction error estimation at each level of the denoising. In the second stage, we can use one of the existing compression approaches such as the uniform one to quantize the remaining coefficients that have large absolute values and are the noiseless dominant coefficients.

The main advantage of the proposed method is in simultaneous denoising and quantization. Our simulation results show that this method is superior to the separate denoising and compression not only by having a better output SNR but also by providing a much more compressed version of the data (less number of quantization levels in low SNRs). To show this property, a procedure is proposed to compress the hyperspectral image using the proposed quantization method, 3D-Wavelet transform and Set Partitioning in Hierarchical Trees (SPIHT).

5.2.1 Proposed Method: Two-Stage Quantizer (TSQ)

Figure 5.1 illustrates the performance of the two-stage quantizer. In this figure noisy and noiseless coefficients are depicted. As it can be seen, the noisy coefficients can be divided into two parts. The small coefficients which are highly affected by noise which are called *noise dominant* and the second part which are the larger coefficients are not changed with noise. The second part is *noiseless dominant* part.

At this stage, TSQ denoises the noise dominant coefficients. The denoising is accomplished by a multi-stage quantization. The denoising will automatically halt once it recognizes that the remaining coefficients are noiseless dominant. Next, the noiseless dominant coefficients are quantized by one of the existing quantization approaches.

Quantization Stage 1: Denoising by Multiple Stage Quantization

Multi-stage Denoising: We expand the application of MNDL for further denoising as follows. The MNDL step has kept m^* largest absolute value coefficients. Lets denote Q_1 as the kept coefficient with smallest absolute. Therefore, all the coefficients smaller than Q_1 are quantized to level zero. Using this notation, (5.4) can be used in the following form to find the first quantization levels.

$$Q_1 = \arg_{\theta_T} \min z_T(\bar{\theta}, \theta_T) \tag{5.5}$$

After the first level, there are other small coefficients left which are affected by noise but if we map them two zero it causes a big amount of error. Consequently, we should check if

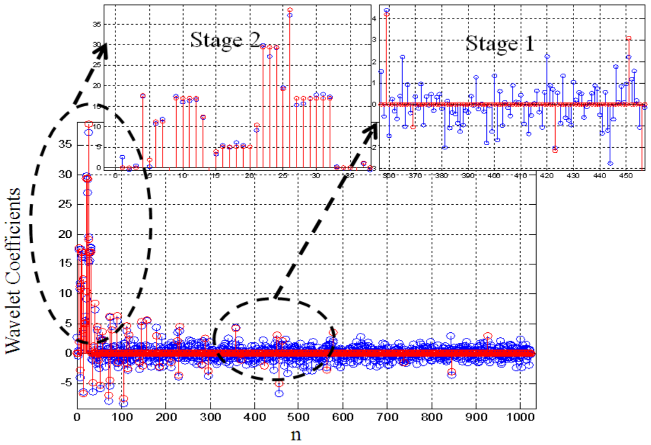


Figure 5.1: Noisy and noiseless coefficients zoomed in two distinct areas: (1) region of coefficients with small absolute values (noise dominant) that are quantized in the first stage; (b) region of coefficients with large absolute values (noiseless dominant) that are quantized in the second stage.

there are any coefficients larger than Q_1 that contain considerable amount of noise and can be quantized to the level of Q_1. To answer this question we use MNDL for further denoising as follows: For the second stage of the denoising, we search for value Q_2 among the kept coefficients, that is larger than Q_1, such that mapping all the coefficients between Q_1 and Q_2 to Q_1 minimizes the DL or equivalently z_T. Equation (5.4) can be used for this level as follows:

$$Q_2 = \arg_{\theta_{T,Q_1}} \min z_T(\bar{\theta}, \theta_{T,Q_1}) \tag{5.6}$$

where θ_{T,Q_1} is defined as:

$$\theta_{T,Q_1}(i) = \begin{cases} 0 & \text{if } |\theta(i)| < Q_1, \\ sgn(\theta_i)Q_1 & \text{if } Q_1 < |\theta(i)| < T, \\ \theta(i) & \text{ if otherwise.} \end{cases} \tag{5.7}$$

This stage denoises the data further as it makes the DL and therefore the reconstruction error smaller. Having smaller error means that the output of the second stage MNDL processing is closer to the noiseless version of the data and therefore the data has been further denoised. Consequently, the second stage has further quantized the data by denoising. Using this process the jth quantization level is calculated by:

$$Q_j = \arg_{\theta_{T,Q_{j-1}}} \min z_T(\bar{\theta}, \theta_{T,Q_{j-1}}) \tag{5.8}$$

in which $\bar{\theta}_{T,Q_{j-1}}$ is defined as follows:

$$\theta_{T,Q_{j-1}}(i) = \begin{cases} \theta_{T,Q_{j-2}}(i) & \text{if } |\theta(i)| < Q_{j-1}, \\ sgn(\theta_i)Q_{j-1} & \text{if } Q_{j-1} < |\theta(i)| < T, \\ \theta(i) & \text{ if otherwise.} \end{cases} \tag{5.9}$$

This process continues until the output of the MNDL denoising is the same as the input and in order to minimize the MDNL, almost all the input coefficients are kept at level M. On the other word, once we get to the quantization level Q_M, the MNDL process stops providing any further levels larger than Q_M by finding no minimum value in the bound which means all the remaining coefficients are important.

This indicates that the reconstruction error can not any further be minimized by denoising. Thus, the remaining undenoised coefficients at this stage have noiseless dominant components and no further denoising is needed.

This part can either be used as a denoised version of the input signal on itself or it can be used as the input of the second stage for further quantization. In simulation part both of the cases are tested.

Quantization Stage 2: Noiseless Dominant Coefficient Quantization

We have denoted the last quantization level found by the multi-stage MNDL denoising process as Q_M. From the previous stage, it is known that the kept coefficients, that have larger absolute value than Q_M, are noiseless dominant. Thus, for this stage of quantization we can use one of the conventional methods of quantization such as high rate uniform scalar quantizer, since it is near optimal [44].

5.2.2 Experimental Results

In this part we plan to quantize a noisy signal to make it ready for efficient lossy compression. Hence the compression is more common in images, we test the proposed algorithm with noisy images.
We test the algorithm in two ways. First of all, we apply the algorithm to denoise the images shown in Figure 5.2. The output SNR of the proposed denoising algorithm (first stage of the algorithm introduce above) is compared with a simple MNDL, SureShrink and BayesShrink. As it can be seen in table 5.1 the SNRs are improved.

Figure 5.2: The images which are used to test the algorithm, from top-right clockwise: Lena, Barbara, Baboon and CameraMan.

In second test, we use the proposed algorithm to quantize noisy hyperspectral image. Then these quantized hyperspectral images are compressed with Set Partitioning in Hierarchical Trees (SPIHT) and the output SNR of the proposed method is compared with uniform quantization of the noisy data and uniform quantization of the denoised data with

VisuShrink. Thus, we have a lossy compression algorithm which contains 1) MNDL multilevel denoising and quantization, 2) Uniform scalar quantization, and 3) SPIHT compression. Note that since our input images are already noisy, we have two types of SNRs in our analysis; one for the input image, denoted by input SNR (ISNR), and one for the output image, denoted by peak SNR (PSNR) defined in 2.8.

Multistage Denoising of the Test Images

At this part we use MNDL iteratively as it is described in section 5.2.1. We use the images shown in figure 5.2 with SNRs of 5, 10, 20, and 30 while image size is 512×512.
Figure 5.3 shows different stages of the proposed algorithm for Mandrill image when SNR is 10. The first stage of the algorithm (Multi-Stage denoising) finds three levels for noise dominant part and it halts after the three stages.

The available image thresholding methods are SureShrink [8] and BayesShrink [10]. To test the proposed algorithm in Table 5.1 we compare the proposed method with MNDL, SureShrink and BayesShrink. MSE values are calculated for 50 runs of the different denoising methods. The results in this table show the strength of the multistage MNDL denoising in comparison with the other denoising methods. The PSNR of the proposed method outperforms the other methods except for Lena in which BayesShrink is working better than the other methods. Although the multistage denoising has better performance in comparison with normal MNDL, it is more complex and a time consuming procedure since it applies MNDL iteratively.

Since the PSNR of the proposed multi-stage denoising algorithm is more than the PSNR of the other methods, using any quantization method for the noiseless dominant part we get a closer output from the proposed method in comparison with the other denoising methods.

Hyperspectral Image Compression

In this part, TSQ is used in combination with 3D wavelet transform and Set Partitioning in Hierarchical Trees (SPIHT) to compress Hyperspectral Images. Since the Hyperspectral

Table 5.1: PSNR of applying multistage MNDL denoising on the test images shown in figure 5.2 and comparing it with one levels MNDL.

Image	CameraMan				Lena			
ISNR	5	10	20	30	5	10	20	30
Proposed method	22.63	26.01	33.77	42.33	23.80	27.12	35.14	44.11
One Level MSE	18.89	23.11	32.56	42.37	20.54	25.07	35.01	43.99
BayesShrink	22.04	25.44	32.34	42.38	23.69	27.20	35.79	44.67
SureShrink	17.57	22.31	25.73	26.01	19.65	24.67	27.91	28.19

Image	Barbara				Mandrill			
ISNR	5	10	20	30	5	10	20	30
Proposed Method	25.43	28.84	37.55	50.43	23.26	26.54	35.05	46.5
One Level MSE	25.11	28.34	35.53	48.38	23.01	26.47	35.53	45.88
BayesShrink	25.25	29.07	37.54	50.43	22.25	26.81	36.54	46.43
SureShrink	23.65	28.15	31.32	40.8	21.43	25.46	26.67	26.8

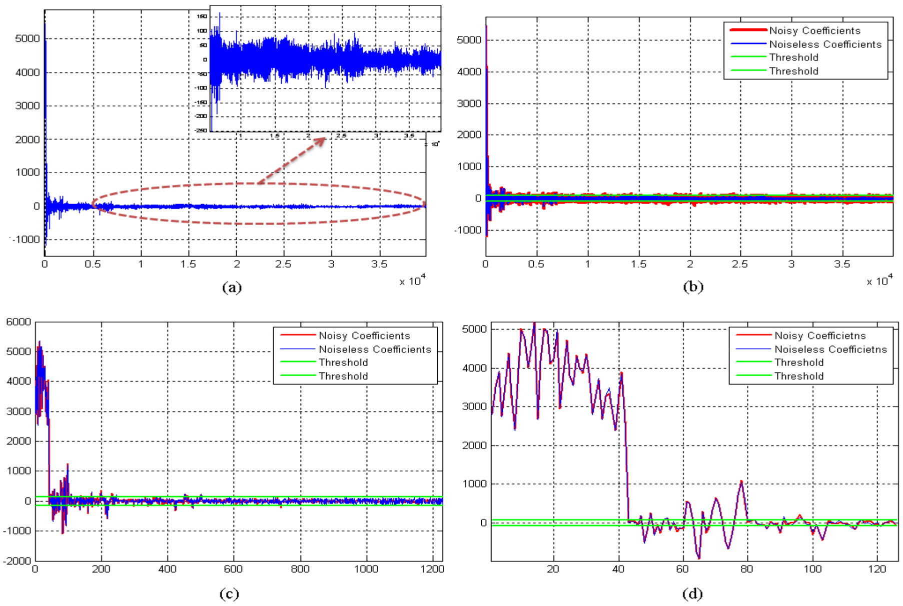


Figure 5.3: Multistage denoising process for Mandrill (SNR=10). (a) noiseless wavelet coefficients of Mandrill, (b) the first threshold value provided by MNDL, (c) second level in multistage denoising, and (d) last stage after which the process is halted.

image is a 3D data, as it can be seen in figure 5.4, to find its wavelet coefficients, 3D wavelet transform is used. Before showing the results, 3D wavelet transform and SPIHT are briefly introduced.

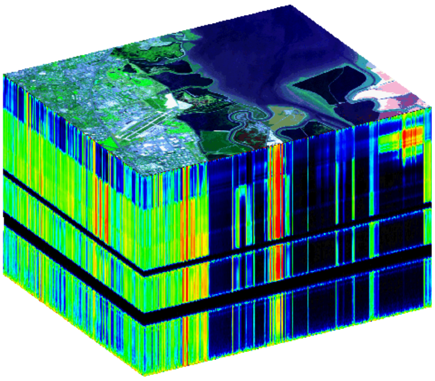


Figure 5.4: AVIRIS hyperspectral image, used in simulations.

3D wavelet transform: Hyperspectral image is a 3D combination of 2D images captured in different spectral bands. Since all the images are captured from a single scene, the spectral bands are highly correlated. To reduce the amount of the correlation between the bands it is helpful to use 3D-wavelet transform. 3D wavelets can be constructed as separable products of 1D wavelets by applying a 1D-wavelet transform in three directions (x,y,z).

if the data is of size $N_1 \times N_2 \times N_3$, then after applying the 1D analysis filter bank to the first dimension we have two subband data sets, each of size $\frac{N_1}{2} \times N_2 \times N_3$. After applying the 1D analysis filter bank to the second dimension we have four subband data sets, each of size $\frac{N_1}{2} \times \frac{N_2}{2} \times N_3$. Applying the 1D analysis filter bank to the third dimension gives eight subband data sets, each of size $\frac{N_1}{2} \times \frac{N_2}{2} \times \frac{N_3}{2}$.

Set Partitioning in Hierarchical Trees: For bit plane mapping and compression of the quantized coefficients, we use SPIHT [47]. SPIHT is one of the most efficient image compression algorithms. We selected SPIHT because SPIHT and its predecessor, the embedded zero-tree wavelet coder, are of the most efficient image compression algorithms. In addition, they have better quality in comparison with vector quantization and JPEG. It does not require training and producing an embedded bit stream. The effectiveness of the SPIHT algorithm originates from the efficient subset partitioning and the compact form of the significance information. The SPIHT algorithm defines spatial orientation trees, sets of coordinates, and recursive set partitioning rules. The algorithm is composed of two passes: a sorting pass and a refinement pass. It is implemented by alternately scanning three ordered lists: list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP). Which are defined as follows:

LIS, List of insignificant sets: contains sets of wavelet coefficients which are defined by tree structures, and which had been found to have magnitude smaller than a threshold (are insignificant). The sets exclude the coefficient corresponding to the tree or all sub-tree roots, and have at least four elements.

LIP, List of insignificant pixels: contains individual coefficients that have magnitude smaller than the threshold.

LSP, List of significant pixels: pixels found to have magnitude larger than the threshold (are significant).

The algorithm codes the most important wavelet transform coefficients first, and transmits the bits so that an increasingly refined copy of the original image can be obtained progressively. The partial ordering of the transform coefficients is a result of comparisons of coefficient magnitudes to a set of octavely decreasing thresholds, with the initial threshold being the largest power of 2 which is smaller than the magnitude of the largest coefficient. At any time, coefficients with magnitudes larger than the current threshold are considered to be significant. During the sorting pass the significance of LIP and LIS are tested, followed by removal (as appropriate) to LSP and set splitting operations to maintain the insignifi-

Table 5.2: PSNR of compressed images with different methods and for different Input SNR (ISNR).

Compression Rate	Noiseless	Noisy	Proposed method	Denoise & Quantize
ISNR=20				
16:1	30.82	28.76	**30.23**	28.91
8:1	35.93	31.22	**34.6**	33.15
4:1	42.72	35.16	**40.7**	38.98
ISNR=10				
16:1	30.82	22.43	**28.14**	28.05
8:1	31.42	23.9	**31.416**	30.34
4:1	42.72	27.72	**39.88**	38.13
ISNR=5				
16:1	30.4	18.04	26.83	**27.13**
8:1	35.93	19.7	**30.2**	29.91
4:1	42.72	24.06	**37.08**	36.18

cance property of the lists. In the refinement pass, the ith most significant bits in the LSP, which contains the coordinates of the significant pixels, are scanned and output. The SPIHT algorithm reduces the threshold and repeats the two passes until the bit budget is met.

The hyperspectral data collected by the AVIRIS satellite from the Indian Pines test site in northwestern Indiana is considered [48]. The data set is composed of 185 spectral bands and contains 145 × 145 pixels (21025 pixels). SPIHT is applied on four types of data: 1) noiseless images, 2) noisy images, 3) images quantized with the proposed two stage method, and 4) noisy image that is first denoised with BayesShrink and then quantized with the near uniform quantizer [44]. In Table 5.2, PSNR of the output images are given. To decrease the time of processing the images are resized to 50 × 50.

As the table shows the PSNR of the outputs are improved by using the proposed method compare to the results of the method that first denoises with BayesShrink and then quantizes with a uniform quantizer [44]. In the proposed approach, for large ISNR, such as ISNR > 100, almost all the coefficients are automatically quantized in the second stage, while for ISNR≤ 20, more than %90 of the coefficients are quantized in the first stage. For example when ISNR is 20, about %93 of the coefficients are quantized in only four levels.

5.3 Retrieving Quantized Signal from Its Noisy Version

In this part we propose an algorithm to retrieve a quantized data from its noisy version. In conventional scalar quantization encoders, source signal is transformed using a transform such as wavelet transform to reduce the spatial redundancies between the samples, and to improve the performance of the quantization system. Accordingly, the transformed coefficients are quantized. Thus, in a quantization/compresion system we are dealing with quantized noisy wavelet coefficients. However, the proposed algorithm can also be used directly on the noisy quantized signal (in spatial domain). To find the optimum quantization levels, a multistage process minimizes MSE at each level by using MNDL algorithm [15]. Consequently, the procedure denoises and recovers the quantized data simultaneously.

Assume that the noiseless quantized coefficient vector

$$\bar{\theta} = [\bar{\theta}(1), \bar{\theta}(2), ..., \bar{\theta}(N)]^T \tag{5.10}$$

of length N, is corrupted by an additive noise v (2.4). In this problem the input signal is quantized so $\theta = [\theta(1), \theta(2), ..., \theta(N)]^T$ is the available noisy quantized coefficients. The noise coefficient, $v(i)$, is zero mean additive white Gaussian noise (AWGN) with variance of σ_w^2. If the noise variance is unknown, it is estimated by MAD method [6].

The main difference of this proposed algorithm with the other denoising algorithms is the prior knowledge, that the desired signal is a quantized one. No other information, such as the quantization levels or the number of the quantization levels, is available about the data. The applications of the method is in widespread digital signal reconstruction from its noisy version.

The main advantage of the proposed method is that not only it denoises the data, but also using the extra prior assumption, the denoised data is itself a quantized data with optimum

MSE. Independent from the quantization method (such as Lloyd-Max [49, 50], entropy-constrained quantizer [51] or any other type of quantizer), we would like to denoise the data with the only knowledge that the original data is a quantized one.

5.3.1 Proposed Method: Multiple Stage Denoiser and Quantizer (MSDQ)

The goal in the quantization problem is to minimize the distortion for a given rate. In quantizer side, the source signal is quantized using one of the known quantization methods which satisfies the Rate-Distortion constraint. In the quantization decoder side, to find the best estimate of the quantization levels we use MNDL basis selection algorithm [15]. As it was said before, MNDL provides an upper and a lower bound for the MSE. Using these two bounds, we can keep a number of basis to reach the minimum distortion [15].

In low noise conditions, the T_is shown in Figure 5.5 are clearly distinguished to be found using MNDL. However, in noisy conditions as shown in Figure 5.6, the statistics features of the quantized signal are completely changed and dequantization of such a signal with the same levels used for quantization, increases the difference between the source signal and the dequantized one. In this condition MNDL chooses the levels in a way to minimize the MSE, which can be different from the levels used in quantizer.

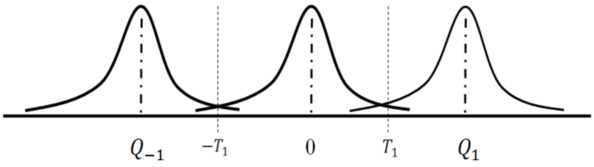


Figure 5.5: Distribution of the noisy quantized data, θ_i, around noiseless quantized data $\bar{\theta}_i$ in which $Q = [Q_1, .., Q_M]$ is the quantization levels of the noiseless coefficients and $T = [T_1, .., T_{M-1}]$ is the decision.

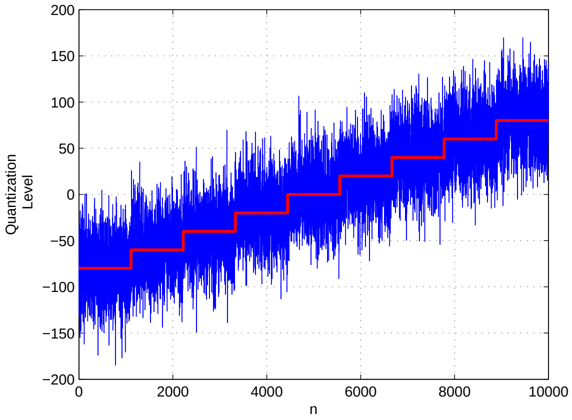


Figure 5.6: Noiseless and noisy coefficients sorted based on the quantization levels when there are 9 levels and SNR is 5.

First Quantization/Denoising Stage

In the proposed algorithm, MNDL is iteratively applied on noisy quantized coefficient. The first step of the algorithm is different since in this step we should map the coefficients lower than the first found threshold value to zero. This is exactly the same as MNDL thresholding. However in the next steps, we find the quantization level and map the data between the two adjacent threshold values to the corresponding quantization level. Thus, we should adapt the MNDL with this change.

For the first step, consider the denoising via hard thresholding. In this case the data with threshold T provides

$$\hat{\theta}_T^{(1)}(i) = \begin{cases} 0 & \text{if } \theta(i) < T, \\ \theta(i) & \text{if otherwise.} \end{cases} \tag{5.11}$$

The superscript (1) indicates the output of the first step of the process.

The MNDL approach provides a probabilistic worse-case estimate of the description length, $\widetilde{DL}$ defined in (5.1), and chooses the threshold that minimizes this criterion:

$$T_1 = \arg_T \min \widetilde{DL}((\hat{\theta}(T), \bar{\theta})) \tag{5.12}$$

This first step thresholding provides T_1 which is the optimum quantization level boundary around zero that minimizes the description length of the noiseless data.

Iterative Quantizarion/Denoising Process

In the presence of Gaussian noise the quantized noisy coefficients are distributed around the noiseless coefficients as it is shown in Figure 5.5. The involved distribution of the noisy coefficient is given in (5.2). In the current case, $\bar{\theta}(i)$ is an element of the set of M level of quantization $Q = [Q_1, ..., Q_M]$. In the first stage we used this property to estimate the set of values that can best be mapped to level zero by denoising the data. To estimate the second quantization level, we can use the thresholded data from the first level as follows. The input of the second stage of the quantizer is

$$\hat{\theta}^{(2)}_{[T_1,T]}(i) = \begin{cases} 0 & \text{if } |\theta(i)| < T_1 \\ \frac{\sum_{i\in\{i:T_1\leq\theta(i)<T\}}\theta_i}{\#\{i:T_1\leq\theta(i)<T\}} & \text{if } T_1 \leq |\theta(i)| < T \\ \theta(i) & \text{otherwise.} \end{cases} \tag{5.13}$$

In the second stage we find the optimum T that minimizes the noiseless description length of the data

$$T_2 = \arg_T \min \widetilde{DL}((\hat{\theta}^{(2)}_{[T_1,T]}, \bar{\theta})) \tag{5.14}$$

This will provide the first quantizing level estimate which is

$$\hat{Q}_1 = \frac{\sum_{i\in\{i:T_1\leq\theta(i)<T_2\}}\theta_i}{\#\{i : T_1 \leq \theta(i) < T_2\}} \tag{5.15}$$

This quantization level is found based on the centroid of area under probability distribution function (pdf) [49, 50]. We shall continue this iterative process to estimate all the quantization levels. For the jth step of the quantization we have

$$\hat{\theta}^{(j)}_{[T_1,\cdots,T_{j-1},T]}(i) = \begin{cases} \theta^{(j-1)}_{[T_1,\cdots,T_{j-1}]}(i) & \text{if } |\theta(i)| < T_{j-1} \\ \frac{\sum_{i\in\{i:T_{j-1}\leq|\theta(i)|<T\}}\theta_i}{\#\{i:T_{j-1}\leq|\theta(i)|<T\}} & \text{if } T_{j-1} \leq |\theta(i)| < T \\ \theta(i) & \text{otherwise.} \end{cases} \tag{5.16}$$

The superscript shows the stage number and the subscript indicates the decision levels up to that stage.

$$T_j = \arg_T \min \widetilde{DL}((\hat{\theta}^{(j)}_{[T_1,\cdots,T_{j-1},T]}, \bar{\theta})) \tag{5.17}$$

That will provide the $j-1$th quantizing level estimate

$$\hat{Q}_{j-1} = \frac{\sum_{i\in\{i:T_{j-1}\leq|\theta(i)|<T_j\}} \theta_i}{\#\{i : T_{j-1} \leq |\theta(i)| < T_j\}} \tag{5.18}$$

which is the mean value of the samples in each interval.

We can use our prior knowledge that the signal is already quantized, to optimize the MNDL. In MNDL, the threshold values are the absolute value of the sorted wavelet coefficients. When the signal is quantized using a scalar quantizer, since the coefficients in each bin have the same value, the threshold values for each bin have the same value. Thus, we can use this property to find the threshold values more accurately. Figure 5.7 shows this property for a signal with high and low SNRs. As it can be seen in this figure, we can use the second derivative of the bound to find the best threshold values and reconstruction levels. When the bounds are changed, a local maximum happens in the second derivative which can be used as the threshold value in that stage. Thus, the last pick in the second derivative is used to choose the coefficient which should be mapped at each stage.

5.3.2 Simulation Results

To test the algorithm, the analytical signals shown in Figure 5.8 are used with different SNRs. Figures 5.9 and 5.10 show the distribution histogram of the noisy samples of the first analytical test signal. Since the noise is Gaussian, the samples are normally distributed around the quantization levels. In high SNRs as it can be seen in Figure 5.9, in which SNR is 30, the levels are completely distinguishable. As it is shown in that figure, in these cases the algorithm finds exactly the original quantization levels.

However, when the SNR is lower such as the case shown in Figure 5.10, where SNR is 15, the levels are mixed. It means the level of some of the noisy samples is changed and it

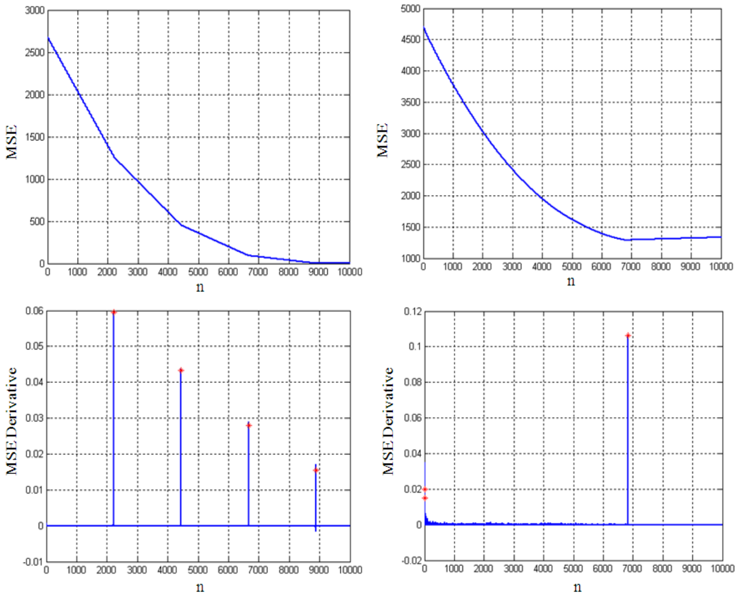


Figure 5.7: Top left: Upper bound of Z_T provided by MNDL and Bottom left: its second derivative for a high SNR case, Top Right: Upper bound of Z_T found by MNDL for a low SNR signal and Bottom Right: its second derivative. Where the bins are changed, local maximums happens. We use these maximums to find the thresholds in MNDL.

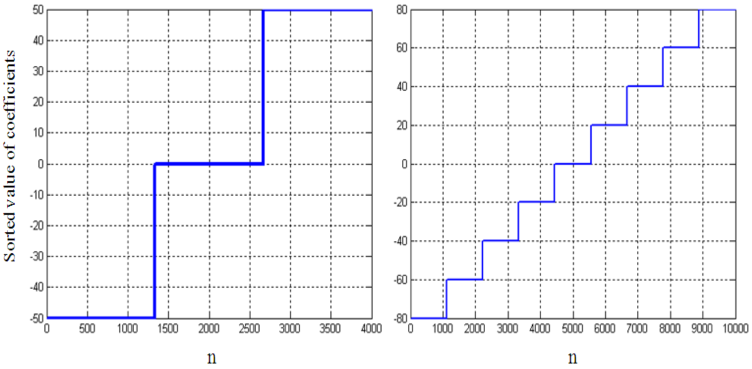


Figure 5.8: The test signals used to test the proposed algorithm.

is not possible to find the original level which the sample belongs to. In the case shown in figure 5.10, the levels change is started but they are still recognizable. In the low SNR cases the algorithm finds some small threshold values which is around the local maximums in the valleys between the two quantization levels. When the samples are not very low SNR, such as this example in which SNR is 15, using a post processing these levels can be pruned. For example here we simply remove the thresholds which are smaller than the $\frac{1}{3}$ of the mean of the all the found thresholds. Using this post processing, for this example the original quantization levels are found.

When SNR is very low, like figure 5.11, it is almost impossible to retrieve the original levels from the noisy samples. In these cases the proposed algorithm finds a set of levels which causes a lower error than the original levels in some cases. In other word, if we use the original levels to find the threshold values, for example using *Lloyd-max* criterion $T_i = \frac{Q_i+Q_{i+1}}{2}$, and map the noisy samples on the original levels the MSE would be more than when we use the proposed method to find the best levels.
This test is repeated with the 3-level quantized signal. The retrieved results are confirmed for this signal also. In Figure 5.12, SNR is 30 and the levels are the same with the original ones. However in Figure 5.13, SNR is 10 and the found levels are different from the original ones, but using the original levels the Normalized MSE (NMSE) defined in (2.7) is 0.0727 which is larger than NMSE of using the levels found by proposed algorithm, which is 0.0650.
In tables 5.4 and 5.3, the NMSE of test signal dequantization using the original levels and the levels found by the proposed algorithm are compared.
It may sound logical to use the existing denoising thresholding methods first to get rid of small values of noise and then quantize the denoised data. Thus, these methods are compared in the tables also. Since the only method which gives a universal threshold value is Universal Threshold [6] we have used this method here.

As it can be seen in most cases the propose method has better results even in comparison with the original levels. In three level signal, the levels are pretty distinguishable for SNRs

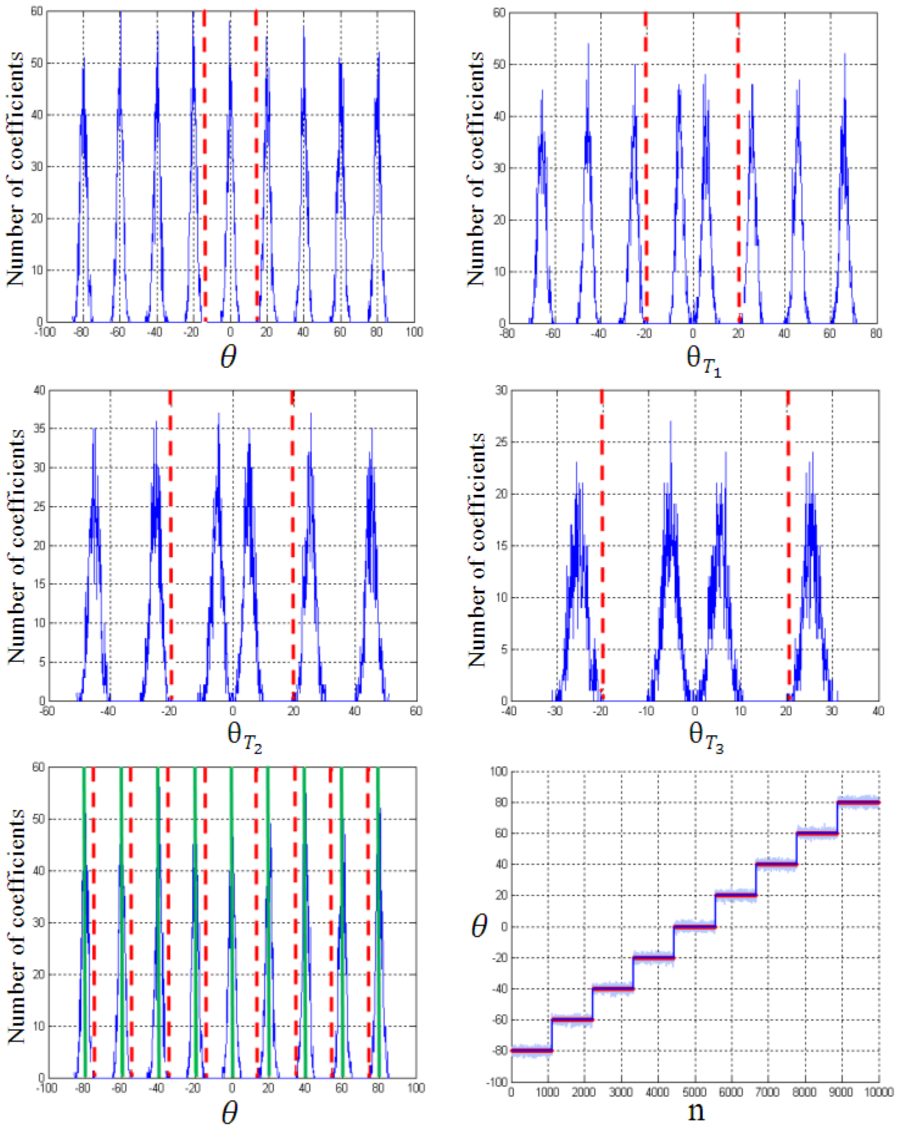


Figure 5.9: The histogram of the noisy coefficients (SNR=30) and the threshold value (dashed red line) at each level are shown in this figure. When SNR is high at each level we find one of the quantization levels. All the quantization levels (solid green line) and the threshold values (dashed red line) are shown in Bottom left figure. Bottom right, shows the denoised quantized signal.

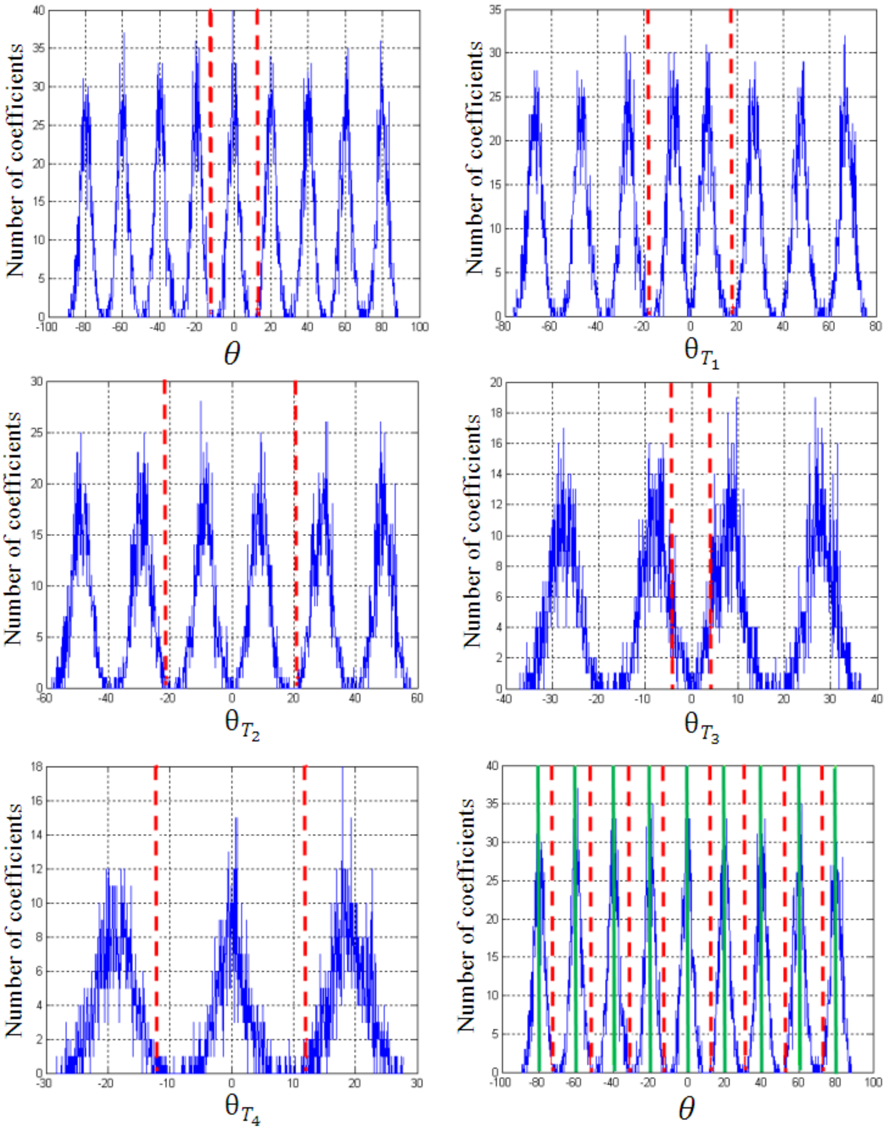


Figure 5.10: The histogram of the noisy coefficients (SNR=10) and the threshold value (dashed red line) at each level are shown in this figure. When SNR is low we could have some extra levels. In a post-processing step (using a pruning algorithm) some of them will be removed. The quantization levels (solid green line) and the threshold values (dashed red line) are shown in the figure at Bottom right.

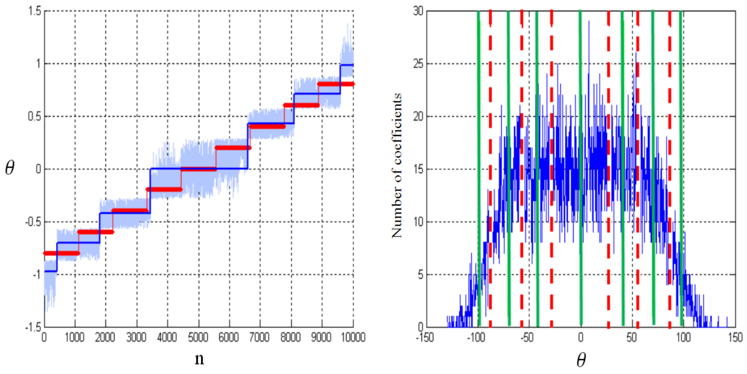


Figure 5.11: Dequantized signal with the proposed method when SNR is 5. The levels are different from the original levels. However, the error (MSE) is lower. Left: Levels found by proposed algorithm are shown by dark blue, red line shows the original levels, and the light blue shows the noisy coefficients at each level. Left: Dark Blue shows the histogram of the noisy coefficients, dashed Red lines are the threshold values and the solid green lines are the quantization levels.

Table 5.3: NMSE of dequantizing the signal with 9 levels shown in figure 5.8 with four different method: (1) Using the original levels (2) Using the proposed method (3) Denoising with VisuShrink and then dequantizing with the original levels (4) Denoising with MNDL and then using the original levels to dequantize the signal.

	5	10	15	20
Proposed Method	0.2869	0.0709	0.0261	0.0043
Using Original levels	0.2724	0.0999	0.0401	0.0078
Denoised by visu	0.2885	0.1090	0.0386	0.0077
Denoised by MNDL	0.4126	0.1024	0.0384	0.0074

Table 5.4: NMSE of dequantizing the signal with 3 levels shown in figure 5.8 with four different method: (1) Using the original levels (2) Using the proposed method (3) Denoising with VisuShrink and then dequantizing with the original levels (4) Denoising with MNDL and then using the original levels to dequantize the signal.

	5	10	15	20
Proposed Method	0.3362	0.0668	0.00103	0
Using Original levels	0.3521	0.0705	0.0011	0
Denoised by visu	0.3490	0.0705	0.0011	0
Denoised by MNDL	0.3387	0.0701	0.0011	0

larger than 20 and the error is almost zero, either by using the proposed method or the original levels.

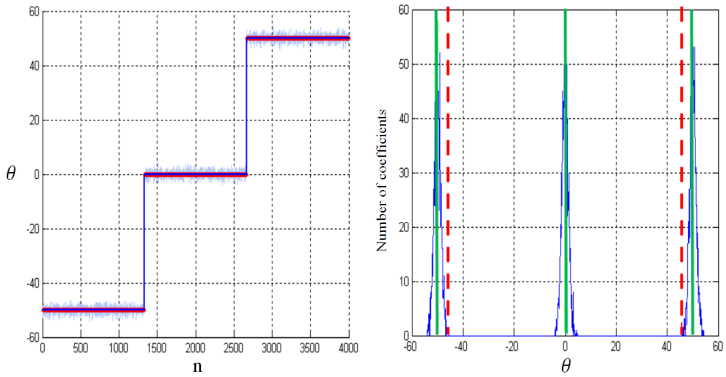


Figure 5.12: Dequantized signal with the proposed method when SNR is 30. The levels are different from the original levels but the error (MSE) is lower. Left: Levels found by proposed algorithm are shown by dark blue, red line shows the original levels and the light light blue shows the noisy coefficients at each level. Left: Dark Blue shows the histogram of the noisy coefficients, dashed Red lines are the threshold values and the dotted green lines are the quantization levels.

5.4 Concluding Remarks

In this chapter we proposed two methods to retrieve signal from its noisy version in digital domain. The difference between these two methods is the type of the input signal. In the first part we proposed a two-stage quantization and compression method for noisy images. In the second part we proposed a denoising algorithm for when our prior knowledge on the desired data is that the signal is digital.

We used the "Two-stage quantization of noisy signals" to compress the hyperspectral image, the 3D-wavelet transform was used. At the first stage of the proposed method, multistage MNDL denoising was used to quantize the noisy coefficients. The algorithm halts once the level of quantization is comparable with noiseless dominant coefficients. At the second stage, the remaining noiseless coefficients were quantized by a uniform quantizer, which is a near optimal quantizer. The results showed the strength of the proposed approach in providing

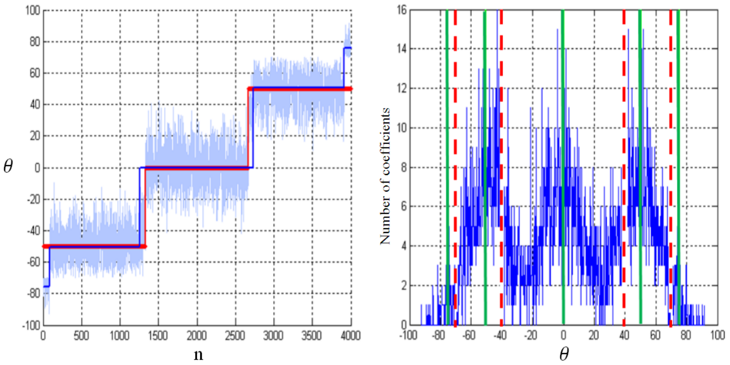


Figure 5.13: Dequantized signal with the proposed method when SNR is 10. The levels are different from the original levels but the error (MSE) is lower. Left: Levels found by proposed algorithm are shown by dark blue, red line shows the original levels and the light light blue shows the noisy coefficients at each level. Left: Dark Blue shows the histogram of the noisy coefficients, dashed Red lines are the threshold values and the solid green lines are the quantization levels.

the minimum number of quantization levels with the best (maximum) value of quantization PSNR.

In the second part we aimed to find the optimum digital estimate of the noiseless data using the available noisy data. The existing denoising methods of a digital signal need additional assumptions for example about the quantization levels of the data, the quantizer type, or at least the number of levels. The proposed denoising algorithm pioneers in using only the knowledge that the desired data is quantized with no additional assumption. The method only denoised the data, but provided the optimum quantized estimate of the original signal in the MSE sense. The well known denoising methods in this scenario were the thresholding ones that only set small coefficients, attributed to the noise, to zero. None of the existing denoising approaches were able to use the extra prior knowledge on the data that was its quantized nature. In the proposed method, a multistage denoiser implemented the MNDL algorithm at each stage to provide the optimum quantized levels of the data estimate.

Through the simulations we elaborated that the quantized estimate of the desired data in high SNRs had the same number of quantization levels as the noiseless data. However, the optimum number of levels of the signal estimate decreased as the SNR decreased. In very low

SNRs the additive noise corrupted the data to the level that the algorithms output has very few quantization level. This is an important advantage of the proposed multistage denoiser that is robust to the noise and variations of the noise variance.

This method treated the additive noise at nonzero levels as the quantization noise rather than the additive noise. In our approach, each stage optimally removed the additive noise at the next quantization level. As a result, the estimate provided by the proposed method had smaller MSE compare to the former approach. We illustrated this fact in our simulation results.

Chapter 6

Noise Variance Estimation in BayesShrink

Almost all denoising approaches rely heavily on the unknown value of the additive noise variance. Consequently, the first step of denoising is dedicated to estimating the noise variance using the same available data that needs to be denoised. The noise variance estimate which is proposed in [6] by Donoho and Johnstone is currently used as an standard method for noise variance estimation. As it was seen in section 2.3.1, in this method noise variance is estimated through a method called Scaled Median absolute Deviation (MAD) computed from the high-pass wavelet coefficients of the first level of the transform (the diagonal direction of decomposition level one). As its name suggests the MAD is the median of the absolute values of the coefficients. The MAD-based methods are useful especially for the sparse signal which have a small amount of signal energy in the details subbands. Some signals such as *Baboon* image, which can be seen in Figure 5.2, contains a lot of high frequency and detail coefficients so *MAD-based* method is not useful for these cases. In table 6.1 you can see the median of the detail coefficients of the images in Figure 5.2.

Table 6.1: Median of the detail coefficients of the test images shown in figure 5.2.

Images	Baboon	Barbara	Lena	CameraMan
Median	5.5	2	1.5	1.5

As it can be seen this value is high in *Baboon* even when it is noiseless, which causes a big error when MAD is used for noise variance estimation.

In [52] a noise variance estimation other than MAD is used based on the universal threshold and VisuShrink, since the VisuShrink is optimal in a Min-Max sense. In this method the Universal threshold is computed for the noisy coefficients using MAD for noise variance estimation. Then, it keeps the coefficients which are smaller than the threshold value as the noise coefficient. In an iterative algorithm, using MAD and Universal thresholding the noises are pruned until the result converges. These remaining coefficients are considered as noise and its variance is used as the estimation of the noise variance.

Some other methods are proposed for noise variance estimation but most of them are applicable in special cases such as power systems, OFDM systems and *etc.* [53, 54, 55].

In [56, 46] a noise variance estimation method is proposed based on MNDL. This method looks for the noise variance which makes the MSE, minimum. An interval, where typically the noise variances happens in that interval, is considered and for each noise variance the minimum MSE is estimated by MNDL. Comparing these MSE, the best noise variance is the one which makes the MSE minimum.

In this chapter we propose a method called *Residual Autocorrelation Power* (RAP), which is an adaptive method for noise variance estimation to improve the BayesShrink denoising [10, 9]. *RAP* uses the differences between the autocorrelation of noisy image and the pure noise to estimate the noise variance [57]. BayesShrink finds a threshold that minimizes the Bayesian Risk based on the assumption that the images have properties of a *Generalized Gaussian signal.* This method has been used in different applications to denoise the input signals [58, 59, 60] and is one of the best available denoising methods.

Our propose method uses BayesShrink to denoise the signal with a set of considered noise variances. If we are using the right noise variance value, the removed part should be noise. To check which noise variance is the best estimation of the real noise variance, we use the autocorrelation of the removed part (the coefficients bellow the threshold value.). Figure 6.1

shows the coefficients used to computed the autocorrelation. The coefficients smaller than T and larger than $-T$ are kept.

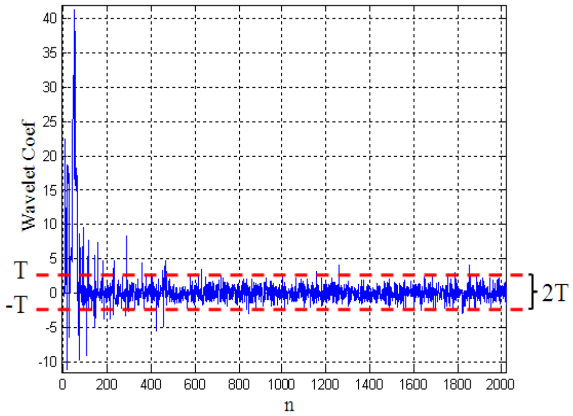


Figure 6.1: Noisy wavelet coefficient of a sample signal and the corresponding threshold value found by BayesShrink. In the proposed algorithm the coefficients in the $2T$ interval are kept as noise coefficients.

Formally, the discrete autocorrelation R at lag j for signal x_n is

$$R(j) = \sum_n (x_n - m)(x_{n-j} - m) \tag{6.1}$$

where m is the mean value of x_n. For a two-Dimensional signal, $x_{n,r}$, autocorrelation can be computed by

$$R(j,i) = \sum_{n,r} (x_{n,r} - m)(x_{n-j,r-i} - m) \tag{6.2}$$

Informally, autocorrelation can be thought as the similarity between observations as a function of the time separation between them.

6.1 Residual Autocorrelation Power (RAP)

In RAP we aim to find the noise variance using BayesShrink and Autocorrelation property of Gaussian noise. We know Autocorrelation of Gaussian noise is an impulse function and power of this autocorrelation is smaller than other autocorrelations.

We expect that BayesShrink removes the noise and keeps the noiseless signal. In addition, threshold value of BayesShrink is a function of noise variance and noiseless signal variance. Thus, if we use a good estimation of noise variance in BayesShrink to denoise the noisy signal, the difference between the input and output signal will be a good estimation of the additive noise.

In our proposed method, we use different noise variances to calculate the BayesShrink's threshold and apply it to the noisy signal. In the next step we find the difference between the noisy and denoised signal which is the removed noise by BayesShrink. If the noise variance, used in BayesShrink, is close to the variance of the noise which has corrupted the input signal, we expect to have a small power.

6.1.1 Gaussian Noise Autocorrelation

The Additive White Gaussian Noise (AWGN) has an important property which can be utilized in the process of variance estimation. It is known that the autocorrelation of this noise is of the following form:

$$R_{ww}[i,j] = \sigma^2 \delta[i,j] \tag{6.3}$$

where δ is the dirac delta function that is one at $i = 0, j = 0$ and zero at all the other locations $i \neq 0, j \neq 0$. This property of the additive noise can be used in evaluating the denoising error.

While the autocorrelation of the noise is an impulse, the autocorrelation of the noise-free image is different from an impulse. The autocorrelation of a 2D zero mean AWGN and that of the Lena image (noiseless and noisy) are shown in Figures 6.2, 6.3 and 6.4.

In the following this property of the AWGN is employed in noise variance estimation.

As it was seen in section 2.4.3, BayesShrink [10, 9] aims to minimize a Bayesian Risk and proposes a subband dependent threshold for image denoising which is a function of noise variance σ_w^2 and noiseless standard deviation, given in (2.25).

Since the noise variance and the noiseless standard deviation are unavailable the threshold

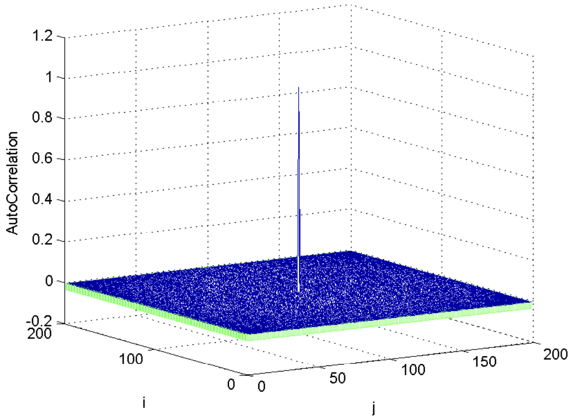


Figure 6.2: Autocorrelation of a 2D-zero mean AWGN with $\sigma = 1$.

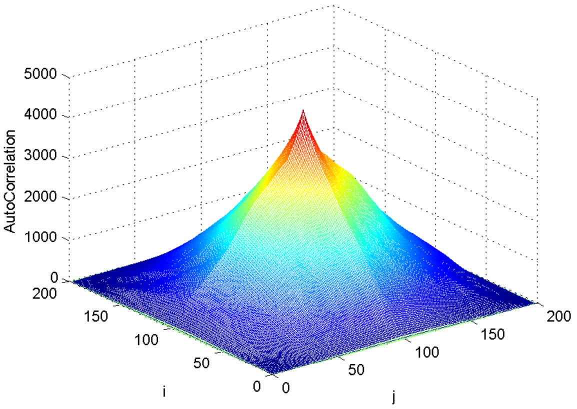


Figure 6.3: Autocorrelation of noise-free Lena.

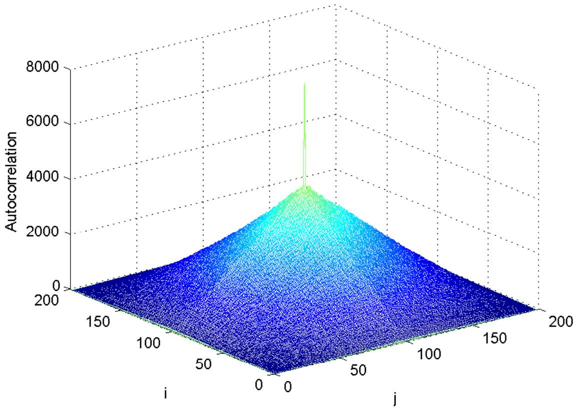


Figure 6.4: Autocorrelation of noisy Lena with $\sigma^2 = 10$.

is estimated as follows

$$T(\hat{\sigma}_w) = \frac{\hat{\sigma}_w^2}{\hat{\sigma}_{\bar{y}}} \tag{6.4}$$

where $\hat{\sigma}_w$ is the noise variance estimate using the MAD-based approach and the estimate of the subband noiseless coefficient std is

$$\hat{\sigma}_{\bar{y}} = \sqrt{max(\sigma_y^2 - \hat{\sigma}_w^2, 0)} \tag{6.5}$$

where θ is the available subband coefficients of the noisy observed signal.

6.1.2 Residual Autocorrelation Power (RAP) and Noise Variance Estimation

Consider a range of candidates for noise variance $\Sigma = [\sigma_1, \sigma_2, ..., \sigma_m, ..., \sigma_M]$ (for example from zero to 20 with steps of 0.1). For each noise variance σ_m, use the BayesShrink and denoise the data with threshold

$$T(\sigma_m) = \frac{\sigma_m^2}{\hat{\sigma}_{\bar{\theta}_m}} \tag{6.6}$$

where $\hat{\sigma}_{\bar{y}}$ is a subband stddev estimate using σ_m as the noise stddev.

$$\hat{\sigma}_{\bar{\theta}_m} = \sqrt{\max(\hat{\sigma}_{\bar{y}} - \sigma_m^2, 0)} \tag{6.7}$$

The error associated to this denoising is

$$\hat{w}_m = y - \hat{y}_m \tag{6.8}$$

where $\hat{y_m}$ is the denoised data. Therefore, the autocorrelation of the residuals is $R_{\hat{w}_m \hat{w}_m}$ and the l_2 norm of this error is

$$P_m = \frac{1}{N} \sum_{i,j} (R_{\hat{w}_m \hat{w}_m}[i,j])^2 \tag{6.9}$$

Now assuming $\hat{w}_m$ as the noise we can find the area under the autocorrelation curve of the noise using equation (6.2) for each σ_i. For the values higher than the real noise variance the remaining part is a real noise, so we expect a constant area under autocorrelation curve for the values higher than the real noise variance which can be used as a criteria to estimate the noise variance.

In Figure 6.5 you can see the area under the autocorrelation curve, it can be seen after a particular variance the area under the autocorrelation curve remains constant. The variance, in which the area remains constant, is very close to the real variance of the noise. To find this value we have used the difference between the two adjacent points, as it can be seen in figure 6.6 it is zero from big variances until a variance which is very close to the real noise variance. Thus, we can select this variance as an estimation for real noise variance. Considering:

$$D_m = P_m(i+1) - P_m(i) \tag{6.10}$$

We should find the first point after the maximum of D_{A_m} which is close to zero sufficiently.

$$m_{max} = \arg_m \max D_m \tag{6.11}$$

$$m^* = \min_{m > m_{max}} \{m : D_m < 10^{-2} D_{m_{max}}\} \tag{6.12}$$

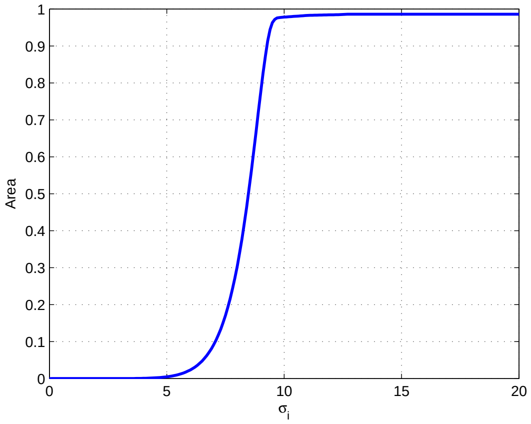


Figure 6.5: Area under the curve of the autocorrelation of the estimated noises with different variances (σ_i) when Real variance is 10.

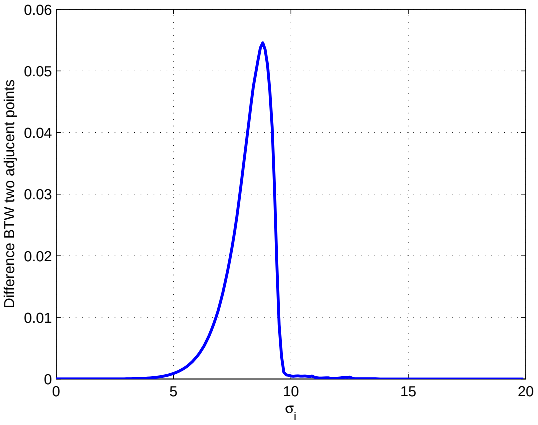


Figure 6.6: Difference between Area under curve for Adjacent points in Figure 6.5, as it can be seen it is close to zero from 10 which is the value of the real noise variance.

and the RAP stddev of the noise stddev is

$$\hat{\sigma}(RAP) = \sigma_m^*. \tag{6.13}$$

This proposed algorithm is tested with other wavelet shrinkage methods such as SureShrink and VisuShrink as the thresholding method used to find the best estimation of the real noise variance, but they are not useful in this case.

6.2 Simulation Results

The optimized BayesShrink method using the proposed noise variance estimation method is used to denoise the images which are shown in Figure 5.2. The estimated variances from *MAD* and the proposed method and their corresponding *MSE*s are given in the tables 6.2 and 6.3. In figure (6.7) you can see a noisy image and its denoised image using BayesShrink and the proposed method.

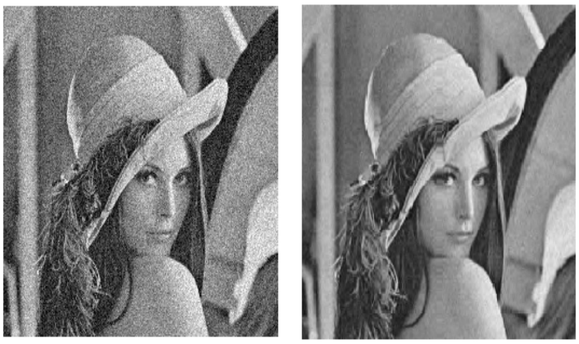


Figure 6.7: Noisy Lena image with $\sigma_w^2 = 10$ and denoised image with BayesShrink using proposed method for Noise Variance estimation.

As it can be seen the results of the *MAD* method are very close to the estimations made by the proposed method. So we can make a hybrid method which is very fast and accurate, a first estimation is made by MAD then an interval from a little less and a little more than

Table 6.2: Comparison of the estimated noise variances of the test images calculated by MAD and proposed method

	Baboon			*Barbara*			*Lena*			*Camera Man*		
True Variance (σ)	5	10	15	5	10	15	5	10	15	5	10	15
RAP ($\widehat{\sigma}$)	5.1	9.9	14.9	4.8	10.3	15.2	5.1	10.4	14.7	4.6	9.8	14.9
MAD ($\widehat{\sigma}$)	7.4	18.9	27	7.2	12.04	18	5.7	9.4	14.8	6.1	12.7	16.7

Table 6.3: Comparison of the MSE values of *MAD* and the optimized BayesShrink using the proposed method for Variance estimation.

	Signal to Noise Ratio (SNR)			
Estimator	10	1	0.5	0.1
Baboon				
MAD	31.43	78.81	88.32	85.75
RAP	5.44	37.48	39.87	41.04
Barbara				
MAD	5.75	32.27	37.14	40.46
RAP	5.2	29.75	33.27	35.64
Lena				
MAD	5.03	28.1	32.9	33.6
RAP	5.03	26.8	29.7	31.57
Camera Man				
MAD	4.47	25.44	25.84	26.9
RAP	3.97	23.2	25.39	27.4

this value is checked using the proposed method to find the best noise variance estimation.

6.3 Concluding Remarks

In this chapter we proposed a new method for noise variance estimation to improve the BayesShrink. The proposed method uses the difference between the autocorrelation of the noisy image and the pure noise to estimate the noise variance. Since MAD is the standard noise variance estimation method in denoising applications, to test the performance of the proposed method, it was compared with MAD.

It was shown that RAP has closer estimations to the true noise variance and it has lower MSEs.

Although RAP noise variance has more accurate estimations in comparison with MAD-based methods, it is more complex and is not useful for for online processing. To reduce the amount of complexity and calculation cost a hybrid method was proposed that uses the estimated variance by MAD as a first estimation and an interval around this first estimation is used in RAP.

Chapter 7

Bayesian Estimation of General Gaussian Distributed (GGD) Signals in Presence of Additive Noise

In this chapter we provide a soft thresholding method. The proposed threshold value and soft thresholding method is a linear estimation of the Bayesian estimation of General Gaussian Distributed (GGD) signals in presence of additive noise. GGD is used in different applications to model the natural signals [11, 12, 13, 14]. We estimate the noiseless coefficients using the Bayes estimator applied on observed noisy coefficients. A threshold value is found that makes soft threshold function the best linear estimation of the Bayes estimator. In addition, we study the Bayes estimators for GGD data to provide the theoretical justification for BayesShrink.

7.1 Bayesian Estimate of GGD Signals

The mean of the posterior distribution provides an unbiased least-squares estimate of $\bar{\theta}$, given measurement θ [61].

$$\begin{aligned}\hat{\theta} &= \int \bar{\theta} f_{\bar{\theta}|\theta}(\bar{\theta}|\theta)\ \mathrm{d}\bar{\theta} \qquad (7.1)\\ &= \frac{\int \bar{\theta} f_{\theta|\bar{\theta}}(\theta|\bar{\theta}) f_{\bar{\theta}}(\bar{\theta})\ \mathrm{d}\bar{\theta}}{\int f_{\theta|\bar{\theta}}(\theta|\bar{\theta}) f_{\bar{\theta}}(\bar{\theta})\ \mathrm{d}\bar{\theta}}\end{aligned}$$

Since noise is additive Gaussian noise, from (5.2) we have, $f_{\theta|\bar{\theta}}(\theta|\bar{\theta}) = f_v(\theta-\bar{\theta}) = \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\frac{(\bar{\theta}-\theta)^2}{2\sigma_w^2}}$. Thus (7.1) can be written as,

$$\hat{\theta} = \frac{\int \bar{\theta} f_v(\theta - \bar{\theta}) f_{\bar{\theta}}(\bar{\theta}) \, \mathrm{d}\bar{\theta}}{\int f_v(\theta - \bar{\theta}) f_{\bar{\theta}}(\bar{\theta}) \, \mathrm{d}\bar{\theta}} \tag{7.2}$$

Equation (7.2) can not be solved analytically for all values of β. The only β values, that we can find the close form solution for them are 1 and 2. When β is one, GGD is the same as Laplacian distribution and when β is two, GGD describes the Gaussian distribution. For other values of β, we should use numerical calculations to find the Bayes estimation.

When β is one or data distribution is Gaussian, the Bayes estimator is as follows.

$$\hat{\theta} = \frac{\sigma_{\bar{y}}^2}{\sigma_{\bar{y}}^2 + \sigma_w^2}\theta \tag{7.3}$$

Details of calculations are shown in Appendix A.

When the signal follows Laplacian distribution, β is one, the denoised coefficients can be calculated by:

$$\hat{\theta} = \frac{e^{\frac{\sqrt{2}\theta}{\sigma_{\bar{y}}}}(2\sigma_w^2 + \sqrt{2}\sigma_{\bar{y}}\theta)Q(\frac{\sqrt{2}\sigma_w}{\sigma_{\bar{y}}} + \frac{\theta}{\sigma_w}) - e^{-\frac{\sqrt{2}\theta}{\sigma_{\bar{y}}}}(2\sigma_w^2 - \sqrt{2}\sigma_{\bar{y}}\theta)Q(\frac{\sqrt{2}\sigma_w}{\sigma_{\bar{y}}} - \frac{\theta}{\sigma_w})}{\sqrt{2}\sigma_{\bar{y}}[e^{\frac{\sqrt{2}\theta}{\sigma_{\bar{y}}}}Q(\frac{\sqrt{2}\sigma_w}{\sigma_{\bar{y}}} + \frac{\theta}{\sigma_w}) + e^{\frac{-\sqrt{2}\theta}{\sigma_{\bar{y}}}}Q(\frac{\sqrt{2}\sigma_w}{\sigma_{\bar{y}}} - \frac{\theta}{\sigma_w})]} \tag{7.4}$$

In Figure 7.1, $\hat{\theta}$ is shown for different values of β while $\sigma_{\bar{y}}$ and σ_w are one.

As it can be seen it is very close to the form of soft thresholding. If we decrease the value of β, for example to 0.1, the Bayesian estimator will be very close to hard thresholding. This case is shown in Figure 7.2 when $\sigma_{\bar{y}}$ and σ_w are one.

To test the effect of variances on Bayesian estimation, $\sigma_{\bar{y}}$ is kept equal to one and σ_w is changed to 3. Figure 7.3 shows the effect of noise variance changes. As it can be seen, when σ_w is higher than $\sigma_{\bar{y}}$, the estimator is more similar to hard threshold transfer function.

Regarding to the shown result, for sharp distribution (when β is small) and when SNR is small, hard thresholding is closer to the optimum estimator. On the other hand, for the signal with large SNR and β between 0.5 and 1, which contains a wide class of natural images [10], soft threshold is a better estimation than hard threshold.

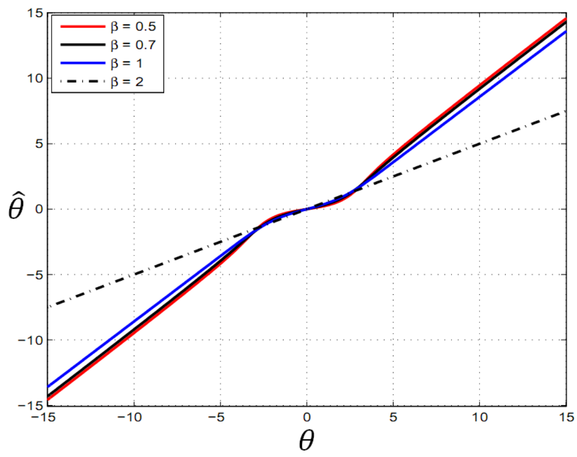


Figure 7.1: Bayesian estimation curves with different βs when σ_w and $\sigma_{\bar{y}}$ are one.

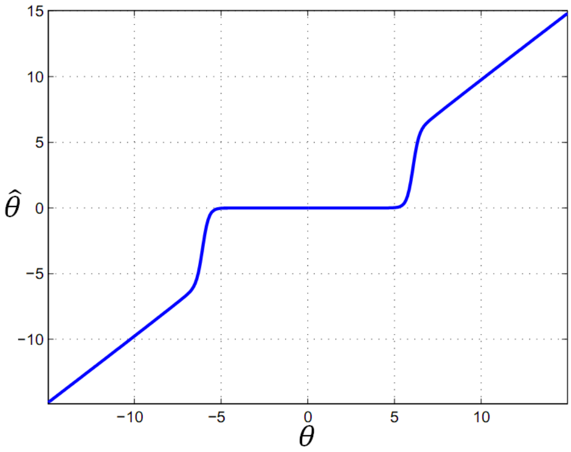


Figure 7.2: Bayesian estimation curve when β is 0.1 and σ_w and $\sigma_{\bar{y}}$ is one.

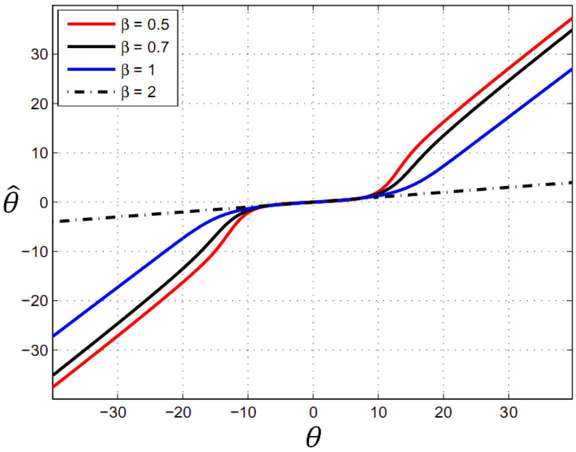


Figure 7.3: Bayesian estimation curve when β is 0.1 and $\sigma_w = 3$ and $\sigma_{\bar{y}} = 1$. When noise variance is higher than the signal's variance, the Bayesian curve converges to hard threshold.

As it can be seen in Figure 7.1, for the large coefficients, slope of the curves is constant and it is one. Using this property we can find the threshold value. Threshold value can be thought as a point in which the curve slope suddenly changes. There is a maximum in the derivative of the curve at this point. Figure 7.4 shows the Bayesian estimator transfer functions with different βs and the derivative of the curves.

We use the maximum value of the derivative of the Bayes estimation as the threshold value. To find the relationship between β, $\sigma_{\bar{y}}$ and σ_w with the threshold value, we have drawn three curves by fixing two parameters and changing the third parameter and plotting the threshold values in each case. Then we use least square curve fitting to find the relationship between the parameters.

7.1.1 Least Square Curve Fitting for Bayesian Estimator

To find the relationship between the parameters, the threshold values are drawn and the best-fit curve is found by least square method. The method of least squares assumes that the best-fit curve of a given type is the curve that has the minimal sum of the deviations squared (least square error) from a given set of data. In the other words, if $\{x_i, i = 1, ..., n\}$ be the independent variable and $\{y_i, i = 1, ..., n\}$ be the dependent variable, Least Square

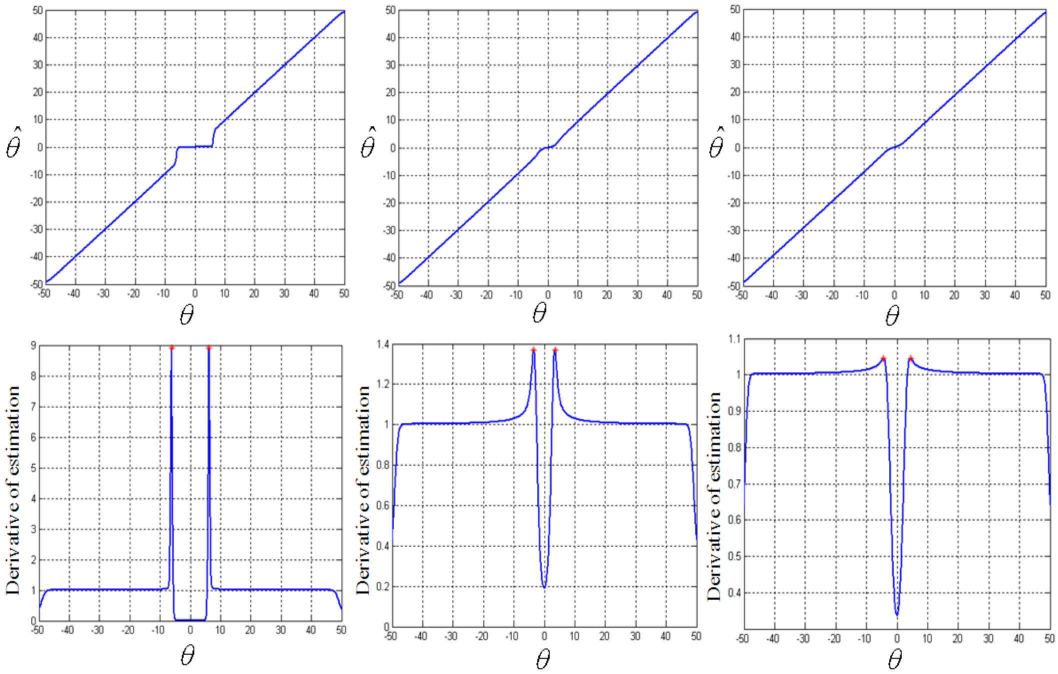


Figure 7.4: Left: $\beta = 0.1$, Middle: $\beta = 0.5$ and Right $\beta = 0.9$. Top figure are the transfer functions and the bottom figures are their derivative.

tries to minimize Π which is defined as:

$$\Pi = \sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} [y_i - f(x_i)]^2 \tag{7.5}$$

in which $f(x)$ is the fitting curve.

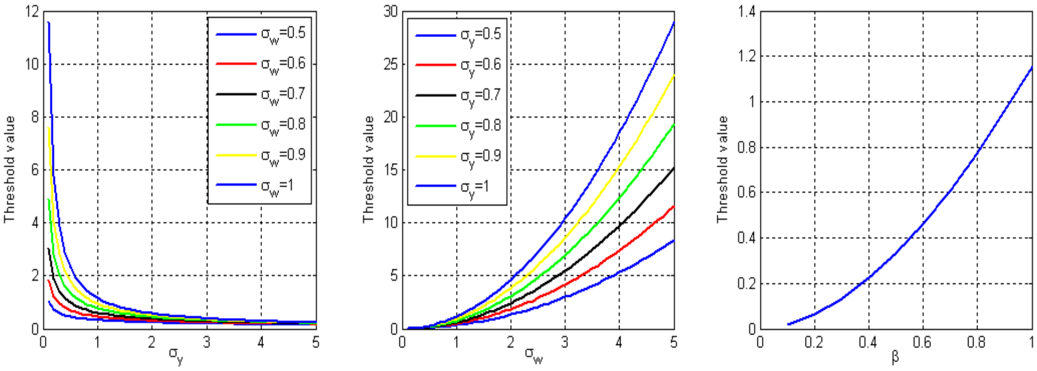


Figure 7.5: Left: Threshold value vs. σ_w while β and $\sigma_{\bar{y}}$ are constant, Middle: Threshold value vs. $\sigma_{\bar{y}}$ while β and σ_w are constant, and Right: Threshold value vs. β while $\sigma_{\bar{y}}$ and σ_w are constant.

Using figure 7.5, it can be seen that $T \propto \frac{1}{\sigma_{\bar{y}}^{\beta}}$, $T \propto \sigma_w^2$, and $T \propto \beta^{1.8}$ and finally the best match is:

$$T = \frac{\sqrt{2}\beta^{1.8}\sigma_w^2}{\sigma_{\bar{y}}^{\beta}} \tag{7.6}$$

When β is one, the distribution is Laplacian and the threshold value is $T = \frac{\sqrt{2}\sigma_w^2}{\sigma_{\bar{y}}}$ which is different with BayesShrink in a $\sqrt{2}$. This method enables us to generalize the BayesShrink denoising by making the threshold value related to β.

Since the provided threshold is a function of β, it must be estimated. In the next section the used shape parameter estimation method is introduced.

7.1.2 GGD Parameter Estimation

There are three methods available for β estimation in Generalized Gaussian Distributed signals: 1) Moment Matching Estimatiors (MME) which uses moments of GGD signals;

2) Entropy matching Estimator (EME) that relies on matching the entropy of the GGD modeled distribution with that of a set of empirical data; and 3) Maximum Likelihood Estimator (MLE) that finds the shape parameter which maximizes the likelihood [62].

MME is an accurate and a simple method that is usefull when β is between 0.18 and 2 [62]. This interval covers most of the natural signals. Thus, we use MME for GGD parameter estimation.

To estimate β, we use the second and forth moments. Kurtosis of GGD has the following form:

$$\kappa_y = \frac{1}{\sigma_y^4}(6\sigma_w^2\sigma_y^2 - 3\sigma_w^4 + (\sigma_y^2 - \sigma_w^2)^2\frac{\Gamma(\frac{1}{\beta})\Gamma(\frac{5}{\beta})}{\Gamma^2(\frac{3}{\beta})}) \tag{7.7}$$

Using MAD estimation, (2.17), we can estimate the noise variance. On the other hand, σ_y and κ_y can be estimated from the observed noisy data $\{y_i, i = 1, ..., N\}$. Kurtosis is defined as:

$$\kappa_y = \frac{\sum_{i=1}^{N}(y_i - \bar{Y})^4}{(N-1)\sigma_y^4} \tag{7.8}$$

Where $\bar{Y}$ is mean value of y_is. Using these equations β can be found.

7.2 BayesShrink and Generalized BayesShrink

BayesShrink is one of the well known methods in wavelet shrinkage and denoising methods. It works based on the assumption that the wavelet coefficients of the image subbands follow General Gaussian distribution in which the shape parameter, β, is often between 0.5 and 1. This distribution is highly accepted in image processing communities for wavelet coefficients of image subbands [11, 12, 13, 14].

BayesShrink has a very good performance for a wide class of images but it does not have an analytical proof and the threshold value is numerically optimized for the GGD signals corrupted with *i.i.d* Gaussian noise.

To check how BayesShrink works, we compare the BayesShrink transfer function with the optimum transfer function found by Bayesian estimator. Figure 7.6 shows this difference. In this figure variance of signal ($\sigma_{\bar{y}}$) and noise variance (σ_w) are fixed and β is changed.

As it can be seen, the estimation function is changed with changing the shape parameter. However, the threshold value in BayesShrink, $\frac{\sigma_w^2}{\sigma_{\bar{y}}}$, is independent from value of β. BayesShrink is optimized to minimize the Bayesian Risk in average for different β values. This independency decreases the complexity of method since there is no need to estimate β.

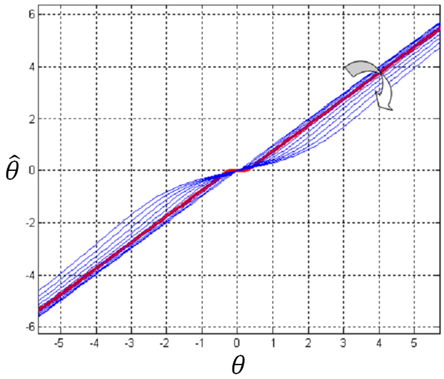


Figure 7.6: Effect of increasing β on Bayesian curve. β increases in direction of arrow.

Our proposed method is a generalized form of BayesShrink. Thus, we can name it Generalized BayesShrink. In simulation results we show that Generalized BayesShrink has better results in comparison with BayesShrink in sense of MSE. However, our proposed method is related to shape parameter of GGD distribution, β. Consequently, we need to estimate β first which increases the possibility of error and complexity.

7.3 Simulation Results

Since our proposed threshold value depends to shape parameter, β, it should be estimated before denoising. Like other estimations β estimation has error on itself. To have better comparison with other methods, we do two different tests to compare our proposed threshold with BayesShrink. In the first one we test the threshold values with analytically made wavelet coefficients following GGD with known shape parameter which does not need β estimation. In the next test, we apply the soft threshold method with the proposed threshold value and BayesShrink's threshold on natural and medical images.

Table 7.1: PSNR of denoised data with BayesShrink and the proposed threshold applied on analytical data.

SNR	5	10	15	20	25	30
$\beta = 0.1$						
BayesShrink	43.53	52.13	56.83	62.11	62.68	70.39
Proposed method	**47.23**	**54.13**	**57.89**	**62.43**	**63.23**	**70.76**
$\beta = 0.3$						
BayesShrink	43.94	43.22	49.33	55.14	58.71	68.60
Proposed method	**44.74**	**45.23**	**51.62**	**56.22**	**59.10**	**68.69**
$\beta = 0.5$						
BayesShrink	41.23	43.57	46.40	49.20	55.60	58.06
Proposed method	**42.18**	**45.15**	**47.42**	**49.22**	**55.61**	**58.07**
$\beta = 0.8$						
BayesShrink	33.05	35.16	46.42	49.40	53.76	61.35
Proposed method	**34.93**	**35.64**	**46.57**	**49.43**	**53.77**	**61.35**
$\beta = 1$						
BayesShrink	34.89	37.15	44.09	46.01	51.53	59.47
Proposed method	**34.99**	**37.17**	**44.11**	**46.02**	51.53	59.47

In the first test we use analytical data with β equal to 0.1, 0.3, 0.5, 0.8 and 1. The results are shown in Table 7.1. In this table data length, which is used instead of wavelet coefficients, is 1000 and the test is done with 100 runs. As it can be seen in Figure 7.6, when β is small BayesShrink is vary different from the Bayes estimation. This difference can be seen in Table 7.1 as well, since the improvement in our method is considerable when β is small. The numbers in the table are PSNR of the denoised images which is defined in (2.8).

In the second test, we use the images shown in Figure 7.10. Using these images, we have the results given in Table 7.2.

As it can be seen in Table 7.2 when input SNR is low BayesShrink is working better. If we compare these results with results of Table 7.1 in which in all the cases our proposed method works at least as well as BayesShrink, we can conclude that the larger error in low SNRs is because of the errors in parameter estimation.

Performance of BayesShrink and the proposed method is compared in Figures **??**,**??**, and **??**. As it can be seen in these figures the proposed method preserved the edges better than BayesShrink.

Figure 7.7: Test images, Top from left to right: CameraMan, Lena and Mandrill. Bottom from left to right: Peppers, Einstein and Coco.

Table 7.2: PSNR of denoised images with BayesShrink and the proposed threshold.

SNR	5	10	15	20	25	30
CameraMan						
BayesShrink	**23.00**	24.20	24.97	25.37	25.54	25.606
Proposed method	22.63	**24.42**	**25.10**	**25.42**	**25.56**	**25.607**
Lena						
BayesShrink	**27.96**	30.52	32.66	34.16	34.98	35.324
Proposed method	27.60	**30.68**	**32.89**	**34.25**	**35.10**	**35.327**
Mandrill						
BayesShrink	**23.28**	**24.77**	25.54	25.854	25.980	26.017
Proposed method	22.89	24.69	25.54	25.854	25.980	26.017
Peppers						
BayesShrink	**24.45**	27.20	29.18	30.56	31.12	31.322
Proposed method	24.40	**27.52**	**29.36**	**30.61**	**31.13**	**31.323**
Einstein						
BayesShrink	**30.25**	33.12	35.18	36.69	37.452	37.77
Proposed method	30.24	**33.23**	**35.26**	**36.71**	**37.456**	**37.78**
Coco						
BayesShrink	28.44	31.41	34.12	36.25	37.52	38.130
Proposed method	**28.63**	**31.80**	**34.45**	**36.39**	**37.56**	**38.136**

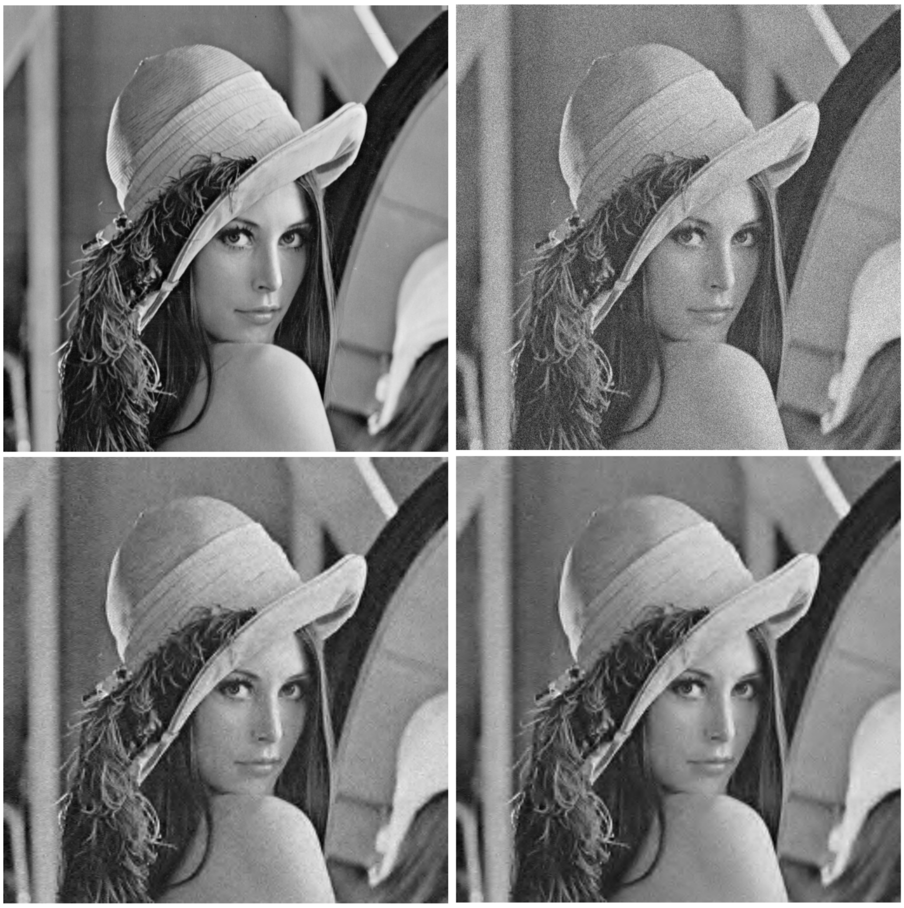


Figure 7.8: Comparison of Lena image denoised with BayesShrink and the proposed method. Top left: noiseless image, Top right: noisy image with SNR=10, Bottom Left: denoised with BayesShrink, and Bottom right: denoised with the proposed method.

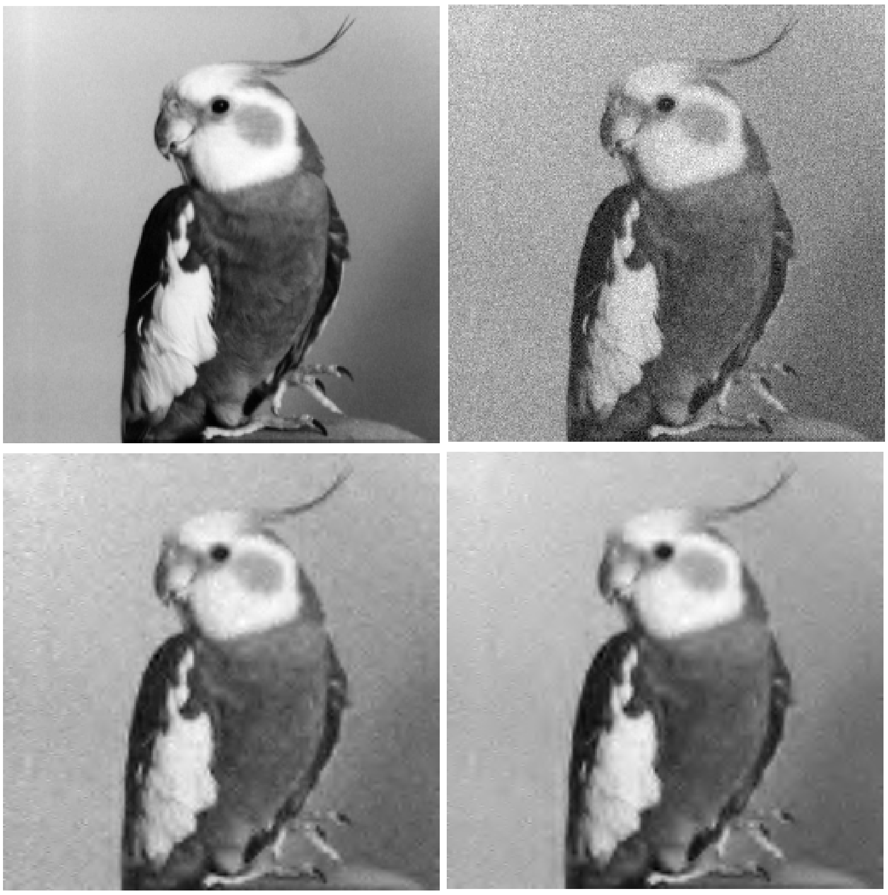


Figure 7.9: Comparison of Coco image denoised with BayesShrink and the proposed method. Top left: noiseless image, Top right: noisy image with SNR=10, Bottom Left: denoised with BayesShrink, and Bottom right: denoised with the proposed method.

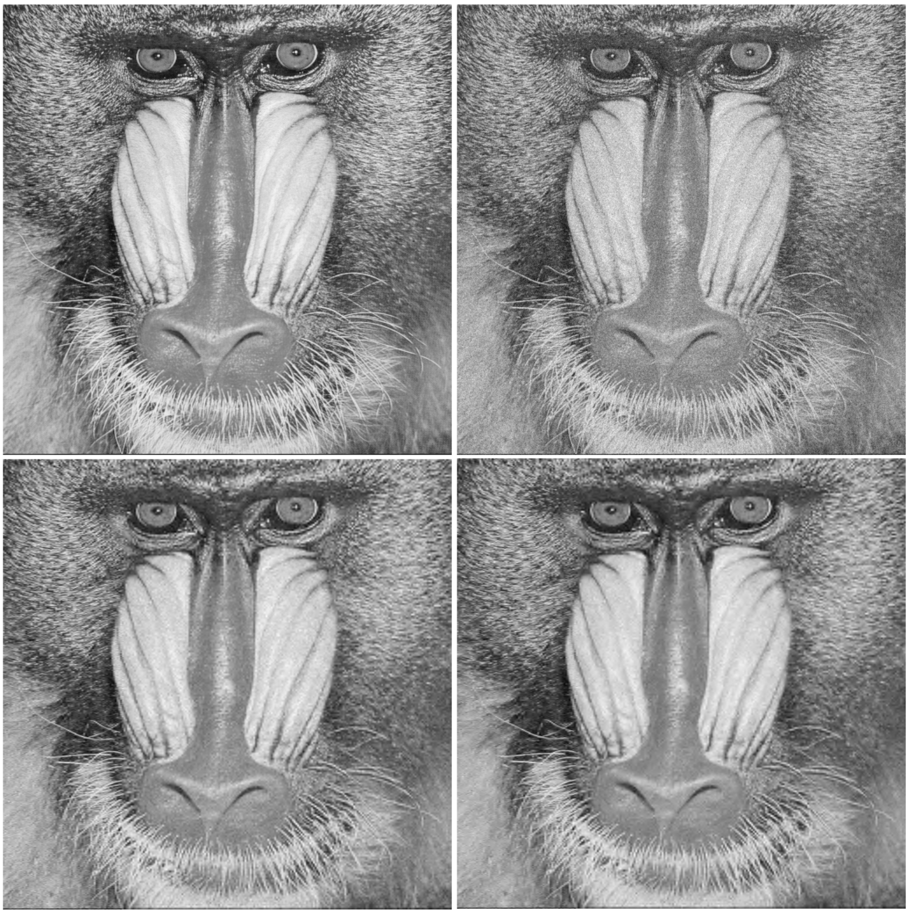


Figure 7.10: Comparison of Mandrill image denoised with BayesShrink and the proposed method. Top left: noiseless image, Top right: noisy image with SNR=10, Bottom Left: denoised with BayesShrink, and Bottom right: denoised with the proposed method.

7.3.1 Simulation Results for Medical Images

As the last test, we compare our proposed method with BayesShrink applied on Medical Images. We use miniMIAS [63] Mammography images shown in Figure 7.11 and Head CT scan images shown in Figure 7.12.

The images size is 512×512, the test is ran for 50 times and Daubechies wavelet with 4 taps is used. The results of miniMIAS are shown in Table 7.3. It can be seen that our proposed method is working better than BayesShrink.

Results of CT scan images are shown in Table 7.4. This table confirms the strength of the proposed method in comparison with BayesShrink. The only cases that BayesShrink is working better is low SNR cases where SNR is five.

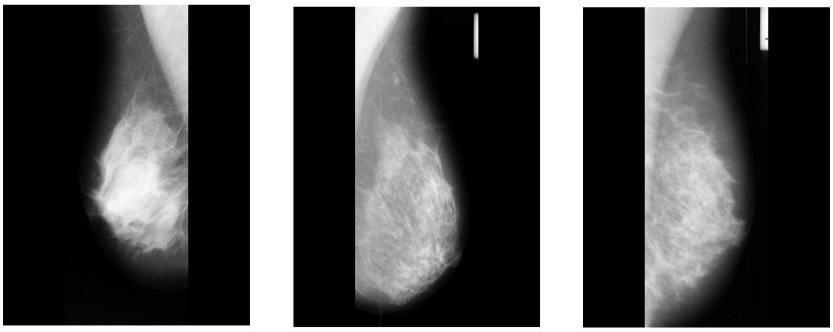


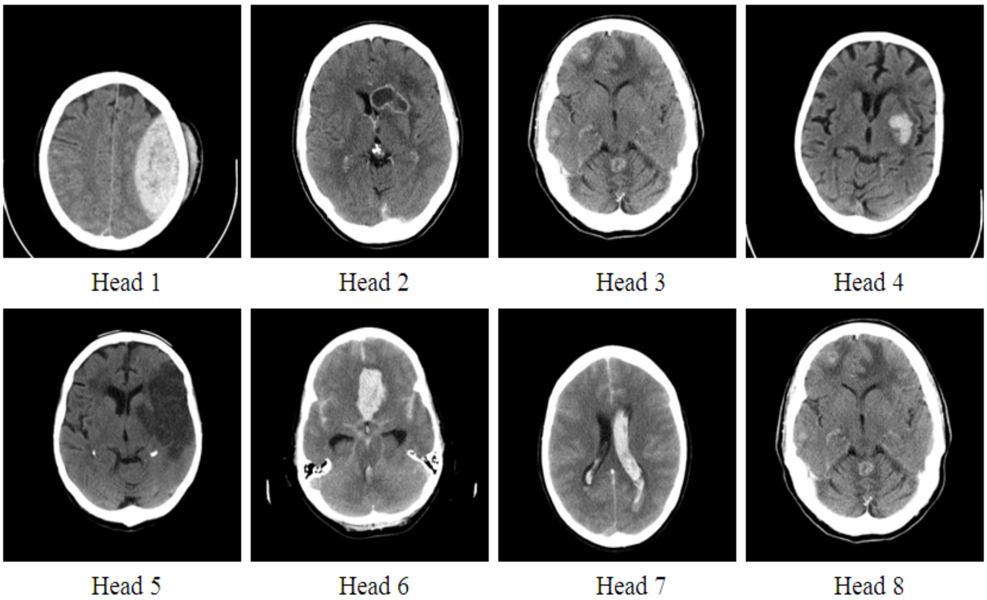
Figure 7.11: miniMIAS mammography images used in our test, Right: mdb001, Middle: mdb100, Left: mdb200.

7.4 Concluding Remarks

A new soft thresholding method, Generalized BayesShrink, was proposed based on Bayesian estimation of General Gaussian Distributed signals. The proposed method generalized the BayesShrink. It was shown that the proposed method outperforms BayesShrink in most cases. However, it is dependent to the shape parameter of GGD, β. Because of this dependency, it is more complex than BayesShrink. It was shown that when β is low or

Table 7.3: PSNR of denoised mammogram with BayesShrink and the proposed threshold applied on medical test images shown in Figure 7.11.

SNR	5	10	15	20	25	30
mdb001						
BayesShrink	27.95	28.66	33.82	35.78	37.73	38.50
Proposed method	**28.53**	**29.22**	**33.99**	**35.82**	**37.78**	38.50
mdb100						
BayesShrink	25.51	29.75	32.70	35.37	37.26	38.44
Proposed method	**25.99**	**30.68**	**33.27**	**35.64**	**37.34**	**38.46**
mdb200						
BayesShrink	26.71	29.85	31.90	34.60	36.23	37.40
Proposed method	**26.95**	**30.93**	**32.72**	**34.99**	**36.35**	**37.43**



Head 1 Head 2 Head 3 Head 4

Head 5 Head 6 Head 7 Head 8

Figure 7.12: CT scan test images.

Table 7.4: PSNR of denoised CT images with BayesShrink and the proposed threshold applied on medical test images shown in Figure 7.12.

SNR	5	10	15	20	25	30
Head 1						
BayesShrink	**23.65**	26.57	28.34	29.80	30.75	31.15
Proposed method	23.35	**27.09**	**28.90**	**30.05**	**30.83**	**31.63**
Head 2						
BayesShrink	**24.12**	26.47	28.63	30.06	30.85	31.21
Proposed method	23.80	**26.65**	**29.01**	**30.26**	**30.91**	**31.23**
Head 3						
BayesShrink	**23.49**	26.21	28.37	30.12	31.14	31.50
Proposed method	23.19	**26.49**	**28.81**	**30.38**	**31.22**	**31.61**
Head 4						
BayesShrink	24.31	26.98	28.83	30.25	31.06	31.42
Proposed method	24.31	**27.47**	**29.27**	**30.44**	**31.12**	**31.44**
Head 5						
BayesShrink	25.89	28.13	30.07	31.63	32.49	32.87
Proposed method	**26.05**	**28.73**	**30.51**	**31.80**	**32.54**	**32.89**
Head 6						
BayesShrink	**24.32**	26.67	28.79	30.31	31.20	31.61
Proposed method	24.07	**26.98**	**29.20**	**30.52**	**31.27**	**31.63**
Head 7						
BayesShrink	**24.91**	27.50	28.97	30.87	31.73	32.21
Proposed method	24.86	**27.92**	**29.43**	**31.08**	**31.79**	**32.22**
Head 8						
BayesShrink	**23.12**	26.04	28.39	30.11	31.12	31.58
Proposed method	22.73	**26.17**	**28.83**	**30.36**	**31.20**	**31.60**

noise variance is higher than signal variance, hard thresholding gives a better estimation of Bayesian estimator. However, soft thresholding estimates the Bayesian estimator more accurately when SNR is high and β is between 0.5 and 1 which contains a big class of natural images. From the numerically calculated Bayes curves, it could be seen that BayesShrink is not optimal in most cases but it is in average a very good estimation of the Bayesian estimator.

Chapter 8

Conclusions

In this thesis we studied four problems in data denoising: 1) MSE estimation, 2)Analog data denoising, 3) Digital Data denoising, and 4) noise variance estimation.

In Chapter 3 MSE estimation was studied in denoising methods which are using soft threshold. The proposed method, estimated MSE based on the threshold value used in soft threshold and did not need any information other than the noisy input data.

Three analog denoising methods were provided in the thesis. A denoising method was proposed in Chapter 3 based on the MSE EstiMation (MSEEM). The MSE in MSEEM was very close to the minimum possible MSE of soft thresholding. Another analog data denoising method, NIDe, was proposed in Chapter 4. NIDe used a noise signature to validate if the coefficients have noise like behavior or not and denoised the signal by removing the noise like coefficients. This method used a masking denoising method which makes it different from other existing thresholding and denoising methods. Generalized BayesShrink method is proposed in Chapter 7.

Chapter 5 studied the denoising problem in digital domain. This chapter had two main contributions. In the first part, Two Stage Quantization (TSQ) was proposed. TSQ, simultaneously denoised and quantized the noisy input data. This procedure was done in two stages, in the first stage a multistage denoising method was proposed based on MNDL which handled the noise dominant part of the signal. In the second stage, the remaining part of the input signal which is noiseless dominant was handled in which one of the existing quantization methods was used to quantize this remained part. This method is a new method that generalized the idea of dead zone estimation to a multi-level noise removal. The simulation results showed that this method outperfoms the separate denoising and quantization methods. An example of this method was shown for hyperspectral image denoising and compression. In the second part, Multiple Stage Denoiser and Quantizer (MSDQ) was proposed. The input signal of MSDQ was a noisy quantized/digital signal. MSDQ found the best reconstruction levels to make the reconstructed signal as close as possible to the noiseless signal. When the input SNR was high, the reconstruction levels were very close to the original levels that were used to quantize the signal. On the other hand, when the input SNR was low, the found levels could be different with the original levels in number and value. It was shown by the simulations that, in MSE sense, the reconstructed signal by MSDQ was closer to the noiseless signal in comparison with the reconstructed signal with original levels.

One of the first steps in almost all the denoising methods is to estimate the noise variance. In chapter 6 we studied the noise variance estimation problem. The standard method for noise variance estimation is MAD-based method. In this chapter we showed that for high frequency wavelet coefficients, this method is not efficient. A novel noise variance estimation was proposed based on Residual Autocorrelation Power (RAP) for BayesShrink. The strength of RAP method was shown in simulation results. RAP outperformed the MAD-based noise variance estimation in most cases and improved BayesShrink approach.

In Chapter 7 we studied Bayes estimators for General Gaussian Distributed (GGD) data and provided the theoretical justification for BayesShrink. This study enabled us to generalize the BayesShrink threshold to Generalized BayesShrink which outperformed the BayesShrink itself. Among the proposed methods for analog data denoising, NIDe had the best perfomance in colored noise cases. Performance of NIDe and Generalized BayesShrink were very close in white cases and both of them were better than MSEEM. MSE in NIDe was more than the MSE of Generalized BayesShrink in non-sparse cases while Generalized BayesShrink had a consistent performance in both sparse and non-sparse cases.

Appendix A

Details of Bayesian Estimation when Shape Parameter Is Two

The mean of the posterior distribution provides an unbiased least-squares estimate of $\bar{\theta}$ is calculated from (7.1). Since:

$$\frac{f_v(\theta-\bar{\theta})f_{\bar{\theta}}(\bar{\theta})}{f_\theta(\theta)} = \frac{\frac{1}{\sqrt{2\pi}\sigma_w}e^{\frac{-(\theta-\bar{\theta})^2}{2\sigma_w^2}} \times \frac{1}{\sqrt{2\pi}\sigma_{\bar{y}}}e^{\frac{-(\bar{\theta})^2}{2\sigma_{\bar{y}}^2}}}{\frac{1}{\sqrt{2\pi}\sigma_y}e^{\frac{-(\theta)^2}{2\sigma_y^2}}} \tag{A.1}$$

where $\sigma_y^2 = \sigma_{\bar{y}}^2 + \sigma_w^2$. Now, we simplify (A.1) to:

$$\frac{f_v(\theta-\bar{\theta})f_{\bar{\theta}}(\bar{\theta})}{f_\theta(\theta)} = \frac{1}{\sqrt{2\pi}\frac{\sigma_w\sigma_{\bar{\theta}}}{\sigma_y}}e^{\frac{-\sigma_{\bar{y}}^2\sigma_y^2(\bar{\theta}-\theta)^2-\sigma_w^2\sigma_y^2\bar{\theta}^2+\theta^2\sigma_{\bar{y}}^2\sigma_w^2}{2\sigma_w^2\sigma_{\bar{y}}^2\sigma_y^2}} \tag{A.2}$$

$$\frac{1}{\sqrt{2\pi}\frac{\sigma_w\sigma_{\bar{\theta}}}{\sigma_y}}e^{\frac{\theta^2(\sigma_{\bar{y}}^2\sigma_w^2-\sigma_{\bar{y}}^2\sigma_y^2)-\bar{\theta}^2(\sigma_w^2\sigma_y^2+\sigma_{\bar{y}}^2\sigma_y^2)+2\theta\bar{\theta}\sigma_{\bar{y}}^2\sigma_y^2}{2\sigma_w^2\sigma_{\bar{y}}^2\sigma_y^2}}$$

On the other hand:

$$(\sigma_{\bar{y}}^2\sigma_w^2 - \sigma_{\bar{y}}^2\sigma_y^2) = -\sigma_{\bar{y}}^4 \tag{A.3}$$

and,

$$(\sigma_w^2\sigma_y^2 + \sigma_{\bar{y}}^2\sigma_y^2) = \sigma_y^4 \tag{A.4}$$

By substituting (A.3) and (A.4) in (A.2) we have:

$$\frac{f_v(\theta-\bar{\theta})f_{\bar{\theta}}(\bar{\theta})}{f_\theta(\theta)} = \frac{1}{\sqrt{2\pi}\frac{\sigma_w\sigma_{\bar{\theta}}}{\sigma_y}}e^{\frac{-(\theta\sigma_{\bar{y}}^2-th\bar{e}ta\sigma_y^2)^2}{2\sigma_w^2\sigma_{\bar{y}}^2\sigma_y^2}} \tag{A.5}$$

$$= \frac{1}{\sqrt{2\pi}\frac{\sigma_w \sigma_{\bar{\theta}}}{\sigma_y}} e^{\frac{-(\theta\frac{\sigma_{\bar{y}}^2}{\sigma_y^2} - th\bar{e}ta)^2}{2\sigma_w^2 \sigma_{\bar{y}}^2 \sigma_y^2}}$$

Thus, $\frac{f_v(\theta-\bar{\theta}) f_{\bar{\theta}}(\bar{\theta})}{f_\theta(\theta)}$ follows Gaussian Distribution and its mean is $\theta\frac{\sigma_{\bar{y}}^2}{\sigma_y^2}$ and as the result:

$$\hat{\theta} = \theta \frac{\sigma_{\bar{y}}^2}{\sigma_{\bar{y}}^2 + \sigma_w^2} \tag{A.6}$$

Bibliography

[1] A. V. Oppenheim, and R. W. Schafer, *Discrete-time signal processing*, Prentice-Hall,Engle-wood Cli's, NJ, 1989.

[2] A. V. Oppenheim, A. S. Willsky, and I. T. Young, *Signals and systems*, Prentice-Hall, Engle wood Cliffs, NJ, 1983.

[3] S. S. Haykin, *Adaptive filter theory*, Prentice-Hall, Upper Saddle River, NJ, third edition, 1996.

[4] Y. Washizawa, Y. Yamashita, "Non-linear Wiener filter in reproducing kernel Hilbert space", in *Proc. 18th International Conference on Pattern Recognition, ICPR*, 2006, pp. 967-970.

[5] J. Chen, J. Benesty, Yiteng Huang, and S. Doclo, "New insights into the noise reduction Wiener Filter", *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1218-1234, Jul. 2006.

[6] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaption via wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, 1994.

[7] D. L. Donoho, "De-noising by soft thresholding," *IEEE Trans. Inform. Theory*, vol. 41, no. 3 pp. 613-627, May 1995.

[8] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Assos.*, vol. 90, no. 432, pp. 1200-1224, Dec. 1995.

[9] S. G. Chang, B. Yu, and M. Vetterli, “Spatially adaptive wavelet thresholding with context modeling for image denoising,” *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1522-1531, Sept. 2000.

[10] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive Wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Processing.* vol. 9, no. 9, pp. 1532-1546, Sep. 2000.

[11] R. L. Joshi, V. J. Crump, and T. R. Fisher, “Image subband coding using arithmetic coded trellis coded quantization,” *IEEE Trans. Circuits sys. Video Technol.*, vol. 5, no. 6, pp. 515-523, Dec. 1995.

[12] S. M. LoPresto, K. Ramchandran, and M. T. Orchard, “Image coding based on micture modeling of wavelet coefficients and a fast esimation quantization framework,” in *Proc. Data Compression Conf. Snowbird, UT*, Mar. 1997, pp. 221-230.

[13] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 647-693, July 1989.

[14] E. Simoncelli and E. Adelson, “Noise removal via Bayesian wavelet coring,” in *Proc. IEEE international Conf. Image Processing*, Sept. 1996, vol. 1, pp. 379-382.

[15] S. Beheshti and M.A. Dahleh, “A new information-theoretic approach to signal denoising and best basis selection,” *IEEE Trans. Signal Processing*, vol. 53, no. 10, pp. 3613-3624, Oct. 2005.

[16] N. Nikvand, S. Beheshti, and X. Fernando, “Soft thresholding by noise invalidation,” in *Proc. Queen’s Biennial Symposium on Communications (QBSC)*, June 2008, pp. 235-238.

[17] D. Gabor, “Theory of communication”, *Journal of I.E.E.*, vol. 93, pp. 429-441, 1946.

[18] S. Mallat. *A Wavelet tour of signal processing.* Academic Press, second edition, 1999.

[19] Vetterli M. and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 9, p. 2207-2232, Sept. 1992.

[20] C. S. Burrus, R. A. Gopinath, and H. Guo, *Interodction to wavelets and wavelet transforms, a primer*, Upper Saddle River, NJ (USA): Prentice Hall, 1998.

[21] I. Daubechies, *Ten lectures on wavelet*, Philadelphia: SIAM, 2nd ed, 1992.
CBMS-NSF regional conference series in applied mathematics 61.

[22] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 4, no. 46, pp. 886-902, Apr. 1998.

[23] M. R. Leadbetter, G. Lindgren, and H. Rootzen, *Extremes and related properties of random sequences and processes* Springer Series in Statistics, Springer-Verlag, New York, Heidelberg, Berlin, 1983.

[24] B. Vodivak and P. Muller, "An introduction to wavelets In Bayesian Inference in wavelet-based model," *Springer-Verlag*, 1990.

[25] S. Beheshti, M. Hashemi, X. Zhang, and N. Nikvand, "Noise invalidation denoising," to Appear in *IEEE Trans. Signal Processing*, Jan. 2011.

[26] H. Chipman, E. Kolaczyk, and R. McCulloch, "Adaptive Bayesian wavelet shrinkage," *J. Amer. Statistics Assoc.*, vol. 92, pp. 1413-1421, 1997.

[27] F. Abramovich, T. Sapatinas, and B. W. Silverman, "Wavelet thresholding via a Bayesian approach," *J. Roy Statistica Soc. Ser. B.*, vol. 60, pp.725-749, 1998.

[28] M. Figueriredo and R. Nowak, "Wavelet based image estimation: An empirical bayes approach using jeffreys' noninformationative prior," *IEEE Trans. Image Processing*, vol. 10, no. 9, pp. 1322-1331, Sept. 2001.

[29] F. Abramovich, T. Sapatinas, and B. W. Silverman, “Wavelet thresholding via a Bayesian approach,” *J. Royal Stat. Soc.* vol. 60, no.4, pp. 725749, 1998.

[30] J. Romberg, H. Choi, and R. Baraniuk, “Baesian tree-structured image modeling using wavelet-domain hidden markov models,” *IEEE Trans. Image Processing*, vol. 10, no. 7, pp. 1056-1068, July 2001.

[31] L. Sendur and I. W. Selensic, “Bivariate shrinkage function for wavelet-based denoising exploiting interscale dependency,” *IEEE trans. Signal Processing*, vol. 50, no. 11, pp. 2744-2756, Nov. 2002.

[32] B. Z. Bobrovski and M. Zakai, “A lower bound on the estimation error for certain diffusion problems,” *IEEE Trans. Inform. Theory*, vol. 22, no. 1, pp. 4552, Jan. 1976.

[33] A. J. Weiss and E. Weinstein, “A lower bound on the mean-square error in random parameter estimation,” *IEEE Trans. Inform. Theory*, vol. 31, no. 5, pp. 680-682, Sept. 1985.

[34] E. Weinstein and A. J. Weiss, “A general class of lower bounds in parameter estimation,” *IEEE Trans. Inform. Theory*, vol. 34, no. 2, pp. 338-342, Mar. 1988.

[35] H. Cramer, *Mathematical Methods of Statistics.* Princeton, NJ: Princeton Univ. Press, 1946.

[36] C. R. Rao, “Information and the accuracy attainable in the estimation of statistical parameters,” *Bulletin of the Calcutta Mathematical Society*, vol. 37, pp. 81-89, 1945.

[37] S. Bellini and G. Tartara, “Bounds on error in signal parameter estimation,” *IEEE Trans. Communication*, vol. 22, no. 3, pp. 340342, 1974.

[38] D. Chazan, M. Zakai, and J.Ziv, “Improved lower bounds on signal parameter estimation,” *IEEE Trans. Inform. Theory*, vol. 21, no. 1, pp. 9093, Jan. 1975.

[39] K. L. Bell, Y. Steinberg, Y. Ephraim, and H. L. Van Trees, "Extended ZivZakai lower bound for vector parameter estimation," *IEEE Trans. Inform. Theory*, vol. 43, no. 2, pp. 624637, Mar. 1997.

[40] J. Ziv and M. Zakai, "Some lower bounds on signal parameter estimation," *IEEE Trans. Inform. Theory*, vol. 15, no. 3, pp. 386391, May 1969.

[41] E. Sejdic, C. M. Steele, and T. Chau, "A procedure for denoising of dual-axis swallowing accelerometry signals", *Physiological Measurements*, vol. 31, no. 1, pp. N1-N9, Jan. 2010.

[42] S. Beheshti, A. Fakhrzadeh, and S. Krishnan, "Noiseless codelength in Wavelet denoising," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, Article ID 641842, Feb. 2010.

[43] T. Blu, F. Luisier, "The SURE-LET Approach to Image Denoising", *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778-2786, November 2007.

[44] Y. Jinhua, "Advantages of uniform scalar dead-zone quantization in image coding system," in *Proc. International Conference on Communications, Circuits and Systems*, June 2004, vol.2, pp.805-808.

[45] M. Hashemi, S. Beheshti, and M. Farzam, "Two stage quantization of noisy hyperspectral images," in *Proc. 25th Queens Biennial Symposium on Communications (QBSC)*, 2010, pp. 340-343.

[46] S. Beheshti and M.A. Dahleh, "Noisy data and impulse response estimation," *IEEE Trans. Signal Processing,* vol. 58, no.2, pp. 510-521, Feb. 2010.

[47] A. Said and W.A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp.243-250, Jun. 1996.

[48] A.B. Kiely and M.A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Trans. Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2672-2678, Aug. 2009.

[49] S.P. Lloyd, "Least squared quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129-137, Mar. 1982.

[50] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Informorm. Theory*, vol. 6, no. 1, pp.7-12, Mar. 1960.

[51] , P. A. Chou and B. J. Betts, "When optimal entropy-constrained quantization have only a finite number of codewords," in *Proc. IEEE international Symp. Inform. Theory, Cambridge, MA, USA*, Aug. 1998, pp. 97-102.

[52] J. Han and L. Xu, "A new method for variance estimation of white noise corrupting a signal," *Instrumentation and Measurement Technology Conference, IMTC 2006.*, Apr. 2006, pp. 1447-1451.

[53] B. Jiang,W. Wang,X. Gao, "Polynomial-Based Noise Variance Estimation for MIMO-SCBT Systems," *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 497-500, June 2009.

[54] D. Makovoz, "Noise Variance Estimation In Signal Processing," in *Proc. IEEE International Symposium on Signal Processing and Information Technology*, Aug. 2006, pp. 364-369.

[55] Y. Rui, M. Li, X. Zhang, L. Tang, and S. Feng, "A noise variance optimization method for 2x1-Dimensional Wiener filtered channel estimation," in *Proc. IEEE Wireless Communications and Networking Conference, WCNC 2007*, Mar. 2007, pp. 232-236.

[56] S. Beheshti and M.A. Dahleh, "Noise variance in signal denoising," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2003, Vol. 6, pp. 185-188.

[57] M. Hashemi and S. Beheshti, “Adaptive noise variance estimation in BayesShrink,” *IEEE Signal Processing Letters*, vol. 17, no. 1, pp 12-15, Jan. 2010.

[58] N. N. Kachouie and P. Fieguth, “A combined Bayesshrink wavelet-Ridgelet technique for image denoising,” in *Proc. IEEE International Conference on Multimedia and Expo 2006* , July 2006, pp. 1917-1920.

[59] Z. Hou, T. S. Koh, “Wavelet shrinkage prefiltering for brain tissue segmentation,” in *Proc. 27th Annual International Conference of the Engineering in Medicine and Biology Society*, Jan. 2006, pp. 1604-1606.

[60] N. Nezamoddini-Kachouie and P. Fieguth, “A Gabor based technique for image denoising,” in *Proc. Canadian Conference on Electrical and Computer Engineering*, May 2005, pp. 980-983.

[61] E. P. Simoncelli, Bayesian denoising of visual images in the wavelet domain. In P. Muller and B. Vidakovic, editors, *Bayesian Inference in Wavelet Based Models,* chapter 18, pp. 291308. Springer-Verlag, New York, June 1999. Lecture Notes in Statistics, vol. 141.

[62] C. Gonzalez-Farias, J. A. Dominguez Molina, and R. M. Rodrigez-Dagnino, “Effcieny of the approximated shape parameter estimator in the generalized Gaussian distribution,” *IEEE Trans. Vehicular Tech.*, vol. 58, no. 8, pp. 4214-4223 , Oct. 2009.

[63] A. A. AbuBaker, “Efficient pre-processing of USF and MIAS mammogram images,” *Journal of Computer Science*, vol. 3, no. 2, pp. 67-75, Feb. 2007.