BAYESIAN INFERENCE AND WAVELET METHODS IN IMAGE PROCESSING

by

SHARAD DEEP SILWAL

M.A., Tribhuvan University, Nepal, 2002

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2009

Approved by:

Co-Major Professor
Dr. Diego Maldonado

Approved by:

Co-Major Professor
Dr. Haiyan Wang

Copyright

Abstract

This report addresses some mathematical and statistical techniques of image processing and their computational implementation. Fundamental theories have been presented, applied and illustrated with examples. To make the report as self-contained as possible, key terminologies have been defined and some classical results and theorems are stated, in the most part, without proof. Some algorithms and techniques of image processing have been described and substantiated with experimentation using MATLAB. Several ways of estimating original images from noisy image data and their corresponding risks are discussed. Two image processing concepts selected to illustrate computational implementation are: “Bayes classification” and “Wavelet denoising”. The discussion of the latter involves introducing a specialized area of mathematics, namely, wavelets. A self-contained theory for wavelets is built by first reviewing basic concepts of Fourier Analysis and then introducing Multi-resolution Analysis and wavelets. For a better understanding of Fourier Analysis techniques in image processing, original solutions to some problems in Fourier Analysis have been worked out. Finally, implementation of the above-mentioned concepts are illustrated with examples and MATLAB codes.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am deeply indebted to my advisors Dr. Diego Maldonado and Dr. Haiyan Wang for their continuous advice and support which helped me accomplish this report. I am also thankful to Dr. James Neill for inspiring me with his critical comments to my work.

Dr. Maldonado, my mathematics professor, is a major source of encouragement to do hard work. His motivational push is one of the main factors contributing to this accomplishment. He is also responsible for suggesting me the topic of this report and training me in wavelets.

My statistics professor Dr. Wang always made me see how the statistical and other mathematical ideas in this report are related. Her pertinent suggestions provided invaluable guidance. She helped me greatly in directing a mix of interdisciplinary subject matter into a Masters report in statistics.

Finally, my gratitude goes to Department of Mathematics, KSU and, in particular, to Dr. Louis Pigno, Department Head, and Dr. David Yetter, Graduate Program Chair, for supporting my study outside of the department keeping my academic advancement in view.

Dedication

To my parents

Chapter 1

Introduction

1.1 Images as matrices

A matrix is an array of finite discrete signals. Alternatively, we can consider it as a two dimensional finite discrete signal. A camera measures light intensity with photoreceptors that perform a uniform sampling over a grid that is supposed to be uniform and the output is an array of picture elements, called pixels, usually, in a rectangular grid[8]. Thus, a rectangular black-and-white image is a matrix where each entry represents an intensity of gray at that spot and ranges from 0 to 255, 0 being the darkest spot and 255 being the brightest spot. For example, consider a matrix A of size $12X12$ randomly generated below using the given MATLAB commands. The image of A generated by MATLAB is given in Figure (1.1).

```
>> A=uint8(255*rand(12,12))

A =

   208   244   173   177   181   192   208   234   145   176    27    46
   231   124   193    81   192    65    62    73   120   191   245    67
    32   204   189   242    70   129   237   193     3   115     1    37
   233    36   100     9   173   178    89   192    86    21   198    35
   161   108   167   112   167   227    50    97    41    58   208   222
    25   234    44    97    41   245    64   145   203   233   222   148
    71   202   180   195    30   140   157    19    79    39    22   140
   139   245     8   203   127    35   121    14   135   211   102    37
   244   167    71    48   245    38    90   135    42   137    66   218
   246     9    12   125    87    66   212   199   154   254   204   159
    40   217    25   114   149   214   149   238    67    20   110    89
   248   238   210   165    57    65   140    33   167   113   232   131

>> imshow(A)
```

A color image is a tensor product of three gray images which correspond to the decompositions of the original image into its three primary color components, namely, red, green and blue. These three colors are linearly independent which means none can be obtained by combining the other two. Furthermore, they generate a basis of the entire color space which means any color shade can be obtained by combining the appropriate shades of these three colors. Thus, the first gray image or the matrix in the tensor product corresponds to the red color component of the color image. Its entries, ranging from 0 to 255, represents red intensities of the pixels. Analogously, the other two matrices in the tensor product has

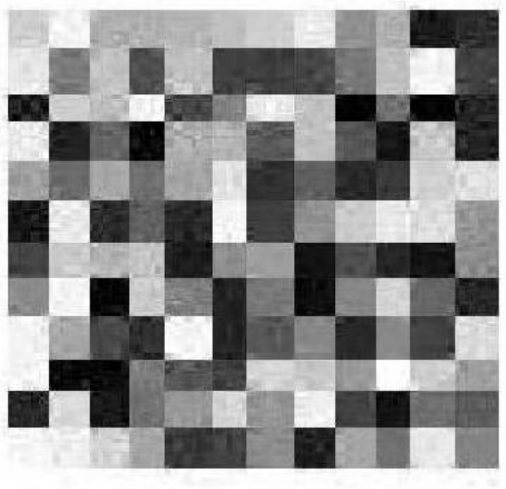

Figure 1.1: *Image of Matrix A.*

as entries green and blue intensities of the pixels. This is illustrated in the color cube of side length 255 given in Figure (1.2).

The purple image in Figure (1.3) (source: http://en.wikipedia.org/wiki/File:KSUWildcats-logo.svg) is written by a tensor product of three matrices each of size $671X1024$. In this example, we generate a matrix corresponding to a given image using the following MATLAB codes:

```
>> imshow(‘wildcat.jpg’)
>> W=imread(‘wildcat.jpg’);
size(W)
ans =
    671    1024     3
```

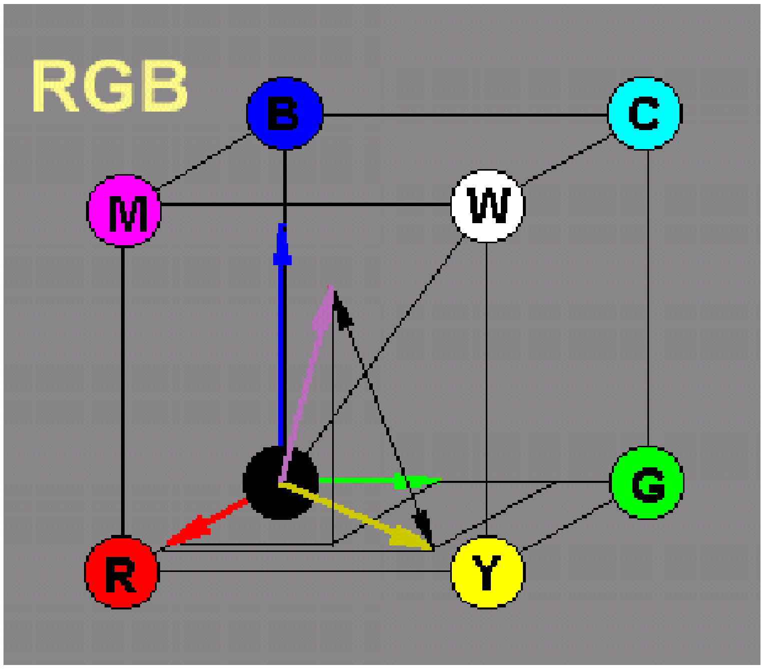

Figure 1.2: *The color cube with axes three primary colors: red, green and blue.*

```
>> RedofW=W(:,:,1);

>> GreenofW=W(:,:,2);

>> BlueofW=W(:,:,3);

>> size(RedofW)


ans =


    671   1024
```

```
>> imwrite(RedofW,‘redwildcat.jpg’)
>> imshow(‘redwildcat.jpg’)
>> imwrite(GreenofW,‘greenwildcat.jpg’)
>> imshow(‘greenwildcat.jpg’)
>> imwrite(BlueofW,‘bluewildcat.jpg’)
>> imshow(‘bluewildcat.jpg’)
```

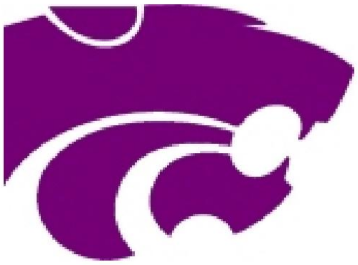

Figure 1.3: *Purple Wildcat (KSU logo).*

The red, blue and green decompositions of Figure (1.3) are illustrated in Figure (1.4). Note that in the figure, instead of intensity of true colors, gray shades corresponding to them are displayed. Also, note the darkness of the second image and the brightness of the third image indicating the fact that there is little green and a lot of blue in purple.

1.2 Objectives of image analysis

A great deal of information can be conveyed by images. Images, as we have seen, are matrices. Consequently, important features of images can be identified and analyzed by

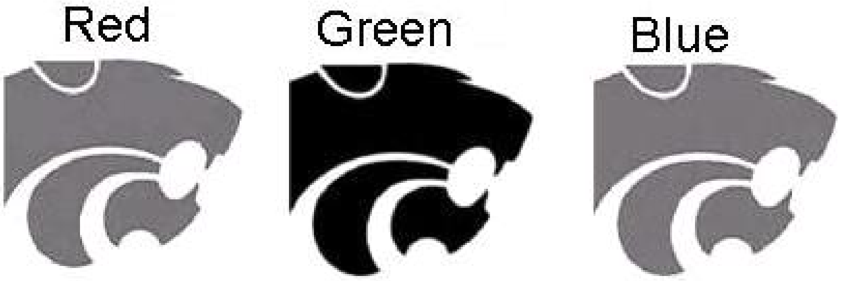

Figure 1.4: *Red, Green and Blue components of Purple Wildcat in Figure* (1.3).

studying properties of matrices. Image analysis is a study of analyzing image features and solving related problems via matrix computations or other more sophisticated mathematical tools. Many theories beyond basic linear algebra have been developed to study image features by studying their corresponding matrices or numerical data. Statistical concepts come in useful in manipulating these data. Bayes decision theory, in particular, is widely used in estimating and interpreting image data. Wavelets is another excellent tool of image analysis to study localized features of images.

Image analysis is also called image processing. The objectives of image processing are quite diverse, ranging from simply observing and identifying certain image features to transforming images into various forms by modifying these features. Two concepts of image analysis that we will be discussing in this report in detail are “classification” and “denoising”. Some other concepts of image processing are image compression, image restoration from partial damage, image enhancement, edge detection, to name but a few.

Images can be viewed as a collection of regions identified as a number of predetermined classes. However, these images or parts of them are not always clearly identifiable to the naked eye. Image classification is, therefore, done with a view to identify image parts as something familiar. Some special imageries, such as medical and satellite, often employs this technique of identification. The objective of image classification is to obtain a dependable object recognition system which will classify all the locations into a number of identified classes.

Images as we observe are often contaminated. One of the objectives of image analysis is to remove this contamination. A noisy image can be processed to reduce its noise and obtain an image which is close to its true form. There are various forms of noises that images come into contact with. The most common one is white Gaussian noise which are additive white dots following a normal distribution with mean zero and some fixed variance. The objective of image denoising is to develop an effective scheme to remove the noise corruption in an image which means removing as much noise as possible without having to compromise the quality of the image too much.

1.3 Summary of methods for image analysis

One concept of image processing is to classify parts of an image into a previously chosen set of classes. This process begins by listing a finite number of classes of interest in an image. Then we try to find a way to identify each and every location of it as some class or the other. For example, medical images might be studied by classifying them into bone, muscle, metal, etc. and one might be interested to identify satellite images as snow, water, settlement, vegetation, etc. One method of image classification is using a Bayes classifier. We would like to be able to develop a class recognition system or a classification function which, to every location of the image, assigns a class. In order to construct this classification function, we first need to obtain decision functions to determine what class an object or a location belongs to. Bayes classifier uses Bayes rule of conditional probability, namely,

$$P(x \cap y) = P(x|y)P(y),$$

in constructing decision functions. A decision function is the probability of a location being from a certain class and we assign some class to a location if the probability of that location coming from that class is greater than the probabilities of it coming from each one of the rest of the classes. A recognition system for an image is built out of a random sample of locations, called training pattern vectors, from each class and then combining means and covariance structures of samples of these classes. Once a recognition system is developed, any region of the image which are not so distinguishable visibly can be classified by using it.

Images become corrupted with noise at every stage of its acquisition. One of the most natural and widely studied image processing concepts is denoising which attempts to obtain the true image or at least a close estimate of the true image from its noisy version. One method of denoising is via wavelet thresholding where the wavelet coefficients of the image, instead of the raw image, goes through the denoising process. Wavelet coefficients are obtained by applying the wavelet transform to the image, also known as doing the wavelet decomposition of an image. Wavelet transform is a localized analogue of Fourier transform. Denoising the Fourier or Wavelet transform of an image is far more effective than denoising the actual image because the whole point of doing the transform is to gather further information on subtle properties and features of an image which are not readily available from the image itself.

Fourier transform of images is done using sinusoidal waves of varying frequency which are infinite in length or duration whereas wavelets are small localized waves so they have varying frequency but finite duration. Wavelet transform provides powerful insight into an image's spatial and frequency characteristics whereas the Fourier transform reveals only an image's frequency attributes[3]. Fourier transform converts the spatial coordinate system (x, y) of an image into the frequency coordinate system (u, v) where v and u are horizontal and vertical frequency variables. The original image $f(x, y)$ can be retrieved by means of the inverse Fourier transform of $F(u, v)$. An image of size $M \times N$ is a set of values $f(x, y)$, for $x = 0, 1, ..., M - 1$ and $y = 0, 1, ..., N - 1$. The Fourier transform formula for this image is

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(ux/M+vy/N)}$$

and the inverse Fourier transform formula for it is

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{i2\pi(ux/M+vy/N)},$$

where $F(u,v)$ are called Fourier coefficients of the image. Although $f(x,y)$ is real, $F(u,v)$ is complex, so the image of the transform can be visualized by computing the spectrum $|F(u,v)|$ or the power spectrum $P(u,v) = |F(u,v)|^2$. The Wavelet transform formula cannot be expressed easily as in the Fourier transform. However, the idea of transform is the same. Just like the sinusoidal waves obtained by expanding $e^{i2\pi(ux/M+vy/N)}$ into sines and cosines in Fourier transform, we have, in Wavelet transform, these waves replaced by wavelets, say $\phi_{jk}(x,y)$, which are scaled and dilated versions of a main wavelet $\phi(x,y)$. In this case, an image $f(x,y)$ is decomposed into a set of wavelet coefficients c_{jk}

$$f(x,y) = \sum_j\sum_k c_{jk}\phi_{jk}(x,y),$$

which is a reminiscent of the inverse Fourier transform formula and is called a reconstruction formula for Wavelet transform.

We enter the wavelet domain by decomposing an image into a set of wavelet coefficients. Then, the denoising process is performed in this domain following a scheme of thesholding these coefficients and, finally, the denoised version of the same is constructed back again in the spatial domain out of this new set of thresholded wavelet coefficients. This is explained in the diagram in Figure (1.5).

A threshold is a fixed value such that all the coefficients that are larger in size than this value are kept and the ones smaller than it are zeroed out. So, by thresholding wavelet coefficients, we are simply removing coefficients smaller than the threshold by setting them equal to zero. There are many schemes of thresholding. In this report, we will be discussing three such schemes, namely, soft thresholding, hard thresholding and Bayes Shrink thresholding. Soft and hard thresholding uses a global threshold that only depends on the variance

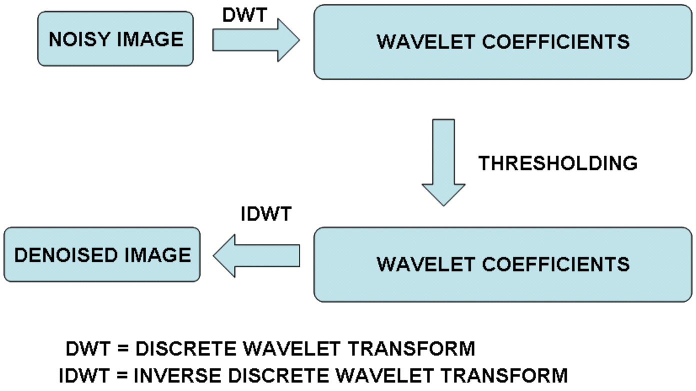

Figure 1.5: *Diagram of image denoising.*

of the noise. Wavelet transform, as will see in later chapters, will produce four different sets of wavelet coefficients. Bayes Shrink thresholding uses three different threshold values for three different sets of coefficients, namely, horizontal detail, vertical detail and diagonal detail, keeping the fourth set of coefficients, called approximation coefficients, as is.

1.4 Organization of chapters

The purpose of this report is to provide a survey of two image processing techniques supplemented with a detailed theory and illustrated examples. The rest of the report is organized as below.

Chapter 2 contains one of the two image processing techniques selected for exemplification, namely, Bayes classification. The chapter begins with some statistical theories in image processing and develops into the main discussion of this technique. Bayesian inference as opposed to classical inference is introduced. Then some theory on Bayes estimator of image data, associated risk and some measures of error follows. Finally, this chapter ends with a detailed treatment of Bayes classifier which is a recognition system for identifying classes in images. An example of the implementation of Bayes classification using MATLAB is presented along with the codes for computation.

Chapters $3, 4$ and 5 are preliminaries to a detailed discussion with examples of image denoising via wavelet thesholding at the end of chapter 6. Beside laying a foundation for this discussion by building a required theory, we will also include other mathematical and statistical theories. Wavelet theory is one of the major developments of Fourier Analysis theory. Chapter 3 is devoted solely to a discussion of Fourier transform techniques. It also includes original solutions to some selected problems in Mallat's text [8]. Chapter 4 deals with estimators in image denoising. Thesholding estimator is a denoised image estimator and noise variance estimator is the parameter used to find a threshold value. Other estimators that seem suitable for discussion are also thrown in. Chapter 5 is on Wavelet transform which includes Multiresolution Analysis, a concept on which Fast Wavelet transform algorithm is based. Fast wavelet transform algorithm is an implementation method of obtaining wavelet decomposition of signals. Chapter 6 begins with an extension of Chapter 5 to a two-dimensional case and in particular to the context of images. Then follows a discussion of wavelet thresholding schemes and, finally, detailed worked-out examples of computational implementation of these schemes. MATLAB codes used for computation and illustration of denoised images are provided.

Chapter 2

Elements of Statistics

This chapter deals with some elementary concepts in statistics that are used in image processing.

2.1 Probability distribution of image data

In chapter 1, we saw that images are matrices. Alternatively, it is a two-dimensional integer-valued discrete function, or more explicitly, a function of the form $f : \{1, 2, ..., M-2, M-1, M\} \times \{1, 2, ..., N-2, N-1, N\} \to \{0, 1, 2, ..., 253, 254, 255\}$. It should be noted, however, that if f is a color image, it is a tensor product of three such functions f_1, f_2 and f_3 corresponding to red, green and blue components of the image, or, in the matrix form, a tensor product of three corresponding matrices. We can, for our convenience, visualize this as three matrices stacked one on top of the other in the order red, green and blue from bottom to top. This idea is further generalized with multi-spectral images which can capture more than just RGB components, for example, it could have a fourth matrix at the top of the stack corresponding to the infrared component of the image. Since a color image, or a multi-spectral image, for that matter, is handled by handling each monochromatic image one at a time and repeating the process to all such images in the tensor product, we can, without loss of generality, talk of monochromatic images only whose corresponding matrices are two-dimensional.

A matrix of size $M \times N$ has $M \times N$ entries that directly correspond in the image to the array of spots of possibly different brightness level called pixel values. Each entry x thus has a probability density function (PDF) or a probability distribution which, in turn, is a discrete function $P_x : \{0, 1, 2, ..., 253, 254, 255\} \to [0, 1]$ such that $\sum\limits_{i=0}^{255} P_x(i) = 1$. Then, the probability distribution of an image f will consist of probability distributions of all the entries of its corresponding matrix, namely, $\{P_x\}_{x \in f}$. In the case of an image f having a number of distinct classes, say, $\omega_j, j = 1, 2, ..., W$, for example, of landmass, sky, water, snow, objects, scenery, etc., the probability distribution of f will be a collection of conditional probability density functions $\{P_j(x|\omega_j)\}, j = 1, 2, ..., W$ and class probabilities $P(\omega_j), j = 1, 2, ..., W$ which, if we omit, is understood as being $P(\omega_j) = 1/W$ for each j, i.e., each class occurring equally likely in the image.

2.2 Bayesian inference

Over the years, Bayesian inference has evolved as one of the popular tools to be used in image processing. Below, we discuss briefly how Bayesian approach is different from the classical frequentist approach, based on a reinterpretation of the following simple yet most important formula.

Bayes' Formula:

$$P(x|y) = \frac{P(y \cap x)}{P(y)} = \frac{P(y|x)P(x)}{P(y)}.$$

where $P(x|y)$ denotes the conditional probability distribution of x given y.

We know images are matrices. In Bayesian analysis, an image is a realization of a random matrix whose probability distribution is known a priori. There could be several ways to specify the prior probability distribution of an entry in a matrix or a spot in an image which is a discrete function defined over the set of pixel values, i.e., $\{0, 1, 2, ..., 253, 254, 255\}$. For complete specification, the prior distribution of a spot could have value 1 at a certain pixel value and 0 at the rest of the pixel values. Or, for complete ignorance, the distribution could be uniform, i.e., having the constant value $\frac{1}{255}$ at every pixel value. Or, there could be assumptions and observations which might suggest us other prior distributions. For example, a spot that is likely to be dark could have, as a prior, a binomial distribution peaked near 0. The choice of the prior is one of the main problems in Bayesian image analysis [12]and we will not be dealing with designing priors in this report, rather, we take the prior distribution for an image data as given, usually, as a Gaussian distribution.

2.3 Frequentist vs. Bayesian

Statistics is required when we want to know about the characteristics, such as mean, variance, proportion, etc. of a certain population. These are called population parameters and in order to know them, a sample data is collected and with the help of appropriate sample statistics, inferences are made about these unknown population parameters. For example, a sample proportion is a ready estimate of a population proportion. This approach is based on the empirical view that probabilities are long-run relative frequencies. However, a null hypothesis rejected at the 5% level is never to be interpreted as there being only 5% chance that the hypothesis is true and a 95% C.I. (confidence interval) (U, L) for a parameter π has a somewhat convoluted interpretation which is "If we had many samples and constructed a 95% C.I. from each sample, then about 95% of all the intervals would contain π. Hopefully, our interval belongs to the larger set of intervals that all contain π." [4] This interpretation being quite convoluted, we look for a new approach in statistics which can provide us with an interpretation that is more natural and straightforward and Bayesian Statistics is the solution. It allows us to make more natural interpretation in most contexts at the expense of not being objective. It uses some prior beliefs or probabilities about the parameter and updates them in light of a sample data. It pools together prior probabilities with the information gathered from a sample data via likelihood function and computes posterior probabilities using Bayes' formula. Posterior distribution of the parameter is the conditional distribution of the parameter given a sample data. The dependence of the posterior distribution is relative on both the parameters of the prior distribution and the sample size of the data and might be adjusted. For example, if a sample size is too small, prior probabilities have to be quite reliable whereas if the sample is large and trustworthy then prior distribution can be taken to be anything, for instance, a uniform distribution. In certain cases, Classical inference is a special case of Bayesian inference corresponding to some conventional choice of prior beliefs [5]. Unlike in classical approach, a 95% C.I. (U, L) for a parameter π can be interpreted as "The probability of π lying in (U, L) is 0.95." [4]

2.4 Bayes estimator

The Bayes principle supposes that images f are realizations of a random matrix F, say of size $M \times N$, with a known prior distribution π. Image data are usually supposed to contain additive white Gaussian noise W, a vector of length MN whose components identically and independently follow the normal distribution with mean 0 and standard deviation σ^2, written $W \sim N(0, \sigma^2 I_{MN})$. Thus, the noisy data is

$$X[n_1, n_2] = F[n_1, n_2] + W[n_1, n_2].$$

The noise $W[k_1, k_2]$ and the image $F[n_1, n_2]$ are supposed to be independent for any $0 \leq k_1, n_1 < M$ and $0 \leq k_2, n_2 < N$. By Bayes' formula, the conditional distribution of F given the observed data X, called the posterior distribution, is specified by the joint distribution of F and W which is the product of the distributions of F and W.

Bayesian inference is to estimate F from the posterior distribution which is the updated distribution in light of the data X. We denote the estimate of F by $\tilde{F}$. The error of the estimation is quantified as the loss function which can be the square Euclidean norm of the difference, i.e., $\|F - \tilde{F}\|^2$.

Since $\tilde{F}$ is an estimate of F from X, we can construct a so-called decision operator D on X specifying the estimator as $\tilde{F} = DX$. Then the risk of the estimator is the expected loss with respect to the probability distribution of the noise W denoted by

$$r(D,F) = E_W\{\|F - DX\|^2\}.$$

The Bayes risk is the expected risk with respect to the prior probability distribution π of the image and is given by

$$r(D,\pi) = E_\pi\{r(D,F)\} = E_{W,\pi}\{\|F - \tilde{F}\|^2\} = \sum_{n_1=0}^{M-1}\sum_{n_2=0}^{N-1} E_{W,\pi}\{|F[n_1,n_2] - \tilde{F}[n_1,n_2]|^2\}.$$

The Bayes estimator $\tilde{F}$ that yields the minimum Bayes risk is found by taking the infimum of the Bayes risk over all linear and non-linear operators. The corresponding Bayes decision estimator D_{nl} is the operator which achieves the minimum Bayes risk

$$r_{nl}(\pi) = \inf_{D \in \mathcal{O}_{nl}} r(D,\pi)$$

and the Bayes estimator is $\tilde{F} = D_{nl}X$ where O_{nl} is the set of all linear and non-linear operators from $\mathbb{C}^{MN}$ to $\mathbb{C}^{MN}$.

We will record, in the simple case of dimension one, some theorems as well as their proofs which are taken from Stephen Mallat's book [8]. These theorems extend naturally for images which are two-dimensional signals, in which case, the argument variables should be understood as being two-dimensional vectors, for example, $n = [n_1, n_2]$ and $k = [k_1, k_2]$.

Theorem 1. [8]The Bayes estimator $\tilde{F}$ that yields the minimum Bayes risk $r_{nl}(\pi)$ is the conditional expectation

$$\tilde{F}[n] = E\{F[n] \mid X[k] = x[k],\, 0 \le k < N\}.$$

Proof [8]: For each $n \in \{0, 1, ..., N-1\}$, let $\pi_n(y)$ be the probability distribution (CDF) of $y = F[n]$. The risk is least with $\tilde{F} = DX$ where each $\tilde{F}[n] = D_nX$ is such that $r(D_n, \pi_n) = E_W\{|F[n] - \tilde{F}[n]|^2\}$ is the minimum. This risk depends on the distribution of $W[n] = X[n] - F[n]$ which, in turn, depends on the conditional probability distribution (CDF) $P_n(x|y)$ of the data $X = x$, given $F[n] = y$:

$$r(D_n, \pi_n) = \int\int (D_n(x) - y)^2 dP_n(x|y) d\pi_n(y).$$

Let $P(x) = \int P_n(x|y) d\pi_n(y)$ be the marginal distribution of X and $\pi_n(y|x)$ be the posterior distribution of $F[n]$ given x, Bayes formula yields

$$r(D_n, \pi_n) = \int \left[\int (D_n(x) - y)^2 d\pi_n(y|x) \right] dP(x).$$

The minimum of the above double integral is achieved by minimizing the inside integral of the right hand side for each x which is attained when its first derivative vanishes:

$$\frac{\partial}{\partial D_n(x)} \int (D_n(x) - y)^2 d\pi_n(y|x) = 0.$$

This yields

$$2 \int (D_n(x) - y)^2 d\pi_n(y|x) = 0$$

which implies that

$$D_n(x) = \int y \, d\pi_n(y|x) = E\{F[n] | X = x\}.$$

Hence, $D_n X = E\{F[n] | X\}$. ■

The estimator $\tilde{F} = DX$ that achieves the linear minimum Bayes risk, namely,

$$r_l(\pi) = \inf_{D \in \mathcal{O}_l} r(D, \pi)$$

is called the Wiener estimator where O_l is the set of all linear operators from $\mathbb{C}^N$ to $\mathbb{C}^N$.

Theorem 2. [8]Without loss of generality assume $E(F[n]) = 0$. A linear estimator $\tilde{F}$ is a Wiener estimator if and only if

$$E\{(F[n] - \tilde{F}[n])X[k]\} = 0, \text{ for } 0 \leq k, n < N. \tag{0.1}$$

Proof [8]: A linear estimator $\tilde{F} = DX$ is a Weiner estimator if and only if for each $0 \leq n < N$, a linear estimation

$$\tilde{F}[n] = D_n X = \sum_{k=0}^{N-1} h[n,k] X[k]$$

minimizes the risk

$$\begin{aligned} r(D_n, \pi_n) &= E_W\{|F[n] - \tilde{F}[n]|^2\} \\ &= E\left\{\left(F[n] - \sum_{k=0}^{N-1} h[n,k]X[k]\right)^2\right\}. \end{aligned}$$

Its minimum is reached if and only if for each $0 \leq k < N$,

$$\frac{\partial r(D_n, \pi_n)}{\partial h[n,k]} = 0$$

$$-2E\left\{\left(F[n] - \sum_{k=0}^{N-1} h[n,k]X[k]\right) X[k]\right\} = 0$$

$$E\{(F[n] - \tilde{F}[n])X[k]\} = 0.$$

■

The equation (0.1) implies that $F[n] - \tilde{F}[n]$ and $X[k]$ are non-correlated for any $0 \leq k, n < N$. Furthermore, since they are jointly Gaussian, they will be independent. In this case, it can be verified that the Weiner estimator $\tilde{F}$ is the Bayes estimator.

2.5 Measures for error

After we get an estimator of the true image, we would like to measure how close we are. We would like to measure the error, which somehow measures the difference between the true image and an estimator. One measure of this error is the square Euclidean norm

$$\|F - \tilde{F}\|^2 = \sum_{n_1=1}^{M} \sum_{n_2=1}^{N} [F(n_1, n_2) - \tilde{F}(n_1, n_2)]^2,$$

which is fairly simple and accurate in most cases. Another not-so-common and not-very-accurate measure could be the maximum absolute entrywise difference

$$\text{Maxabs} = arg\, max[|F(n_1, n_2) - \tilde{F}(n_1, n_2)|].$$

Mean square error (MSE) also gives a measure of this error which is defined as

$$MSE = \frac{1}{MN}\|F - \tilde{F}\|^2 = \frac{1}{MN}\sum_{n_1=1}^{M}\sum_{n_2=1}^{N}[F(n_1, n_2) - \tilde{F}(n_1, n_2)]^2.$$

Two measures of the numerical value of the risk $E\{\|F - \tilde{F}\|^2\}$ that are most common in image processing are Signal-to-Noise Ratio (SNR) and Peak Signal-to-Noise Ratio (PSNR). These are inversely proportional to the risk which means that a higher value of these measures in decibels correspond to a lower risk.

$$SNR = 10log_{10}\left(\frac{E\{\|F\|\}^2}{E\{\|F - \tilde{F}\|\}^2}\right) = 10log_{10}\left(\frac{\sum_{n_1=1}^{M}\sum_{n_2=1}^{N}[F(n_1, n_2)]^2}{\sum_{n_1=1}^{M}\sum_{n_2=1}^{N}[F(n_1, n_2) - \tilde{F}(n_1, n_2)]^2}\right)$$

$$PSNR = 10log_{10}\left(\frac{255^2}{E\{\|F - \tilde{F}\|^2\}}\right) = 10log_{10}\left(\frac{255^2}{\frac{1}{MN}\sum_{n_1=1}^{M}\sum_{n_2=1}^{N}[F(n_1, n_2) - \tilde{F}(n_1, n_2)]^2}\right),$$

where 255 comes from being the maximum possible pixel value for representation of 8 bits per sample.

2.6 Bayes classifier

This section is devoted to illustrate another main concept of Bayesian inference in image processing. One of the objectives in image analysis is to correctly identify natural classes of images. For example, in satellite images, we may wish to classify image parts as water, urban area, mountains, forest, pasture, sand, etc. We usually do this classification with multi-channel images which have richer image data. Here, channels refer to different channels of information available for each pixel. Gray or black-and-white images are single-channel images as there is only a scalar gray intensity value for each pixel. Color images, on the other hand, are multi-channel images with red, green and blue intensity values for each pixel which gives a three-dimensional pattern vector at each pixel. Multi-spectral images could have even higher dimensional pattern vectors as these capture even more channels of pixel information such as infra-red and ultra-violet intensities or frequencies which are not visible to the human eye.

Let us now describe how classification of pattern vectors is done using the Bayes classifier. Let x denote the $n-$dimensional pattern vector and let there be $\omega_j, j = 1, 2, ..., k$ classes. The probability of observing a random pattern vector x from the class ω_j is given, using Bayes' rule, by

$$P(x \cap \omega_j) = P(x|\omega_j)P(\omega_j),$$

where $P(x|\omega_j)$ is the probability density function of the pattern vectors in class ω_j, and $P(\omega_j)$ is the probability of occurrence of class ω_j in the image. So, the decision functions for the Bayes classifier will have to be of the form

$$d_j(x) = P(\omega_j|x) \propto P(x|\omega_j)P(\omega_j),$$

based on which, we make a decision

$$x \in \omega_j \text{ if } d_j(x) > d_i(x) \text{ for all } i = 1, 2, ..., k, i \neq j. \tag{0.2}$$

Then, we will have a recognition or classification function

$$class(x) = j \text{ if } x \in \omega_j.$$

Ties of classes will be resolved arbitrarily.

Usually, we assume each class having a Gaussian PDF:

$$P(x|\omega_j) = \frac{1}{(2\pi)^{n/2}|C_j|^{1/2}} exp\left(-\frac{1}{2}[(x - m_j)^T C_j^{-1}(x - m_j)]\right),$$

where C_j and m_j are the covariance matrix and the mean vector of the pattern population of class ω_j, and $|C_j|$ is the determinant of C_j. In practice, these parameters are not computed from the whole population, but from a sample X of patterns representative of the class of interest. These patterns which are used for parameter estimation are called training patterns. Once we have the parameter estimates of each class, we can classify any image or its part - this is but a collection of pattern vectors with corresponding 2-D locations. These patterns which are used for testing the performance of classification are called test patterns or independent patterns.

Since ln is monotonic, the decision (0.2) is invariant under the ln transformation. So, we can replace $d_j(x)$ by $ln[d_j(x)]$, thereby, using

$$\begin{aligned}d_j(x) &\propto ln[P(x|\omega_j)P(\omega_j)]\\ &= lnP(x|\omega_j) + lnP(\omega_j)\\ &= -\frac{1}{2}ln|C_j| - \frac{1}{2}[(x-m_j)^T C_j^{-1}(x-m_j)] + lnP(\omega_j) - \frac{n}{2}ln(2\pi).\end{aligned}$$

Since the last term is independent of the class, we can drop it and use

$$d_j(x) \propto lnP(\omega_j) - \frac{1}{2}ln|C_j| - \frac{1}{2}[(x-m_j)^T C_j^{-1}(x-m_j)].$$

To compute the expression $[(x-m_j)^T C_j^{-1}(x-m_j)]$ which is called the Mahalanobis distance between vector x and the mean vector m_j of a collection T_j of training pattern vectors of class ω_j with covariance matrix C_j, we use the MATLAB function mahalanobis as below

```
>>mahalanobis(x,Cj,mj)
```

which is written as in page 487 of [3], where, in turn, mean vector and covariance matrix are computed using the MATLAB function covmatrix as below

```
>>[Cj,mj]=covmatrix(Tj)
```

from page 476 of [3].

Finally, we classify a collection X of pattern vectors representing an image part by computing the classification function d using the MATLAB function bayesgauss as below

```
>>d=bayesgauss(X,CA,MA,P)
```

from page 493 of [3], where, in turn, CA and mA are computed as below

```
>>CA=cat(3,C1,C2,...,Ck);
>>mA=cat(2,m1,m2,...,mk);
>>P=[P(w1),P(w2),...,P(wk)];
```

where the built-in MATLAB function cat with argument 3 is a three dimensional concatenation of all the arguments followed by 3, which is a tensor product of these arguments or a stacking of their corresponding matrices one on top of the other and the same function with argument 2 is a two dimensional concatenation of all the arguments followed by 2, which is adjoining the vectors side by side columnwise.

Now let us provide an example of how Bayesian classification of an image is done. This will also illustrate how we can find training patterns T_j representing a certain class ω_j. We are using a satellite image in Figure (2.1) (source: http://images.smh.com.au/2009/03 /17/420880/Pyramids-Cairo-Satellite-600x400.jpg) having four distinct classes, namely, sand (1), settlement(2), vegetation(3) and road(4).

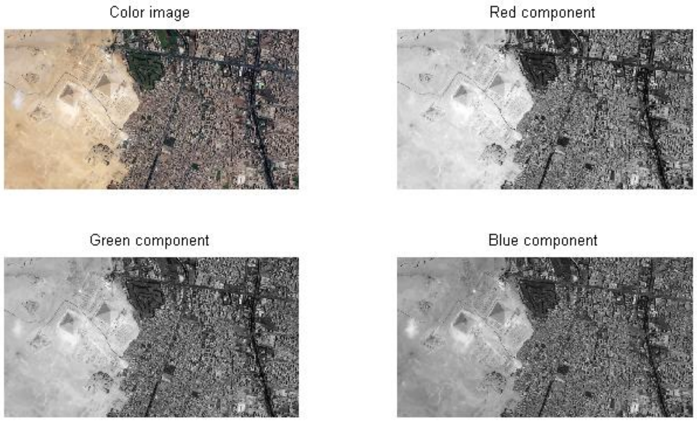

Figure 2.1: *A color image and its RGB components having four distinct classes, namely, sand(1), settlement(2), vegetation(3) and road(4).*

Below are given a few MATLAB commands for the work required to do the computation of Bayes classifier.

```
>> P=imread(‘pyramids.jpg’);

>> P1=P(:,:,1);

>> subplot (2,2,1),imshow (P);title(‘Color image’);

>> subplot (2,2,2),imshow (P1);title (‘Red component’);

>> B1=roipoly(P);

>> B=B1+B2+B3+B4;

>> [X1,R1]=imstack2vectors(P,B1);

>> Y1=X1(1:2:end,:);%extracting training patterns out of X1

>> Z1=X1(2:2:end,:);%extracting independent test patterns out of X1

>> [C1,m1]=covmatrix(Y1);

>> CA=cat(3,C1,C2,C3,C4);

>> mA=cat(2,m1,m2,m3,m4);

>> dX1=bayesgauss(X1,CA,mA);

>> dY1=bayesgauss(Y1,CA,mA);

>> dZ1=bayesgauss(Z1,CA,mA);

>> lIY11=length(find(dY1==1));

>> lIY12=length(find(dY1==2));

>> lIX13=length(find(dX1==3));

>> lIY14=length(find(dY1==4));

>> lIZ11=length(find(dZ1==1));

>> Miss1=find(dX1˜=1);

>> R1new=R1(Miss1,:);

>> B1new=B1;

>> for r=1:length(R1new)

>> B1new(R1new(r,1), R1new(r,2))=0;

>> end

>> Bnew=B1new+B2new+B3new+B4new;
>> testB=roipoly(P);
>> [testX,testR]=imstack2vectors(P,testB);
>> dtestX=bayesgauss(testX,CA,mA);
>> dX=dtestX;
>> R=testR;
>> class1=find(dX==1);
>> Rnew=R(class1,:);
>> Bnew1=testB;
>> for r=1:length(Rnew)
>> Bnew1(Rnew(r,1), Rnew(r,2))=0;
>> end
>> newB=Bnew1+Bnew2+Bnew3+Bnew4;
>> imshow(newB);
```

The built-in MATLAB function “roipoly” displays the image of the argument and lets us select from it a polygonal region of interest interactively using the mouse pointer. After carefully selecting a region of interest from a class, we rearrange all the patterns in that region in a column by using the MATLAB function “imstack2vectors” from page 575 of [3]. Then we obtain training patterns by deleting all even rows from this column and deleting all odd rows from the same column will give us patterns of the same class independent to the training patterns. Here, $B1$ is a region of interest we select from class sand(1). $X1$ is a rearrangement in a column of those pattern vectors of P which lie in the region $B1$ and the order of the arrangement is starting from top to bottom along the first column, the second column, and so on. $R1$ is the column of coordinates of corresponding 2-D locations of these pattern vectors. Next, we extract $Y1$ which are odd rows of $X1$ to implement as training patterns from class sand(1). Independent to these training patterns are $Z1$, the even rows of $X1$, which will be used to test the performance of $Y1$. Similarly, other polygonal regions of interest $B2, B3$ and $B4$ are selected patterns from classes settlement(2), vegetation(3) and road(4) respectively.

How good a recognition system is depends on how good are the parameter estimates of these classes. We can check for stability of these parameter estimates by examining the recognition results obtained with the training and independent patterns for each class. The results are summarized in Table (2.1).

Table 2.1: *Summary of classification results of the recognition system*

	Training Patterns (Y)					
		Classified into				
Actual class	# of samples	1	2	3	4	Correctly classified
1	1658	1640	18	0	0	98.91%
2	1750	1	1712	31	6	97.83%
3	1777	0	36	1649	92	92.8%
4	564	0	9	28	527	93.44%
	Independent Patterns (Z)					
		Classified into				
Actual class	# of samples	1	2	3	4	Correctly classified
1	1658	1641	17	0	0	98.97%
2	1750	4	1713	30	3	97.89%
3	1777	0	24	1660	93	93.42%
4	563	0	11	29	533	92.9%

Figure (2.2) displays the regions purposively selected from each class to build a classification recognition system and the same regions filled with black locations representing misclassification of the system.

Since the percentage of training and independent patterns recognized correctly are about the same, we can declare that our recognition system is quite operable and that we can use it to classify any region of interest into the three regions, namely, sand(1), settlement(2), vegetation(3) and road(4). For this we only need to apply the MATLAB function “bayesgauss” to our region of interest X along with CA and mA which are covariance and mean structures of the four classes in our recognition system. Figure (2.3) displays the region of interest chosen to test the classification performance of the recognition system we have built and the results, black dots indicating locations classified into a particular class. The output is a column vector whose length is equal to the total number of input patterns. Each pattern is classified into one of the four classes which is indicated by the

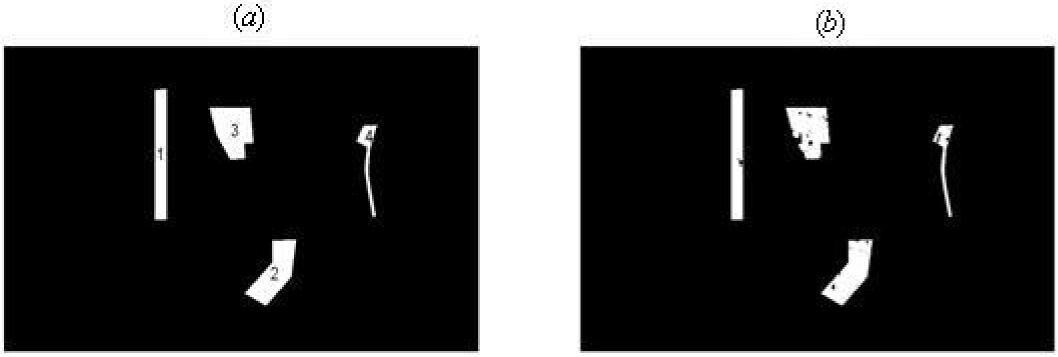

Figure 2.2: *(a)Regions selected from classes: sand(1), settlement(2), vegetation(3) and road(4) for building a recognition system. (b)Black dots represent locations misclassified by the system.*

corresponding entry being either $1, 2, 3$ or 4.

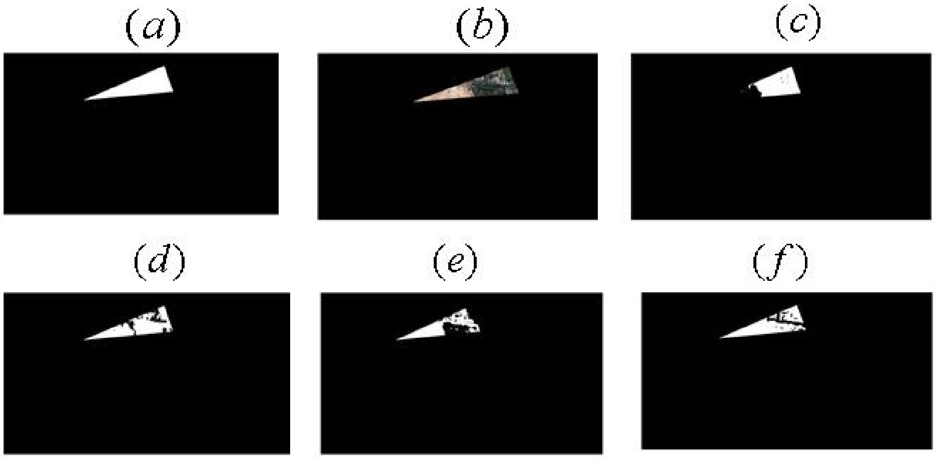

Figure 2.3: *(a)Mask region selected for classification, (b) Actual region obtained by applying the mask to the actual image, (c)Locations identified as "sand" marked by black dots, (d)Locations identified as "settlement" marked by black dots, (e)Locations identified as "vegetation" marked by black dots and (f)Locations identified as "road" marked by black dots.*

Chapter 3

Elements of Fourier Analysis

This chapter provides the basics of Fourier Analysis that is needed to develop a theory of image processing using wavelet methods.

3.1 Some basic concepts in Real Analysis

Basic real analysis is often assumed for the study of Fourier Analysis. Here we review some basic definitions and standard results (without proofs) that we will later be referring to.

Definition 3 (Indicator function)**.** The *indicator function* of an interval $(a, b) \subseteq \mathbb{R}$ is the function

$$\mathbf{1}_{(a,b)}(x) = \begin{cases} 1 & \text{if } \ x \in (a, b) \\ 0 & \text{if } \ x \notin (a, b). \end{cases}$$

Definition 4 (Dirac distribution)**.** The *Dirac distribution* δ is the limit as n goes to ∞ of a sequence of probability distribution functions δ_n whose support is contained in $(-1/n, 1/n)$. It is not a function since $\delta(t) = 0$ for all $t \neq 0$. However, it is a probability distribution and has the total integral equal to 1. The most important property of the Dirac is the following:

$$\int_{\mathbb{R}} \delta(t)\phi(t)\, dt = \phi(0).$$

Definition 5 (Discrete Dirac)**.** The *discrete Dirac* δ is the indicator function of the singleton set $\{0\}$.

Definition 6 ($L(\mathbb{R})$)**.** A complex-valued function $f(t)$ defined on $\mathbb{R}$ is said to be in $L(\mathbb{R})$ if it is *absolutely integrable.*i.e., $\|f\| := \int_{\mathbb{R}} |f(t)|\, dt < \infty$.

Definition 7 ($L^2(\mathbb{R})$)**.** A complex-valued function $f(t)$ defined on $\mathbb{R}$ is said to be in $L^2(\mathbb{R})$ and is called a *finite energy analog signal* if it is *square integrable.*i.e., $\|f\| := \int_{\mathbb{R}} |f(t)|^2\, dt < \infty$.

Definition 8 ($\ell(\mathbb{Z})$)**.** A complex-valued function $f[n]$ defined on $\mathbb{Z}$ is said to be in $\ell(\mathbb{Z})$ if $\|f\| := \sum_{n=-\infty}^{+\infty} |f[n]| < \infty$.

Definition 9 ($\ell^2(\mathbb{Z})$)**.** A complex-valued function $f[n]$ defined on $\mathbb{Z}$ is said to be in $\ell^2(\mathbb{Z})$ and is called a *finite energy discrete signal* if $\|f\| := \sum_{n=-\infty}^{+\infty} |f[n]|^2 < \infty$.

Definition 10 (Convolution)**.** Given two functions $f, g \in L(\mathbb{R})$, their *convolution* is defined to be

$$f \star g(x) = \int_{\mathbb{R}} f(x-y)g(y)\, dy.$$

Likewise, given $f, g \in \ell(\mathbb{Z})$, their *convolution* is defined to be

$$f \star g(x) = \sum_{n=-\infty}^{+\infty} f[n]g(x-n).$$

The convolution pools together regularity properties of both functions. Thus, the lack of regularity of a function can be removed by convolving it with a regular function. Also, convolving a function with itself makes it smoother.

Consider the constant function $\theta_0 = \mathbf{1}_{[0,1)}$ which is called the unit box window. $\theta_0 \star \theta_0$ is a linear function

$$\theta_1(x) = \theta_0 \star \theta_0(x) = \begin{cases} x & \text{if} \quad x \in [0,1] \\ 2-x & \text{if} \quad x \in [1,2] \\ 0 & \text{if} \quad x \notin (0,2). \end{cases}$$

Convolving θ_0 with itself $m+1$ times, we get a box spline θ_m of degree m which is $m-1$ times continuously differentiable.

Definition 11 (Linear Independence)**.** Let H be $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$. A collection of elements $\{\phi_n\} \subseteq H$ is said to be *linearly independent* if no element can be obtained as a linear combination of the rest.

Definition 12 (Basis)**.** Let H be $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$. A collection $\mathcal{B} = \{\phi_n\}$ of linearly independent elements of H is said to be a *basis* of $V \subseteq H$ if every element in V can be decomposed as some linear combination of these elements. Then V is said to be a *subspace* of H generated by $\mathcal{B}$.

Definition 13 (Orthonormal Basis)**.** Let H be $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$. A basis $\{\phi_n\}$ of H is said to be an *orthonormal basis* if $\langle \phi_m, \phi_n \rangle := \int_{\mathbb{R}} \phi_m(t)\phi_n^*(t)\, dt = \delta[n-m]$, where $\phi_n^*(t)$ denotes the complex conjugate of $\phi_n(t)$,i.e., $\phi_n^*(t) = a - ib$ if $\phi_n(t) = a + ib$.

Example 14. $\{\varphi_n(t)\}_{n\in\mathbb{Z}} := \{e^{2\pi itn}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$.

Example 15. $\{E_k(t)\}_{k\in\mathbb{Z}} := \frac{1}{b-a}\{e^{2\pi ikt/(b-a)}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of $L^2([a,b])$.

3.2 Fourier Transform

Fourier Analysis is the study of frequencies and is also called frequency analysis or harmonic analysis. Frequencies occur everywhere in all applied sciences and, in particular, in the study of natural images. In this section, we introduce some terminology as well as some results relevant to the study of our report.

Definition 16 (Fourier Transform)**.** Let $f \in L^2(\mathbb{R})$ be integrable. Then the *(direct) Fourier transform* of f is defined to be

$$\hat{f}(\omega) := \int_R f(t)e^{-i\omega t}\, dt.$$

The Fourier transform is a tool to analyze a signal frequencywise. $\hat{f}(\omega)$ is the correlation of $f(t)$ with a sinusoidal wave $e^{i\omega t}$ and thus gives the measure of how much of the frequency

ω is in the signal f. In other words, $\hat{f}(\omega)$ is the amplitude of the wave $e^{i\omega t}$ in the signal f. Consequently, we can synthesize f back by summing over all the frequencies as follows:

$$f(t) = \frac{1}{2\pi} \int_{\mathbb{R}} f(\omega) e^{i\omega t}\, d\omega.$$

This is also a formula for the *inverse Fourier transform.*The inverse Fourier transform of f is denoted by $\check{f}$. Hence, $f = \check{\hat{f}}$.

The two-dimensional Fourier transform is a natural extension of the one-dimensional case.

Definition 17 (Two-dimensional Fourier Transform)**.** Let $f \in L^2(\mathbb{R}^2)$ be integrable. Then the *two-dimensional Fourier transform* of f is defined to be

$$\hat{f}(\omega_1, \omega_2) := \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x_1, x_2) e^{-i(\omega_1 x_1 + \omega_2 x_2)}\, dx_1\, dx_2.$$

Definition 18 (seperable function)**.** A function f on $\mathbb{R}^2$ is said to be *separable* in rectangular coordinates (x_1, x_2) if there exist functions f_1 and f_2 on $\mathbb{R}$ such that $f(x_1, x_2) = f_1(x_1) f_2(x_2)$.

Some elementary properties of the Fourier transform are listed below. Note that $(\cdot)$ represents the argument of $f(\cdot)$. Given $f, g \in L^2(\mathbb{R})$, $b \in \mathbb{R}$ and $s > 0$, we have

(i) $\widehat{f+g} = \hat{f} + \hat{g}$.

(ii) $\widehat{bf} = b\hat{f}$.

(iii) $\widehat{f(\cdot - y)}(\omega) = e^{-iy\omega} \hat{f}(\omega)$.

(iv) $\widehat{e^{-iy\cdot} f(\cdot)}(\omega) = \hat{f}(\omega - y)$.

(v) $\widehat{f(s\cdot)}(\omega) = s^{-1} \hat{f}(s^{-1}\omega)$.

(vi) $\widehat{f^{(p)}(\cdot)}(\omega) = \widehat{\frac{d^p f(\cdot)}{d(\cdot)^p}}(\omega) = (i\omega)^p \hat{f}(\omega)$.

(vii) $\widehat{f \star g} = \hat{f}\hat{g}$.

(viii) Denoting $\bar{f}(x) = f(-x)$, we have $\bar{\hat{f}} = \hat{\bar{f}}$.

(ix) Denoting by f^*, the complex conjugate of f, we have $\hat{f}^* = \bar{\hat{f}}^*$.

(x) Denoting $\tilde{f} = \bar{f}^*$, the Hermitian conjugate of f, we have $\widehat{\tilde{f}} = \hat{f}^*$.
In particular, $\widehat{f \star (\tilde{f})} = |\hat{f}|^2$.

(xi) (Plancherel's Identity) $\|f\| = \frac{1}{2\pi}\|\hat{f}\|$.

(xii) (Parseval's Identity) $\langle f, g\rangle := \int_{\mathbb{R}} f(t)g(t)^* \, dt = \frac{1}{2\pi}\int_{\mathbb{R}} \hat{f}(t)\hat{g}(t)^* \, dt =: \frac{1}{2\pi}\langle \hat{f}, \hat{g}\rangle$.

(xiii) If $f \in L^2(\mathbb{R}^2)$ is separable with $f(x_1, x_2) = f_1(x_1)f_2(x_2)$, then its Fourier transform is $\hat{f}(\omega_1, \omega_2) = \hat{f}_1(\omega_1)\hat{f}_2(\omega_2)$, where $\hat{f}_1$ and $\hat{f}_2$ are one-dimensional Fourier transforms of f_1 and f_2.

There is an interesting interplay between global regularity and decay of the absolute value of a signal and its Fourier transform. The following statements give some information regarding this interplay.

Theorem 19. [8][Riemann-Lebesgue lemma] If $f \in L^2(\mathbb{R})$, then $\lim\limits_{\omega\to\pm\infty} \hat{f}(\omega) = 0$.

Theorem 20. [8]A function f is bounded and p times continuously differentiable with bounded derivatives if

$$\int_{-\infty}^{+\infty} |\hat{f}(\omega)|(1 + |\omega|^p)\, d\omega < \infty.$$

The differentiability of a signal translates into the decay of its direct and inverse Fourier transform. For example, consider the signal $f(x) = e^{-|x|}$. The Fourier transform of f is given by $\hat{f}(\omega) = \frac{1}{1+\omega^2}$. Since $\hat{f}(\omega)$ is infinitely differentiable with all of its derivatives in $L(\mathbb{R})$, we have an exponential decay in f. On the other hand, since f is not differentiable at 0, $\hat{f}(\omega)$ is constrained to have only a second order polynomial decay.

The following theorem says that it is not possible to construct a function of compact support whose Fourier transform has a compact support.

Theorem 21. [8]If $f \neq 0$ has a compact support then $\hat{f}$ cannot be zero on a whole interval. Similarly, if $\hat{f} \neq 0$ has a compact support then f cannot be zero on a whole interval.

The state of a one-dimensional free particle is given by $f \in L^2(\mathbb{R})$. The probability density of the location of this particle is $\frac{1}{\|f\|^2}|f(t)|^2$. The probability density of the momentum of this particle is $\frac{1}{2\pi\|f\|^2}|\hat{f}(\omega)|^2$.

Let u and ξ denote the average location and momentum of this particle which are given by

$$u = \frac{1}{\|f\|^2}\int_{-\infty}^{+\infty} t|f(t)|^2\,dt$$

and

$$\xi = \frac{1}{2\pi\|f\|^2}\int_{-\infty}^{+\infty} \omega|\hat{f}(\omega)|^2\,d\omega.$$

The temporal and frequency variances are then given by

$$\sigma_t^2 = \frac{1}{\|f\|^2}\int_{-\infty}^{+\infty} (t-u)^2|f(t)|^2\,dt$$

and

$$\sigma_\omega^2 = \frac{1}{2\pi\|f\|^2}\int_{-\infty}^{+\infty} (\omega-\xi)^2|\hat{f}(\omega)|^2\,d\omega$$

respectively.

Definition 22. [Heisenberg Uncerainty]The temporal variance and the frequency variance of $f \in L^2(\mathbb{R})$ satisfy

$$\sigma_t^2\sigma_\omega^2 \geq \frac{1}{4}.$$

Heisenberg Uncertainty principle states the underlying trade-off between time and frequency localization. This is also clear from property (5.1) of the Fourier transform that if we localize f by scaling it by $0 < s < 1$, then $\hat{f}$ gets scaled by $1/s$, thereby losing in frequency the localization we gained in time.

For the sake of convenience in computer application, we often descritize an analog signal. This is done by uniformly sampling an analog signal $f(t)$ at interval p and recording its sample values $\{f(np)\}_{n\in\mathbb{Z}}$. The descritized version of f is the following weighted Dirac sum

$$f_d(t) = \sum_{n=-\infty}^{+\infty} f(np)\delta(t-np).$$

The following theorem says that sampling an analog signal periodizes its Fourier transform.

Theorem 23. [8] Let f_d be the discrete signal obtained by sampling f at interval p. Then its Fourier transform is given by

$$\hat{f}_d(\omega) := \frac{1}{p} \sum_{n=-\infty}^{+\infty} \hat{f}\left(\omega - \frac{2n\pi}{p}\right),$$

which is periodic with period $\frac{2\pi}{p}$.

In particular, by Theorem 23, the Fourier transform of any finite energy discrete signal $a[n] \in \ell^2(\mathbb{Z})$ is 2π-periodic.

In practice, for example, in the case of an image data, a discrete signal $f[n]$ is known over a finite domain, say, $0 \leq n < N$. In order to apply the Fourier theory, we take the following approach. We extend $f[n]$ over all of $\mathbb{Z}$ with a periodization over N samples, i.e, by redefining it by $f[nmodN]$. The space of signals of period N is an Euclidean space of dimension N. The inner product in this space is defined by

$$\langle f, g \rangle = \sum_{n=0}^{N-1} f[n] g^*[n].$$

Theorem 24. [8]The family $\{e_k[n] := e^{\left(\frac{i2\pi kn}{N}\right)}\}_{0 \leq k < N}$ is an orthogonal basis of the space of signals of period N.

Definition 25 (Discrete Fourier Transform)**.** The *discrete Fourier transform* of a signal f of period N is

$$\hat{f}[k] = \langle f, e_k \rangle = \sum_{n=0}^{N-1} f[n] e^{\left(\frac{-i2\pi kn}{N}\right)}.$$

Since $\|e_k\|^2 = N$, an inverse discrete Fourier transform formula is

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}[k] e^{\left(\frac{i2\pi kn}{N}\right)}.$$

Definition 26 (Circular convolution)**.** The *circular convolution* of two signals f and h of period N, which is again a signal of period N is

$$f \star h[n] = \sum_{p=0}^{N-1} f[p]h[n-p].$$

The following theorem establishes the periodicity of the Fourier transform of a sampled two-dimensional signal.

Theorem 27. [8] Let f_d be the discrete signal obtained by sampling f at interval p_1 along x_1 and at interval p_2 along x_2. Then its Fourier transform is given by

$$\hat{f}_d(\omega_1, \omega_2) := \frac{1}{p_1 p_2} \sum_{n_1, n_2 = -\infty}^{+\infty} \hat{f}\left(\omega_1 - \frac{2n_1\pi}{p_1}, \omega_2 - \frac{2n_2\pi}{p_2}\right),$$

which is periodic with period $\frac{2\pi}{p_1}$ along ω_1 and period $\frac{2\pi}{p_2}$ along ω_2.

Definition 28 (Filter)**.** Modifying a function f with an operator L on $L^2(\mathbb{R})$ defined by $Lf = f \star h$ is called a *frequency filtering* of f. Both the operator L and the function h are referred to as a *filter.* Also, h is alternatively called the *impulse response function* of the filter and $\hat{h}$ is called the *transfer function* of the filter. This operation filters out all frequencies of f outside the support of the transfer function.

The following example illustrates an algorithm of discrete image filtering with the help of separable finite impulse response filters.

Consider an image in a square grid of $(2M+1)^2$ pixels. Suppose the impulse response $h[n_1, n_2]$ of the filter L is separable:

$$h[n_1, n_2] = h_1[n_1]h_2[n_2].$$

Further, suppose $h_1 = h_2 = (2M+1)^{-1}\mathbf{1}_{[-M,M]}$. Then the two-dimensional convolution,

$$Lf[n_1, n_2] = f \star h[n_1, n_2] = \frac{1}{(2M+1)^2} \sum_{p_1=-M}^{M} \sum_{p_2=-M}^{M} f[n_1 - p_1, n_2 - p_2]h[p_1, p_2,] \tag{0.1}$$

after switching f and h on the right hand side, reduces to

$$Lf[n_1, n_2] = \frac{1}{(2M+1)^2} \sum_{p_1=n_1+M}^{n_1-M} \sum_{p_2=n_2+M}^{n_2-M} f[p_1, p_2]. \tag{0.2}$$

A direct calculation with (0.1) requires $(2M+1)^2$ additions per pixel point whereas with the factorization (0.2), it only requires $2(2M+1)$ additions per pixel.

3.3 Problems/Solutions

This section of the report includes solutions of some selected problems in Mallat's book [8]. The problems are selected in a way so that giving too many new definitions could be avoided and the solutions are intended to present some basic analysis techniques before a new Fourier analysis tool called wavelets is introduced in the next chapter.

Problem 1. (#3.4 p.85 in [8])Prove that if $f \in L^2(\mathbb{R})$ and $\sum\limits_{n=-\infty}^{+\infty} f(t-n) \in L^2([0,1])$ then

$$\sum_{n=-\infty}^{+\infty} f(t-n) = \sum_{k=-\infty}^{+\infty} \hat{f}(2k\pi) e^{i2\pi kt}.$$

Solution. Since $\sum\limits_{n=-\infty}^{+\infty} f(t-n) \in L^2([0,1])$ and, by Example (15), $\{e^{2\pi ikt}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of $L^2([0,1])$, there exist coefficients c_k such that

$$\sum_{n=-\infty}^{+\infty} f(t-n) = \sum_{k\in\mathbb{Z}} c_k e^{2\pi ikt}.$$

Taking the inner product with $e^{2\pi ijt}$, by orthonormality, the right hand side reduces to c_j. Thus,

$$\begin{aligned}
c_j &= \sum_{n=-\infty}^{+\infty} \int_0^1 f(t-n) e^{-2\pi ijt}\, dt \\
&= \sum_{n=-\infty}^{+\infty} \int_{-n}^{1-n} f(u) e^{-2\pi ij(n+u)}\, du \qquad \text{(by change of variable } u = t-n) \\
&= \sum_{n=-\infty}^{+\infty} \int_{-n}^{1-n} f(u) e^{-2\pi iju}\, du \\
&= \int_{-\infty}^{+\infty} f(u) e^{-2\pi iju}\, du \\
&= \hat{f}(2\pi j).
\end{aligned}$$

Now, substituting for the coefficients c_k in the above expression for $\sum_{n=-\infty}^{+\infty} f(t-n)$, we finish the proof. ■

Problem 2. (#3.16 p.86 in [8])Let h^{-1} be the inverse of h defined by $h \star h^{-1}[n] = \delta[n]$.

(a) Prove that if h has a finite support then h^{-1} has a finite support if and only if $h[n] = c\delta[n-p]$ for some $c \neq 0$ and $p \in \mathbb{Z}$.

(b) Find a sufficient condition on $\hat{h}(\omega)$ for h^{-1} to be a stable filter.

Remark 29. The constant c in part (a) is missing in the text [8], which is a typo as $h[n]$ can be a multiple of a shifted δ.

Solution. Proof of (a):

First suppose h has a finite support and $h[n] = \delta[n-p]$ for some $p \in \mathbb{Z}$. Taking the Fourier transform of $h[n] = \delta[n-p]$ and $h \star h^{-1}[n] = \delta[n]$ yields $\hat{h}(\omega) = 1$ and $\hat{h}(\omega)\widehat{h^{-1}}(\omega) = 1$ respectively. This implies $\widehat{h^{-1}}(\omega) = 1$. So, the inverse Fourier transform h^{-1} is supported at a single point. Conversely, suppose $\{-N, -N+1, ..., -1, 0, 1, ...N-1, N\}$ and $\{-M, -M+1, ..., -1, 0, 1, ...M-1, M\}$ are finite supports of h and h^{-1} respectively. Taking the Fourier transform of $h \star h^{-1}[n] = \delta[n]$ yields $\hat{h}(\omega)\widehat{h^{-1}}(\omega) = 1$, which, for the finitely supported h and h^{-1}, is

$$\sum_{j=-N}^{N} a_j e^{-i\omega j} \sum_{k=-M}^{M} b_k e^{-i\omega k} = 1, \tag{2.1}$$

where $a_j = \langle h[j], e^{-i\omega j} \rangle$ and $b_k = \left\langle h^{-1}[k], e^{-i\omega k} \right\rangle$ are Fourier coefficients of h and h^{-1} respectively. Since $h \neq 0, \exists\, p \in \mathbb{Z}$ such that $h[p] \neq 0$. Then $a_p \neq 0$, say, $a_p = c$. Next we need to invoke the following result which is easily proved by induction.

Result: If $\sum_{j=-N}^{N} a_j e^{-i\omega j} \sum_{k=-M}^{M} b_k e^{-i\omega k} = 1$ for every ω and $a_0 = 1$. Then $b_0 = 1, a_j = 0$ for every $j \neq 0$, and $b_k = 0$ for every $k \neq 0$.

Using this result in (2.1) we must have $b_p = \frac{1}{c}, a_j = 0$ for every $j \neq p$, and $b_k = 0$ for every $k \neq p$. Hence, we get $\widehat{h^{-1}}(\omega) = \sum_{j=-N}^{N} h[j] e^{-i\omega j} = c$, the inverse Fourier transform of which is $h[n] = c\delta[n-p]$.

Proof of (b):

Recall that a filter h^{-1} is said to be stable if $|f \star h^{-1}[n]|$ is bounded for all bounded $f[n]$. Since $|f \star h^{-1}[n]| \leq \sup_{n\in\mathbb{Z}} |f[n]| \sum_{k=-\infty}^{+\infty} |h^{-1}[k]|$, it is sufficient to get a condition so that $h^{-1} \in \ell(\mathbb{Z})$. Note

$$h^{-1}[j] = \int_{-\pi}^{\pi} \frac{1}{\hat{h}(\omega)} e^{i\omega j}\, d\omega = \int_{-\pi}^{\pi} \frac{1}{\hat{h}(\omega)} cos(\omega j)\, d\omega + i \int_{-\pi}^{\pi} \frac{1}{\hat{h}(\omega)} sin(\omega j)\, d\omega.$$

Assuming $\hat{h}$ is even, the second integral vanishes. Integrating by parts yields

$$h^{-1}[j] = \frac{1}{j} \int_{-\pi}^{\pi} r(\omega) sin(\omega j)\, d\omega,$$

where $r(\omega)$ is the derivative of $\frac{1}{\hat{h}(\omega)}$. Integrating by parts again, further assuming $r(\pi) = r(-\pi) = 0$ so that the boundary terms vanish, we get

$$\begin{aligned} h^{-1}[j] &= \frac{1}{j} \int_{-\pi}^{\pi} r(\omega) sin(\omega j)\, d\omega \\ &\leq \frac{1}{j^2} \int_{-\pi}^{\pi} |r'(\omega)|\, d\omega \\ &\leq \frac{C}{j^2}, \end{aligned}$$

where C comes from one more assumption that r is of bounded variation, which means that there exists a constant C satisfying the last inequality. In particular, r is of bounded variation if $\hat{h} \in C^2[-\pi,\pi]$. Combining all these assumptions the sufficient conditions are: $\hat{h} \in C^2[-\pi,\pi], \hat{h} > 0, \hat{h}$ even, and $(\hat{h})'(-\pi) = (\hat{h})'(\pi) = 0$. ■

Problem 3. (#4.7 *p*.151 in [8]) Prove that a scaling function ϕ defined by

$$|\hat{\phi}(\omega)|^2 = \int_{\omega}^{+\infty} \frac{|\hat{\psi}(\xi)|^2}{\xi}\, d\xi,$$

where ψ is a real wavelet, i.e., a normalized zero-average real-valued finite energy function, satisfies $\|\phi\| = 1$.

Solution. Since ψ is real we have

$$\hat{\psi}(\omega)^* = \int_{\mathbb{R}} \{\psi(t)e^{i\omega t}\}^*\, dt = \int_{\mathbb{R}} \psi(t)e^{-i\omega t}\, dt = \hat{\psi}(-\omega),$$

which means $|\hat{\psi}(\omega)|$ is even. Then, by definition above, $|\hat{\phi}(\omega)|^2$ is even. So,

$$\begin{aligned}
\|\hat{\phi}\|^2 &= 2\int_0^\infty |\hat{\phi}(\omega)|^2 \, d\omega \\
&= 2\int_0^\infty \left(\int_\omega^{+\infty} \frac{|\hat{\psi}(\xi)|^2}{\xi} \, d\xi \right) d\omega \\
&= 2\int_0^\infty \left(\int_1^{+\infty} |\hat{\psi}(s\omega)|^2 \frac{ds}{s} \right) d\omega \qquad \text{(by change of variable } \xi = s\omega) \\
&= 2\int_1^{+\infty} \left(\int_0^\infty |\hat{\psi}(s\omega)|^2 \, d\omega \right) \frac{ds}{s} \qquad \text{(by Fubini)} \\
&= \int_1^{+\infty} \left(2\int_0^\infty |\hat{\psi}(\xi)|^2 \frac{d\xi}{s} \right) \frac{ds}{s} \qquad \text{(by change of variable } \xi = s\omega) \\
&= \|\hat{\psi}\|^2 \int_1^{+\infty} \frac{ds}{s^2} \\
&= \|\hat{\psi}\|^2.
\end{aligned}$$

Note the use of Fubini's theorem, which justifies the interchange of the order of integration. The use is valid, in our case, due to the absolute integrability of the double integral in either order. By Plancherel (xi), this gives $\|\phi\| = \|\psi\|$. Since ψ is a wavelet, we have $\|\psi\| = 1$, and we get the result. ■

Problem 4. (#5.9 *p.*201 in [8]) Let $\hat{g} = \mathbf{1}_{[-\omega_0,\omega_0]}$. Prove that $\{h_{n,k}(t)\}_{(n,k)\in\mathbb{Z}^2} := \{\sqrt{\frac{\pi}{\omega_0}} g_{n,k}(t)\}_{(n,k)\in\mathbb{Z}^2} := \{\sqrt{\frac{\pi}{\omega_0}} g(t - \frac{2\pi n}{\omega_0}) e^{ik\omega_0 t}\}_{(n,k)\in\mathbb{Z}^2}$ is an orthonormal basis of $L^2(\mathbb{R})$.

Remark 30. There is a typo in the problem. The normalizing factor $\sqrt{\frac{\pi}{\omega_0}}$ is missing in the text [8]. This is the correct normalizing factor since

$$\|g_{n,k}\|^2 = \|g\|^2 = \frac{1}{2\pi}\|\hat{g}\|^2 = \frac{1}{2\pi} 2\omega_0 = \frac{\pi}{\omega_0}.$$

Solution. First, we begin with a definition.

Definition 31. Let H be $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$. A collection $\{\phi_n\}$ of elements of H is said to be a *frame* of H if there exist constants $0 < A \le B < \infty$ such that for every $f \in H$,

$$A||f||^2 \le \sum |\langle f, \phi_n \rangle|^2 \le B||f||^2.$$

Then A and B are called frame bounds. If $A = B$, then the frame is said to be tight.

Next, we list some relevant results. Proofs of most of these results are omitted here and can be found in [8].

Result 1. : Given a frame $\{\phi_n\}$ of H, there exists a dual frame $\{\tilde{\phi}_n\}$ such that for every $f \in H$,

$$f = \sum \langle f, \phi_n \rangle \tilde{\phi}_n.$$

Thus, a frame completely characterizes a signal in H.

Result 2. : Let $\{\phi_n\}$ be a frame of H with bounds A and B. If $A \leq B \leq 1$, then the frame consists of linearly independent elements. If $A = B = 1$, then the frame forms an orthonormal basis of H.

Result 3. : $\{\phi_n\}$ is a frame with bounds A and B if and only if $\{\hat{\phi}_n\}$ is a frame with bounds A and B.

Proof of Result (3): Let $\{\phi_n\}$ be a frame of $L^2(\mathbb{R})$ with bounds A and B and $g \in L^2(\mathbb{R})$. There exists $f \in L^2(\mathbb{R})$ such that $g = \hat{f}$. Then we have

$$A||f||^2 \leq \sum |\langle f, \phi_n \rangle|^2 \leq B||f||^2.$$

Then using Parseval (xii) and Plancherel (xi), we compute

$$A||f||^2 \leq \sum |\langle f, \phi_n \rangle|^2 \leq B||f||^2$$

$$A \int |f(t)|^2 \, dt \leq \sum \left| \int f(t) \phi_n^*(t) \, dt \right|^2 \leq B \int |f(t)|^2 \, dt$$

$$A \frac{1}{2\pi} \int |\hat{f}(\omega)|^2 \, d\omega \leq \sum \left| \frac{1}{2\pi} \int \hat{f}(\omega) \hat{\phi}_n^*(\omega) \, d\omega \right|^2 \leq B \frac{1}{2\pi} \int |\hat{f}(\omega)|^2 \, d\omega$$

$$A \frac{1}{2\pi} \int |\hat{f}(\omega)|^2 \, d\omega \leq \frac{1}{2\pi} \sum \left| \int \hat{f}(\omega) \left(\frac{1}{\sqrt{2\pi}} \hat{\phi}_n^*(\omega) \right) \, d\omega \right|^2 \leq B \frac{1}{2\pi} \int |\hat{f}(\omega)|^2 \, d\omega$$

$$A \int |\hat{f}(\omega)|^2 \, d\omega \leq \sum \left| \int \hat{f}(\omega) \left(\frac{1}{\sqrt{2\pi}} \hat{\phi}_n^*(\omega) \right) \, d\omega \right|^2 \leq B \int |\hat{f}(\omega)|^2 \, d\omega$$

$$A||g||^2 \leq \sum |\langle g, \frac{1}{\sqrt{2\pi}}\hat{\phi}_n\rangle|^2 \leq B||g||^2$$

Conversely, assume $\{\hat{\phi}_n\}$ is a frame of $L^2(\mathbb{R})$ with bounds A and B, and $f \in L^2(\mathbb{R})$. There exists $g \in L^2(\mathbb{R})$ such that $g = \hat{f}$. Then we have

$$A||g||^2 \leq \sum |\langle g, \frac{1}{\sqrt{2\pi}}\hat{\phi}_n\rangle|^2 \leq B||g||^2.$$

Now, the same computation as the last one, except, we start at the last line and go backward to arrive at the first line, gives the converse result.

Now, we come back to the solution of our problem. Observe that, in light of Results (3) and (2), all we need to do is prove that $\{G_{n,k}(\omega)\}_{(k,n)\in\mathbb{Z}^2} := \{\frac{1}{\sqrt{2\pi}}\hat{h}_{n,k}(\omega)\}_{(k,n)\in\mathbb{Z}^2} = \{\frac{1}{\sqrt{2\omega_0}}\hat{g}_{n,k}(\omega)\}_{(k,n)\in\mathbb{Z}^2}$ is a frame with bounds $A = B = 1$. Since $\{g_{n,k}(t)\}_{(k,n)\in\mathbb{Z}^2} = \{g(t - \frac{2\pi n}{\omega_0})e^{ik\omega_0 t}\}_{(k,n)\in\mathbb{Z}^2}$ and $\hat{g}(\omega) = \mathbf{1}_{[-\omega_0,\omega_0]}(\omega)$, using properties (iv) and (iii) of the Fourier transform, we get $\hat{g}_{n,k}(\omega) = e^{\frac{-2\pi i n\omega}{\omega_0}}\mathbf{1}_{[-\omega_0,\omega_0]}(\omega - k\omega_0)$. Now, for a $f \in L^2(\mathbb{R})$, we compute

$$\begin{aligned}
&\sum_{(n,k)\in\mathbb{Z}^2} \langle f, G_{n,k}\rangle \langle f, G_{n,k}\rangle^* \\
&= \sum_{(n,k)\in\mathbb{Z}^2} \left\langle f, \frac{1}{\sqrt{2\omega_0}}\hat{g}_{n,k}(\omega)\right\rangle \left\langle f, \frac{1}{\sqrt{2\omega_0}}\hat{g}_{n,k}(\omega)\right\rangle^* \\
&= \frac{1}{2\omega_0}\sum_{(n,k)\in\mathbb{Z}^2} \left(\int_{\mathbb{R}} f(\omega)e^{\frac{-2\pi i n\omega}{\omega_0}}\mathbf{1}_{[-\omega_0,\omega_0]}(\omega - k\omega_0)\,d\omega\right)\left(\int_{\mathbb{R}} f(\omega)e^{\frac{-2\pi i n\omega}{\omega_0}}\mathbf{1}_{[-\omega_0,\omega_0]}(\omega - k\omega_0)\,d\omega\right)^* \\
&= \frac{1}{2\omega_0}\sum_{(n,k)\in\mathbb{Z}^2} \left(\int_{\omega_0(-k-1)}^{\omega_0(-k+1)} f(\omega)e^{\frac{-2\pi i n\omega}{\omega_0}}\,d\omega\right)\left(\int_{\omega_0(-k-1)}^{\omega_0(-k+1)} f(\omega)e^{\frac{-2\pi i n\omega}{\omega_0}}\,d\omega\right)^*
\end{aligned}$$

Next, for each $k \in \mathbb{Z}$, define the interval $I_k = [\omega_0(-k-1), \omega_0(-k+1)]$ and observe that these intervals all have length $2\omega_0$, are disjoint except for a set of measure zero and cover all of $\mathbb{R}$. Thus, given a $f \in L^2(\mathbb{R})$, if we denote f_k the restriction of f to I_k, we have

$$\|f\|^2 = \sum_{k\in\mathbb{Z}} \|f_k\|^2_{L^2(I_k)}.$$

Now, using this result and the fact that the n-th Fourier coefficient of f_k is given by the formula

$$\hat{f}_k[n] = \frac{1}{2\omega_0}\int_{\omega_0(-k-1)}^{\omega_0(-k+1)} f(\omega)e^{\frac{-2\pi i n\omega}{\omega_0}}\,d\omega,$$

we continue the earlier computation

$$
\begin{aligned}
&\sum_{(n,k)\in\mathbb{Z}^2} \langle f, G_{n,k}\rangle \langle f, G_{n,k}\rangle^* \\
&= \frac{1}{2\pi} \sum_{(n,k)\in\mathbb{Z}^2} \left(\int_{\omega_0(-k-1)}^{\omega_0(-k+1)} f(\omega) e^{\frac{-2\pi i n\omega}{\omega_0}}\, d\omega \right) \left(\int_{\omega_0(-k-1)}^{\omega_0(-k+1)} f(\omega) e^{\frac{-2\pi i n\omega}{\omega_0}}\, d\omega \right)^* \\
&= \frac{(2\omega_0)^2 \pi}{2\omega_0} \sum_{(n,k)\in\mathbb{Z}^2} \hat{f}_k[n] \hat{f}_k[n]^* \\
&= 2\omega_0 \sum_{(n,k)\in\mathbb{Z}^2} \|\hat{f}_k[n]\|^2 \\
&= 2\omega_0 \frac{1}{2\omega_0} \sum_{k\in\mathbb{Z}} \|f_k\|^2_{L^2(I_k)} \\
&= \|f\|^2.
\end{aligned}
$$

Thus, $\{G_{n,k}(\omega)\}_{(k,n)\in\mathbb{Z}^2} := \{\frac{1}{\sqrt{2\pi}}\hat{h}_{n,k}(\omega)\}_{(k,n)\in\mathbb{Z}^2}$ is a frame with bounds $A = B = 1$. Invoking Results (3) and (2), this completes the solution of our problem. ■

Chapter 4

Estimators in image denoising

In Chapter 2, we introduced Bayesian inference and the Bayes estimator in the context of image data. In this chapter, we will describe a few more estimators that are commonly used in image processing, especially, in image denoising.

4.1 Minimax estimator

Bayesian framework requires us to define the prior probability distribution of image data. However, it may not be possible to have enough information to specify the probability distribution itself but may be possible, instead, to specify a smaller set the data will belong in. The minimax framework restricts the data to be in a prior set Θ and then tries to minimize the maximum risk over Θ. Natural images as such are so diverse that there exists no stochastic model for obtaining their probability distribution. However, for some images satisfying certain regularity conditions, for example, having a bounded total variation, we can reduce the size of the prior set Θ.

The goal is to estimate $F \in \Theta$ from the noisy data

$$X[n] = F[n] + W[n],\ 0 \leq n < N$$

where $W[n]$ is the white Gaussian noise having a normal distribution with mean 0 and variance σ^2, written $W \sim N(0, \sigma^2 I)$. The risk of an estimator $\tilde{F} = DX$ is $r(D, F) =$

$E\{\|DX - F\|^2\}$. Not knowing the probability distribution of signals $F \in \Theta$, we cannot calculate the expected risk over Θ. The maximum risk is

$$r(D, \Theta) = \sup_{F \in \Theta} E\{\|DX - F\|^2\}.$$

Minimizing this maximum risk over all linear and non-linear operators D, we get the minimax risk:

$$r_{nl}(\Theta) = \inf_{D \in \mathcal{O}_{nl}} r(D, \Theta).$$

In practice, we like to get a simple operator D which is easy to implement and whose maximum risk may not achieve the minimax risk but is close enough. For this purpose, we compute the infimum over linear operators only and get the D which achieves the linear minimax risk

$$r_l(\Theta) = \inf_{D \in \mathcal{O}_l} r(D, \Theta).$$

The following theorem shows that a minimax estimator is a Bayes estimator for a "least favorable" prior distribution. The proof given here can be found in Mallat's book [8].

Theorem 32 (Minimax)**.** [8] For any subset Θ of $\mathbb{C}^N$

$$r_l(\Theta) = \sup_{\pi \in \Theta^*} r_l(\pi) \text{ and } r_{nl}(\Theta) = \sup_{\pi \in \Theta^*} r_{nl}(\pi),$$

where Θ^* is the set of all probability distributions of random vectors whose realizations are in Θ.

Proof [8]: Let $\pi \in \Theta^*$. Then

$$r(D, \pi) = E_{W,\pi}\{\|F - DX\|^2\} \leq \sup_{F \in \Theta} E\{\|F - DX\|^2\} = r(D, \Theta). \tag{4.1}$$

Taking supremum over $\pi \in \Theta^*$, we get

$$\sup_{\pi \in \Theta^*} r(\pi) \leq r(D, \Theta).$$

Further, taking infimum over $D \in \mathcal{O}$ where either $\mathcal{O} = \mathcal{O}_{nl}$ or $\mathcal{O} = \mathcal{O}_l$, we get

$$\sup_{\pi \in \Theta^*} r(\pi) \leq r(\Theta). \tag{4.2}$$

Next, we need to show the reverse inequality: $\sup_{\pi\in\Theta^*} r(\pi) \geq r(\Theta)$.

Case I: Θ is finite, say, $\Theta = \{f_i\}_{1\geq i\geq p}$. Define a risk set:

$$R = \{(y_1, ..., y_p) \in \mathbf{C^p} : \exists\, \mathbf{D} \in \mathcal{O} \text{ with } \mathbf{y_i = r(D, f_i)} \text{ for } \mathbf{1 \geq i \geq p}\},$$

which, by convexity of $\mathcal{O}$, is convex in $\mathbf{C^p}$. For any $\pi = (\pi_1, ..., \pi_p) \in \Theta^*$, we have

$$r(D, \pi) = \sum_{i=1}^{p} \pi_i r(D, f_i) = \sum_{i=1}^{p} \pi_i y_i.$$

The equation $\sum_{i=1}^{p} \pi_i y_i = b$ is a hyperplane P_b in $\mathbf{C^p}$. Now, finding the infimum b_0 of all b for which P_b intersects the risk set R, we obtain

$$b_0 = \inf_{D\in\mathcal{O}} r(D, \pi) = r(\pi).$$

On the other hand, let $Q_c = \{(y_1, ..., y_p) \in \mathbf{C^p} : \mathbf{y_i \leq c}\}$. Then, finding the infimum c_0 of all c for which Q_c intersects the risk set R, we obtain

$$c_0 = \inf_{D\in\mathcal{O}} \sup_{f_i\in\Theta} r(D, f_i) = r(\Theta).$$

We seek to prove that $c_0 \leq b_0$. Let $\tilde{Q}_{c_0}$ be the interior of Q_{c_0}. Now, since both $\tilde{Q}_{c_0}$ and R are convex sets and $\tilde{Q}_{c_0} \cap R = \emptyset$, the hyperplane separation theorem says that there exists a hyperplane of equation

$$\sum_{i=1}^{p} \tau_i y_i = \tau.y = b,$$

with $\tau.y \leq b$ for $y \in \tilde{Q}_{c_0}$ and $\tau.y \geq b$ for $y \in R$. Also, each $\tau_i \geq 0$, for if $\tau_j < 0$ we will have a contradiction by letting $y_j \to -\infty$ with all the other coordinates fixed and finding $\tau.y \to +\infty$ for $y \in \tilde{Q}_{c_0}$. Since $ds \sum_{i=1}^{p} \tau_i > 0$ for any non-trivial case, we can without loss of generality assume $ds \sum_{i=1}^{p} \tau_i = 1$ after normalization which makes τ correspond to a probability distribution. Since $\tau.y \leq b$, by letting $y \in \tilde{Q}_{c_0}$ converge to the corner point $(c_0, ..., c_0)$, we obtain that $c_0 < b$. Further, since $\tau.y \leq b$, for all $y \in R$, we get

$$r(\tau) = \inf_{D\in\mathcal{O}} \sum_{i=1}^{p} \tau_i r(D, f_i) = \inf_{D\in\mathcal{O}} \sum_{i=1}^{p} \tau_i y_i \geq b \geq c_0 = r(\Theta).$$

This yields $\sup_{\pi \in \Theta^*} r(\pi) \geq r(\Theta)$.

Case II: Θ is infinite. When $\mathcal{O} = \mathcal{O}_l$ or $\mathcal{O} = \mathcal{O}_{nl}$ for any prior $\pi \in \Theta^*$, we know from Theorems 1 and 2, that there exists a Bayes decision operator $D \in \mathcal{O}$ which attains $\inf\limits_{D \in \mathcal{O}} r(D, \pi)$. We can find a subset of operators and is compact for an appropriate topology. When $\mathcal{O} = \mathcal{O}_l$, $\mathcal{C}$ can be the set of linear operators of norm smaller than 1. Since $\mathcal{C}$ belongs to a finite dimensional space of linear operators, it is compact . Furthermore, the risk $r(f, D)$ is also continuous in this topology with respect to $D \in \mathcal{C}$.

Let $c < r(\Theta)$. For any $f \in \Theta$, set $\mathcal{S}_f = \{D \in \mathcal{C} : r(D, f) > c\}$. Since $r(D, f)$ is continuous, $\mathcal{S}_f$ is an open set. Since for each $D \in \mathcal{C}$,there exists $f \in \Theta$, with $D \in \mathcal{S}_f$,we have $\mathcal{C} = \cup_{f \in \Theta} \mathcal{S}_f$. By compactness of $\mathcal{C}$, we get $\mathcal{C} = \cup_{i=1}^{p} \mathcal{S}_{f_i}$. The minimum risk over $\Theta_c = \{f_i\}_{1 \leq i \leq p}$ satisfies $r(\Theta_c) = \inf\limits_{D \in \mathcal{O}} \sup\limits_{1 \leq i \leq p} r(D, f_i) \geq c$. Now, for Θ_c which is a finite set, we have from case I that there exists $\tau_c \in \Theta_c^* \subseteq \Theta^*$ such that $r(\tau_c) = r(\Theta_c)$. Next, letting c go to $r(\Theta)$, we get $\sup\limits_{\pi \in \Theta^*} r(\pi) \geq r(\Theta)$. ■

4.2 Diagonal estimator

Definition 33. [Covariance] The *covariance* between two random variables is defined by

$$Cov(X_1, X_2) = E\{(X_1 - E(X_1))(X_2 - E(X_2))^*\}.$$

First, we will establish a few definitions regarding a random vector Y of size N, which is a finite one-dimensional signal. A natural and somewhat complicated extension of these will give us their analogs for a random image of size $N \times N$, which is a two-dimensional discrete signal with period N.

Definition 34. [Covariance matrix] A *covariance matrix of a random vector* Y of size N is the $N \times N$ matrix with entries $R_Y[n, m] = Cov(Y[n], Y[m])$.

Definition 35. [Covariance operator] A *covariance operator* K on $\mathbb{R}^N$ associated with a random vector Y of size N is defined by

$$Kh[n] = \sum_{m=0}^{N-1} R_Y[n,m]h[m],\ 0 \leq N.$$

If h and g are two random vectors of size N,

$$\begin{aligned}\langle Kg, h\rangle &= Cov(\langle Y, h\rangle, \langle Y, g\rangle)\\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} R_Y[n,m]g[m]h^*[n]\\ &= \sum_{m=0}^{N-1}\left(\sum_{n=0}^{N-1} R_Y[n,m]h[n]\right)^* g[m]\\ &= \langle g, K^*h\rangle\end{aligned}$$

Thus, $R_Y[n,m] = R_Y^*[m,n]$ implies that the covariance operator K is self-adjoint. Moreover, it is positive since

$$\langle Kh, h\rangle = E\{|\langle Y, h\rangle|^2\} \geq 0.$$

Hence, appealing to a standard result in Linear Algebra, there exists an orthogonal basis $\{g_k\}_{0\leq k<N}$ such that $Kg_k = \sigma_k^2 g_k$. Then K will be a diagonal matrix in this basis. This basis is called a *Karhunen-Loève basis* of Y. The basis vectors g_k are called the *principal directions.* The eigenvalues are the variances $\sigma_k^2 = \langle Kg_k, g_k\rangle = E\{|\langle Y, g_k\rangle|^2\}$.

Definition 36. [Wide-sense stationary] Y is said to be *wide-sense stationary* if the correlation between any two points depends only on the distance between these points, i.e.,

$$E(Y[n]Y^*[m]) = R[n,m] = R_Y[n-m].$$

Then $Kh[n] = \sum_{k=0}^{N-1} R_Y[n-k]h[k]$ is a convolution whose kernel $R_Y[k]$ is defined for $-N < k < N$.

Definition 37. [Circular stationary] A wide-sense stationary vector Y is called *circular stationary* if $R_Y[n]$ is N-periodic, i.e.,

$$R_Y[n] = R_Y[N+n],\ -N \leq n \leq 0.$$

If $y[n]$ is wide-sense stationary, a periodic extension of it on $\mathbb{Z}$ will be circular stationary. Then the covariance operator K associated with it is a discrete circular convolution.

The eigenvalues of K which are called the *power spectrum,* are

$$\sigma_k^2 = \hat{R}_Y[k] = \sum_{n=0}^{N-1} R_Y[n] exp\left(\frac{-i2k\pi n}{N}\right)$$

and the corresponding eigenvectors are

$$g_k[n] = \frac{1}{\sqrt{n}} exp\left(\frac{i2k\pi n}{N}\right), \; 0 \leq k < N.$$

The following theorem provides us with a diagonal linear estimator in the case when the covariance matrices of the signal F and of the noise W are diagonal in the same Karhunen-Loève basis $\mathcal{B} = \{g_m\}_{0 \leq m < N}$. Denote

$$X_{\mathcal{B}}[m] = \langle X, g_m \rangle, \; \tilde{F}_{\mathcal{B}}[m] = \left\langle \tilde{F}, g_m \right\rangle,$$

$$F_{\mathcal{B}}[m] = \langle F, g_m \rangle, \; W_{\mathcal{B}}[m] = \langle W, g_m \rangle,$$

$$\beta_m^2 = E\{|F_{\mathcal{B}}[m]|^2\} \text{ and } \sigma_m^2 = E\{|W_{\mathcal{B}}[m]|^2\}.$$

Theorem 38. [Weiner][8] If there exists a Karhunen-Loève basis $\mathcal{B} = \{g_m\}_{0 \leq m < N}$ that diagonalizes the covariance matrices of both F and W, then the Weiner estimator is

$$\tilde{F} = \sum_{n=0}^{N-1} \frac{\beta_m^2}{\beta_m^2 + \sigma_m^2} X_{\mathcal{B}}[m] g_m$$

and the resulting minimum linear Bayes risk is

$$r_l(\pi) = \sum_{n=0}^{N-1} \frac{\beta_m^2 \sigma_m^2}{c + \sigma_m^2}.$$

4.3 Oracle estimator

A diagonal estimator of F is given by

$$\tilde{F} = DX = \sum_{m=0}^{N-1} a_m(X_{\mathcal{B}}[m]) X_{\mathcal{B}}[m] g_m.$$

and we must have $|a_m| \leq 1$ to attenuate the noisy coefficients.

Recall that a diagonal estimator is linear when $a_m(X_{\mathcal{B}}[m])$ is constant. For an oracle estimator, we restrict the value of a_m to be either 1 or 0.

An oracle estimator F is given by

$$\tilde{F} = DX = \sum_{m\in\Lambda_\sigma} X_{\mathcal{B}}[m]g_m,$$

where $\Lambda_\sigma = \{0 \leq m < N : |F_{\mathcal{B}}[m]| \geq \sigma\}$. This estimator is impossible to implement since a_m depends on the unknown signal f rather than on the observed signal X.

The resulting risk is

$$r_{pr}(F) = E\{\|F - \tilde{F}\|^2\} = \sum_{m=0}^{N-1} min(|F_{\mathcal{B}}[m]|^2, \sigma^2).$$

4.4 Thresholding estimator

A thresholding estimator is a diagonal estimator where a_m are thresholding functions and the coefficients $a_m(X_{\mathcal{B}}[m])X_{\mathcal{B}}[m]$ of the basis elements g_m are zeroed out if these coefficients are less than a certain value T, called a threshold.

Thus, a thresholding estimator is given by

$$\tilde{F} = DX = \sum_{m=0}^{N-1} \rho_T(X_{\mathcal{B}}[m])g_m,$$

where $\rho_T(x)$ is called the thresholding function.

A hard- or soft-thresholding estimator is $\tilde{F} = DX = \sum_{m=0}^{N-1} \rho_T(X_{\mathcal{B}}[m])g_m$, where

$$\rho_T(x) = \begin{cases} x & \text{if } |x| > T \\ 0 & \text{if } |x| \leq T. \end{cases}$$

for a hard thresholding and

$$\rho_T(x) = \begin{cases} x - T & \text{if } x \geq T \\ x + T & \text{if } x \leq T \\ 0 & \text{if } |x| \leq T. \end{cases}$$

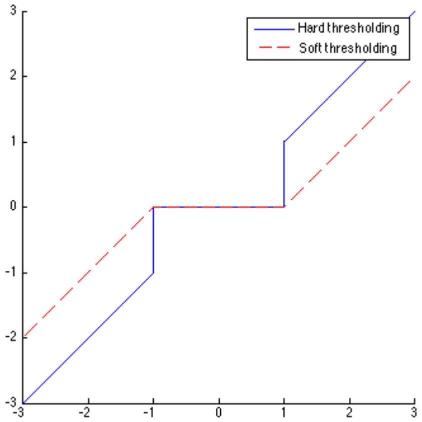

Figure 4.1: *Soft and Hard thresholding with Threshold=1.*

for a soft thresholding. This is visualized in Figure (4.1) (source:http://www.ceremade.dauphine.fr/-peyre/numerical-tour/tours/denoising-wavelet-1d/index-02.png).

The risk of a thresholding estimator is

$$r_{th}(F) = r(D,F) = \sum_{m=0}^{N-1} E\{|F_{\mathcal{B}}[m] - \rho_T(X_{\mathcal{B}}[m])|^2\}.$$

When W is a white noise, $W_{\mathcal{B}}$ is a vector of N independent Gaussian random variables of variance σ^2. Then the threshold is usually chosen to be $T = \sigma\sqrt{2\ln N}$. This is following the principle that T must be chosen to be just above the maximum level of the noise and that [1] the maximum amplitude of the noise has a very high probability of being just below $T = \sigma\sqrt{2\ln N}$:

$$\lim_{N\to\infty} pr\left(T - \frac{\sigma \ln\ln N}{\ln N} \leq \max_{0\leq m<N} |W_{\mathcal{B}}[m]| \leq T\right) = 1.$$

Since a soft thresholding often produces a risk larger than a hard thresholding, we use a threshold $T = \frac{1}{2}\sigma\sqrt{2\ln N}$ with a soft-thresholding to obtain nearly the same risk.

Theorem 39. [8][Donoho, Johnstone] Let $T = \sigma\sqrt{2\ln N}$. The hard- or soft-thresholding estimator risk $r_{th}(F)$ satisfies for all $N \geq 4$,

$$r_{th}(F) \leq (2\ln N + 1)(\sigma^2 + r_{pr}(F)).$$

The factor $2\ln N$ is optimal among diagonal estimators in $\mathcal{B}$:

$$\lim_{N\to\infty} \inf_{D\in\mathcal{O}_d} \sup_{F\in\mathbb{C}^{\mathbb{N}}} \frac{\{E\|F-\tilde{F}\|^2\}}{\sigma^2+r_{pr}(F)} \frac{1}{2\ln N} = 1,$$

where $\mathcal{O}_d$ is the set of all linear or non-linear operators that are diagonal in $\mathcal{B}$.

When the noise is not white, variance differs for each basis element and we have $\sigma_{\mathcal{B}}[m]^2 = E\{|W_{\mathcal{B}}[m]^2|\}$. We, however, still suppose that the noise $W[n]$ has mean zero. In this case, an oracle estimator is

$$\tilde{F} = DX = \sum_{m\in\Lambda_{\sigma_{\mathcal{B}}}} X_{\mathcal{B}}[m]g_m,$$

where $\Lambda_{\sigma_{\mathcal{B}}} = \{0 \leq m < N : |F_{\mathcal{B}}[m]| \geq \sigma_{\mathcal{B}}[m]\}$ and its risk is

$$r_{pr}(F) = E\{\|F-\tilde{F}\|^2\} = \sum_{m=0}^{N-1} \min(|F_{\mathcal{B}}[m]|^2, \sigma_{\mathcal{B}}[m]^2).$$

A hard- or soft-thresholding estimator in the case of a colored noise is given by

$$\tilde{F} = DX = \sum_{m=0}^{N-1} \rho_{T_m}(X_{\mathcal{B}}[m])g_m,$$

where the threshold T_m varies as a function of m.

Theorem 40. [8][Donoho, Johnstone] A hard- or soft-thresholding estimator with $T_m = \sigma_{\mathcal{B}}[m]\sqrt{2\ln N}$ for $0 \leq m < N$ has a risk $r_{th}(F)$ that satisfies, for any $N \geq 4$,

$$r_{th}(F) \leq (2\ln N + 1)\big(\bar{\sigma}^2 + r_{pr}(F)\big),$$

where $\bar{\sigma}^2 = N^{-1}\sum_{m=0}^{N-1}\sigma_{\mathcal{B}}[m]^2$.

Chapter 5

Wavelet Transform

5.1 Multiresolution Analysis (MRA)

We dedicate this section to understanding the most important concept of image processing studied in our report, namely, Multiresolution Analysis. It is the basis of Fast Wavelet transform algorithm, which is a computational method for obtaining wavelet decomposition of signals. We begin with its abstract definition but eventually see how it is simplified when it is completely characterized by some discrete filter.

Definition 41 (Riesz Basis)**.** Let H be $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$. A basis $\{\phi_n\}$ of H is called a *Riesz basis* if there exist constants $0 < A \leq B < \infty$ such that for every $f \in H$,

$$A||f||^2 \leq \sum |\langle f, \phi_n \rangle|^2 \leq B||f||^2.$$

A and B are called Riesz bounds of this basis.

Note that $\langle f, \phi_n \rangle$ are decomposition coefficients of f. Hence,

$$f = \sum \langle f, \phi_n \rangle \phi_n.$$

Then

$$||f||^2 := \langle f, f \rangle = \sum_m \sum_n \langle f, \phi_m \rangle \langle f, \phi_n \rangle \langle \phi_m \phi_n \rangle.$$

If $\{\phi_n\}$ is an orthonormal basis then we get the equality

$$||f||^2 = \sum |\langle f, \phi_n \rangle|^2.$$

Thus, a Riesz basis is weaker than an orthonormal basis. In other words, an orthonormal basis is a Riesz basis with Riesz bounds $A = B = 1$.

Definition 42 (MRA)**.** A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ is an *MRA* if the following six properties hold:

$$\forall (j,k) \in \mathbb{Z}^2,\ f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j, \tag{4.1}$$

$$\forall j \in \mathbb{Z},\ V_{j+1} \subseteq V_j, \tag{4.2}$$

$$\forall j \in \mathbb{Z},\ f(t) \in V_j \Leftrightarrow f\left(\frac{t}{2}\right) \in V_{j+1}, \tag{4.3}$$

$$\lim_{j\to+\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}, \tag{4.4}$$

$$\lim_{j\to-\infty} V_j = \text{Closure}\left(\bigcup_{j=-\infty}^{+\infty} V_j\right) = L^2(\mathbb{R}), \tag{4.5}$$

and there exists θ such that $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of V_0.

$L^2(\mathbb{R})$ is the space of analog finite energy signals. An MRA is a nested sequence of subspaces of $L^2(\mathbb{R})$ satisfying a number of properties. These subspaces are approximations of signals at dyadic resolution levels. For example, the orthogonal projection of $f \in L^2(\mathbb{R})$ onto V_j, denoted by $P_{V_j} f$ and computed by minimizing $\|f - f_j\|$ over all of $f_j \in V_j$, is the approximation of f at resolution 2^{-j}. With this understanding, the properties in the definition are only natural. Property (4.2) specifies that coarser resolution approximations are included in finer resolution approximations. The intuition of properties (4.4) and (4.5) is that reducing the resolution eventually loses the whole signal and, on the other hand, increasing the resolution eventually recovers it wholly. Finally, properties (4.1) and (4.3) puts the information on resolution in a dyadic setting by making sure all translations of dyadic multiples are within the same resolution and by specifying that a transition from one resolution to another is through a dyadic dilation. In fact, denoting $\theta_{j,n}(t) = \frac{1}{\sqrt{2^j}}\theta\left(\frac{t}{2^j} - n\right)$, the family $\{\theta_{j,n}\}_{n\in\mathbb{Z}}$ is a Riesz basis of V_j for all $j \in \mathbb{Z}$. Note the inverse relation that the scale 2^j corresponds to the resolution 2^{-j}.

The following proposition sets the criterion for a $\theta \in H$ to form a Riesz basis of H.

Proposition 43. Let H be $L^2(\mathbb{R})$ or $\ell^2(\mathbb{Z})$. Then $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of H if and only if there exist constants $0 < A \leq B < \infty$ such that for every $\omega \in [-\pi, \pi]$,

$$A \leq \sum |\hat{\theta}(\omega + 2k\pi)|^2 \leq B.$$

With the above proposition in mind, the abstract construction of an MRA boils down to constructing a θ satisfying Proposition 43.

Once we have an MRA with a Riesz basis, we would like to go one better and construct an MRA with an orthonormal basis. Just like the θ in an MRA with a Riesz basis, there is a corresponding function ϕ in an MRA with an orthonormal basis which we call the scaling function of the MRA.

Definition 44. [Scaling function] Let $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA with $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ as a Riesz basis of V_0. A function ϕ whose Fourier transform satisfies

$$\hat{\phi}(\omega) = \frac{\hat{\theta}(\omega)}{\left(\sum_{k=-\infty}^{+\infty} |\hat{\theta}(\omega + 2k\pi)|^2\right)^{1/2}}. \tag{4.6}$$

is called the *scaling function.*

Definition (44) provides the recipe for the construction of the scaling function when we have an MRA with a Riesz basis. Theorem 45 specifies ϕ as an orthonormal counterpart of θ. In particular, $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_0. The proof is taken from Stephen Mallat's book [8].

Theorem 45. [8] Let $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA with a Riesz basis $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ of V_0 and ϕ be the corresponding scaling function. Let us denote

$$\phi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\phi\left(\frac{t}{2^j} - n\right).$$

The family $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_j for all $j \in \mathbb{Z}$.

Proof [8]: First note that $\phi_{j,n}$ is a dilated and shifted version of ϕ in such a way that the L^2-norm or the total energy is conserved. The indices j and n are scale and location parameters respectively. Figure (5.1) (source: www-ssc.igpp.ucla.edu/.../russell/ESS265/Ch8/) illustrates this scaling phenomenon of a function.

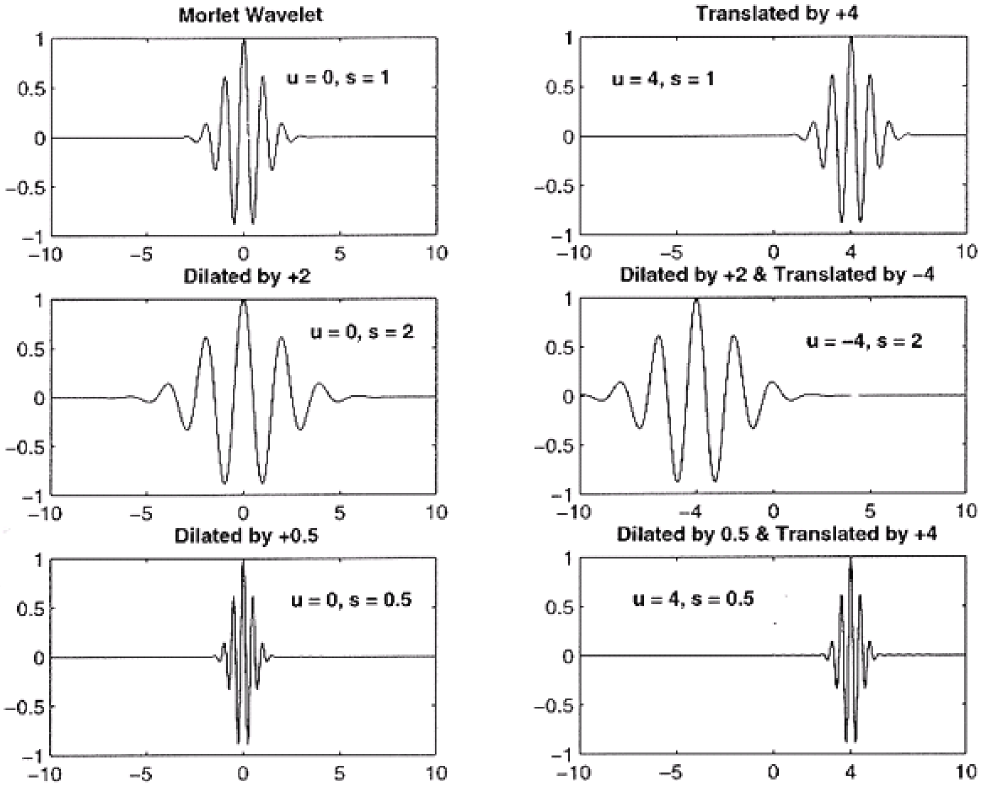

Figure 5.1: *The original function and its dilations and translations.*

We begin with a $\phi \in V_0$ and find a condition for ϕ so that $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ forms an orthonormal basis of V_0. Then, $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$, which are the scaled versions of $\{\phi(t-n)\}_{n\in\mathbb{Z}}$, will be an orthonormal basis of V_j for all $j \in \mathbb{Z}$ due to property (4.3) of an MRA and the fact that scaling preserves orthonormality.

Note that $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_0 if and only if

$$\begin{aligned}\delta[p-n] &= \langle \phi(t-n), \phi(t-p)\rangle \\ &= \int_{\mathbb{R}} \phi(t-n)\phi^*(t-p)\, dt \\ &= \phi \star (\phi^*)(p-n).\end{aligned}$$

Replacing $(p-n)$ by n, we get $\phi\star(\phi^*)(n) = \delta[n]$. Using property (x), the Fourier transform of $\phi\star(\tilde{\phi})(t), t\in\mathbb{R}$ is $|\hat{\phi}(\omega)|^2$, where $\tilde{\phi}(t) = \phi^*(-t)$. But by Theorem 23, the Fourier transform of $\phi\star(\tilde{\phi})(n), n\in\mathbb{Z}$ is periodic with period 1. So, we obtain

$$\sum_{k=-\infty}^{+\infty} |\hat{\phi}(\omega+2k\pi)|^2 = 1, \tag{4.7}$$

which is the necessary and sufficient condition for ϕ so that $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ forms an orthonormal basis of V_j for all $j\in\mathbb{Z}$. Now, since $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ is a basis of V_0, any $\phi\in V_0$ is determined by its decomposition coefficients $a[n]$ in terms of basis elements:

$$\phi(t) = \sum_{n=-\infty}^{+\infty} a[n]\theta(t-n).$$

Taking the Fourier transform of both sides and using property (vii) of the Fourier transform, we obtain

$$\hat{\phi}(\omega) = \hat{a}(\omega)\hat{\theta}(\omega).$$

Now choosing

$$\hat{a}(\omega) = \left(\sum_{k=-\infty}^{+\infty} |\hat{\phi}(\omega+2k\pi)|^2\right)^{-\frac{1}{2}},$$

ϕ satisfies the necessary and sufficient condition (4.7) and we are done. ■

The advantage of having an orthonormal basis instead of a Riesz basis is that it allows us to go from one approximation space to the next more easily via a discrete filter $h[n]$. Since $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_0 and $2^{-1/2}\phi(t/2)\in V_1\subseteq V_0$, there exist coefficients $h[n]$ for us to have the representation

$$\frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} h[n]\phi(t-n).$$

Taking the Fourier transform of both sides yields

$$\hat{\phi}(\omega) = \frac{1}{\sqrt{2}}\hat{h}\left(\frac{\omega}{2}\right)\hat{\phi}\left(\frac{\omega}{2}\right). \tag{4.8}$$

Also, by employing the orthonormality of $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ in the above representation, we obtain the following formula for $h[n]$:

$$h[n] = \langle 2^{-1/2}\phi(t/2), \phi(t-n)\rangle.$$

Since the transfer function $\hat{h}(\omega)$ is supported near the origin, $h[n]$ is called the *low-pass filter* of an MRA. We shall see later that $h[n]$ is one of the fundamental computational tools of Fast Wavelet Transform Algorithm.

The following key theorem lets us go from a scaling function to a low-pass filter and vice versa. Its importance is clear as it makes possible the abstract construction of an MRA out of a simple discrete filter satisfying certain properties.

The necessary part of Theorem 46 says that the low-pass filter obtained through a scaling function must satisfy properties (4.9) and (4.10). The sufficiency part guarantees that a discrete filter $h[n]$ whose Fourier transform is regular enough and which, in addition to the above two necessary properties, satisfies property (4.11) is a low-pass filter of an MRA and the equation (4.12) provides a formula for the the scaling function ϕ of this MRA.

Theorem 46. [Mallat,Meyer[7][9][8]] Let $\phi \in L^2(\mathbb{R})$ be an integrable scaling function. The Fourier series of $h[n] = \langle 2^{-1/2}\phi(t/2), \phi(t-n)\rangle$ satisfies

$$\forall\omega \in \mathbb{R}, |\hat{h}(\omega)|^2 + |\hat{h}(\omega+\pi)|^2 = 2, \tag{4.9}$$

and

$$\hat{h}(0) = \sqrt{2}. \tag{4.10}$$

Conversely, if $\hat{h}(\omega)$ is 2π periodic and continuously differentiable in a neighborhood of $\omega = 0$, if it satisfies (4.9) and (4.10) and if

$$\inf_{\omega\in[-\pi/2,\pi/2]} |\hat{h}(\omega)| > 0, \tag{4.11}$$

then

$$\hat{\phi}(\omega) = \prod_{p=1}^{+\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}} \tag{4.12}$$

is the Fourier transform of a scaling function $\phi \in L^2(\mathbb{R})$.

Definition 47 (Conjugate Mirror Filter)**.** A discrete filter $h[n]$ whose transfer function $\hat{h}(w)$ satisfies the condition

$$\forall \omega \in \mathbb{R}, |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2,$$

is called a *conjugate mirror filter.*

We have seen that an MRA is a sequence of subspaces $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ that are approximations of signals at dyadic resolution levels 2^{-j}. When we make approximations, we lose details. Thus, details are complementary to approximations. Naturally, we should be able to find a sequence of subspaces $\{W_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ which provides the details of signals at each dyadic resolution level. This is achieved in the following way. V_j, the approximations of signals at resolution 2^{-j} is a subspace of V_{j-1}, the approximations at a coarser resolution 2^{-j+1}. Letting W_j be the orthogonal complement of V_j in V_{j-1}, the subspace W_j will provide details of signals that is lost by going from a finer resolution 2^{-j} to a coarser resolution 2^{-j+1}. Thus, we have

$$V_{j-1} = V_j \oplus W_j, \forall j \in \mathbb{Z}.$$

Let us stop for a moment and review what we have seen so far. First, we saw that a scaling function $\phi \in V_0$ is the building block of an MRA whose dilations and translations form an orthonormal basis of the approximation spaces V_j's. Next, we saw that there is a discrete filter $h[n]$ associated to an MRA which can equivalently build an MRA and, in particular, its corresponding scaling function. Finally, we have established W_j's as detail spaces of an MRA which are spaces complementary to approximation spaces. In order to see where we are heading, let us now consider a hypothetical situation.

Just as we have the orthonormal basis $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ of V_j for all $j \in \mathbb{Z}$ generated by translating and dilating the scaling function $\phi \in V_0$, it would be nice to have a function $\psi \in W_0$

which would generate an orthonormal basis $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ of W_j for all $j \in \mathbb{Z}$. And, like a conjugate mirror filter $h[n]$ which completely specifies the scaling function ϕ, we would similarly like to have a mirror filter $g[n]$ which could specify ψ completely. Continuing this thought, if we had such a $\psi \in W_0$ in hand, the relation between the function ψ and the filter $g[n]$ would be established as follows. Scaling ψ by 2^{-1}, we would have $2^{-1/2}\psi(t/2) \in W_1 \subseteq V_0$. Since $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_0, there would exist decomposition coefficients $g[n]$ such that

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} g[n]\phi(t-n).$$

Using orthonormality of $\{\phi(t-n)\}_{n\in\mathbb{Z}}$, we would get the following formula for the mirror filter $g[n]$ in terms of the orthonormal wavelet ψ.

$$g[n] = \left\langle 2^{-1/2}\psi(t/2), \phi(t-n)\right\rangle.$$

Theorem (48) below realizes the hypothetical situation we just described here and provides a formula to compute a function ψ, called the mother wavelet associated to an MRA, whose dilations and translations will indeed form an orthonormal basis of the detail spaces. The discrete filter $g[n]$ obtained in this process is called the *high-pass filter* of the MRA since its transfer function $\hat{g}(\omega)$ is supported away from the origin. The low-pass filter $h[n]$ and the high-pass filter $g[n]$, as we shall see later, are going to be the two fundamental computational tools of Fast Wavelet Transform Algorithm. Figure (5.2) (source: commons.wikimedia.org) illustrates the transfer functions of low- and high-pass filters. The proof of the theorem is adapted from the proof in Stephen Mallat's text [8].

Theorem 48. [Mallat,Meyer[7][8][9]] Let ϕ be a scaling function and h the corresponding conjugate mirror filter. Let ψ be the function having a Fourier transform

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2}}\hat{g}\left(\frac{\omega}{2}\right)\hat{\phi}\left(\frac{\omega}{2}\right), \tag{4.13}$$

with

$$\hat{g}(\omega) = e^{-i\omega}\hat{h}^*(w+\pi). \tag{4.14}$$

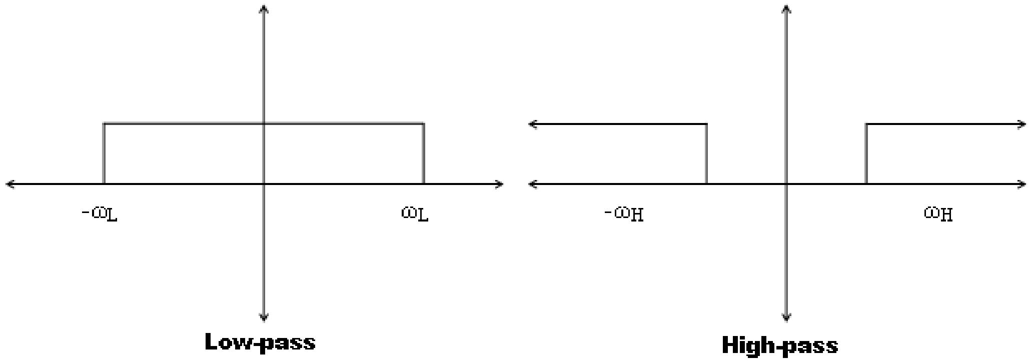

Figure 5.2: *The transfer functions of the low- and high-pass filters.*

Let us denote

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j n}{2^j}\right).$$

For any scale 2^j, $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of W_j. For all scales, $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ is an orthonormal basis of $L^2(\mathbb{R})$.

Proof [8]: First note that the theorem tells us how to construct an orthonormal wavelet ψ, given an MRA with a scaling function ϕ and the corresponding conjugate mirror filter $h[n]$. The following lemma sets necessary and sufficient conditions on $\hat{g}$, which is clearly satisfied by $\hat{g}$ given by the formula (4.14) in the theorem, so that $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ constructed out of $\hat{\psi}$ given by the formula (4.13) in the theorem will be an orthonormal basis of W_j.

Lemma 49. The family $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of W_j if and only if

$$|\hat{g}(\omega)|^2 + |\hat{g}(\omega+\pi)|^2 = 2 \tag{4.15}$$

and

$$\hat{g}(\omega)\hat{h}^*(\omega) + \hat{g}(\omega+\pi)\hat{h}^*(\omega+\pi) = 0. \tag{4.16}$$

Proof of the lemma: Since the result can be extended to arbitrary j with an appropriate scaling, we only need to show that the conditions (4.15) and (4.16) hold if and only if $\{\psi(t-n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of W_0, which, by the orthonormality criterion (4.7) is true if and only if

$$I(\omega) = \sum_{k=-\infty}^{+\infty} |\hat{\psi}(\omega + 2k\pi)|^2 = 1. \tag{4.17}$$

This is true for all ω. Hence, using (4.13), 2π-periodicity of $\hat{g}(\omega)$ and the orthonormal criterion (4.7) for ϕ, we equivalently get (4.15):

$$\begin{aligned} 2 &= \sum_{k=-\infty}^{+\infty} |\hat{g}(\omega + k\pi)|^2 |\hat{\phi}(\omega + k\pi)|^2 \\ &= |\hat{g}(\omega)|^2 \sum_{p=-\infty}^{+\infty} |\hat{\phi}(\omega + 2p\pi)|^2 + |\hat{g}(\omega + \pi)|^2 \sum_{p=-\infty}^{+\infty} |\hat{\phi}(\omega + \pi + 2p\pi)|^2 \\ &= |\hat{g}(\omega)|^2 + |\hat{g}(\omega + \pi)|^2. \end{aligned}$$

W_0 is orthogonal to V_0 if and only if $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ and $\{\psi(t-n)\}_{n\in\mathbb{Z}}$ are orthogonal families of vectors. This is true if and only if $\langle \psi(t), \phi(t-n)\rangle = \psi \star (\phi^*)(n) = 0$, for all $n \in \mathbb{Z}$. Taking the fourier transform using property (x) and Theorem 23, we get $\forall \omega$,

$$\sum_{k=-\infty}^{+\infty} \hat{\psi}(\omega + 2k\pi)\hat{\phi}^*(\omega + 2k\pi) = 0.$$

As before, by inserting expressions (4.13) for $\hat{\psi}$ and (4.8) for $\hat{\phi}$ and using the orthonormality criterion (4.7) for ϕ, we prove that the above condition is equivalent to (4.16). Now, the only thing we need to verify is that $V_{-1} = V_0 \oplus W_0$. Given any $a[n] \in \ell^2(\mathbb{Z})$, define $b[n] \in \ell^2(\mathbb{Z})$ and $c[n] \in \ell^2(\mathbb{Z})$ by

$$\hat{b}(2\omega) = \frac{1}{2}[\hat{a}(\omega)\hat{h}^*(\omega) + \hat{a}(\omega + \pi)\hat{h}^*(\omega + \pi)]$$

and

$$\hat{c}(2\omega) = \frac{1}{2}[\hat{a}(\omega)\hat{g}^*(\omega) + \hat{a}(\omega + \pi)\hat{g}^*(\omega + \pi)].$$

Using (4.15), (4.16) and (4.9), we verify that

$$\hat{a}\left(\frac{\omega}{2}\right) = \hat{b}(\omega)\hat{h}\left(\frac{\omega}{2}\right) + \hat{c}(\omega)\hat{g}\left(\frac{\omega}{2}\right).$$

Due to the expressions (4.13) for $\hat{\psi}$ and (4.8) for $\hat{\phi}$, this is equivalent to

$$\frac{1}{\sqrt{2}}\hat{a}\left(\frac{\omega}{2}\right)\hat{\phi}\left(\frac{\omega}{2}\right) = \hat{b}(\omega)\hat{\phi}(\omega) + \hat{c}(\omega)\hat{\psi}(\omega).$$

Taking the inverse Fourier transform yields

$$\sum_{n=-\infty}^{+\infty} a[n]\sqrt{2}\phi(2[t-2^{-1}n]) = \sum_{n=-\infty}^{+\infty} b[n]\sqrt{2}\phi(t-n) + \sum_{n=-\infty}^{+\infty} c[n]\sqrt{2}\psi(t-n).$$

Thus, we have expressed $\{\sqrt{2}\phi(2t-n)\}_{n\in\mathbb{Z}}$, an orthonormal basis of V_{-1} in terms of $\{\phi(t-n)\}_{n\in\mathbb{Z}}$, an orthonormal basis of V_0 and $\{\psi(t-n)\}_{n\in\mathbb{Z}}$, an orthonormal basis of W_0, which verifies $V_{-1} = V_0 \oplus W_0$. This proves the lemma.

Now, to complete the proof of the theorem, we need to show the collection of all orthonormal bases of $W_j, j \in \mathbb{Z}$ will make up an orthonormal basis of the whole $L^2(\mathbb{R})$. For any $j < l$, W_j is orthogonal to V_j and since $W_l \subseteq V_{l-1} \subseteq V_j$, $W_j, j \in \mathbb{Z}$ are orthogonal. Since $V_{k-1} = W_k \oplus V_k$, we get for any $j < l$,

$$V_l = \oplus_{k=l-1}^{j} W_k \oplus V_j.$$

Letting l go to $-\infty$ and j go to $+\infty$, we get

$$L^2(\mathbb{R}) = \oplus_{k=-\infty}^{+\infty} W_k,$$

due to MRA properties (4.4) and (4.5). Hence the theorem is proved. ■

Finally, it is noteworthy how the two conjugate mirror filters, fundamental computational tools of the so-called Fast Wavelet Transform Algorithm which will be described shortly in Section 5.2, are related with one another. Taking the inverse Fourier transform of (4.14), we get the following:

$$g[n] = (-1)^{1-n} h[1-n].$$

5.2 Wavelets

Recall that our theory of MRA began with just a scaling function ϕ satisfying certain properties in hand. Then we went on to establish that an MRA can be built alternatively from a low-pass filter $h[n]$ or from another corresponding conjugate mirror filter $g[n]$, called the high-pass filter. Then we developed a formula to compute the mother wavelet function

ψ associated with this MRA. We got various equations relating the scaling function ϕ, the mother wavelet ψ, the low-pass filter $h[n]$ and the high-pass filter $g[n]$ with one another.

Definition 50. [Wavelet] A *wavelet* is a function ψ whose dilations and translations $\{\psi_{j,n}\}$ with dilation parameter $-\infty < j < \infty$ and location parameter $-\infty < n < \infty$ defined as

$$\psi_{j,n}(t) := \frac{1}{\sqrt{2^j}}\psi\left(\frac{t - 2^j n}{2^j}\right)$$

form an orthonormal basis of $L^2(\mathbb{R})$.

Figure (5.3) (source: http://www.satmagazine.com/cgi-bin/display-image.cgi?1114554895) illustrates some common wavelets.

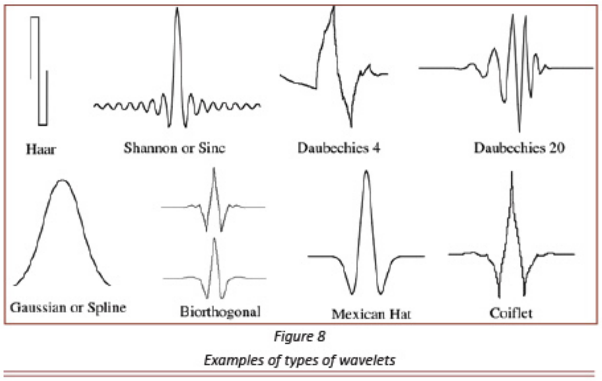

Figure 5.3: *Some common wavelets.*

The function ψ obtained in Theorem 48 is thus a wavelet, called the mother wavelet of the MRA. Note that if ϕ is a scaling funtion, then $\{\phi_{j,n}\}_{-\infty<j,n<\infty}$ also forms an orthonormal basis of $L^2(\mathbb{R})$. Thus, the scaling function ϕ is alternatively called the father wavelet of the MRA. A wavelet has two main properties, namely, it has a mean zero and it is localized. So, it is zero everywhere except in a small domain where it has equal positive and negative values for cancelation.

Summing up once again, an MRA is a nested sequence of approximation spaces $\{V_j\}_{j\in\mathbb{Z}}$ of the signal space $L^2(\mathbb{R})$ and that, complementary to these spaces, there are detail spaces $\{W_j\}_{j\in\mathbb{Z}}$ associated to an MRA. For each fixed j, we have $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ (built by dilating and translating the mother wavelet ψ) and $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ (built by dilating and translating the father wavelet or the scaling function ϕ) forming orthonormal bases of V_j and W_j respectively. We further have that the spaces V_j and W_j are completely characterized by either one of the mirror filters $h[n]$ and $g[n]$.

Fast Wavelet Transform Algorithm consists of a method of decomposing successively each approximation $P_{V_j}f$ into $P_{V_{j+1}}f$,a coarser approximation of f and $P_{W_{j+1}}f$, the details of f lost by switching from a finer resolution 2^{-j-1} to a coarser resolution 2^{-j} and a reverse method of reconstructing a finer approximation from a coarser approximation and the details lost in between. Now since $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ and $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ are orthonormal bases of V_j and W_j respectively, Fast Wavelet Transform can be thought of as decomposing f into the wavelet coefficients

$$a_j[n] = \langle f, \phi_{j,n}\rangle \text{ and } d_j[n] = \langle f, \psi_{j,n}\rangle\,.$$

Let $F \in L^2(\mathbb{R})$ be any signal. Then since $\{\psi_{j,n}\}_{-\infty<j,n<\infty}$ forms an orthonormal basis of $L^2(\mathbb{R})$, we will have

$$F = \sum_{j=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \langle F, \psi_{j,n}\rangle\, \psi_{j,n} \tag{4.18}$$

$\psi_{j,n}$, a dilated and translated version of ψ is a wavelet profile with a scale parameter j and a location parameter n. The equation (4.18) ensures that any signal F can be constructed using infinitely many of these wavelet profiles $\psi_{j,n}$. The wavelet coefficient $\langle F, \psi_{j,n}\rangle$ is given by

$$\langle F, \psi_{j,n}\rangle = \int_{-\infty}^{+\infty} F(t)\psi^*_{j,n}(t)\,dt.$$

It measures the correlation between the signal F and the wavelet profile $\psi_{j,n}$ which quantifies the contribution of this profile in constructing the signal F. Another property of an orthonormal basis is that if we fix a dilation, any two translated wavelet profiles are uncorrelated or orthogonal to each other. This can be explained in terms of correlation as below.

$$\langle \psi_{j,n_1}, \psi_{j,n_2}\rangle = \int_{-\infty}^{+\infty} \psi_{j,n_1}(t)\psi^*_{j,n_2}(t)\,dt = 0,$$

Let $\bar{x}[n] = x[-n]$ and $\check{x}[n]$ denote a factor 2 upsampling of $x[n]$ obtained by expanding by alternate zero insertions:

$$\check{x}[n] = \begin{cases} x[p] & \text{if } \ n = 2p \\ 0 & \text{if } \ n = 2p+1. \end{cases}$$

Note the following Fourier transforms of upsampled and downsampled signals: The Fourier transform of $y[n] = x[2n]$ which is the signal $x[n]$ downsampled by factor 2 is

$$\hat{y}(2\omega) = \sum_{n=-\infty}^{+\infty} x[2n]e^{-i2n\omega} = \frac{1}{2}\left(\hat{x}(\omega) + \hat{x}(\omega+\pi)\right). \tag{4.19}$$

The Fourier transform of $y[n] = \check{x}[n]$ which is the signal $x[n]$ upsampled by factor 2 is

$$\hat{y}(\omega) = \sum_{n=-\infty}^{+\infty} x[n]e^{-i2n\omega} = \hat{x}(2\omega). \tag{4.20}$$

The following theorem shows that fast wavelet coefficients are calculated with a cascade of discrete convolutions and factor 2 downsamplings.

Theorem 51. [Mallat[6][7][8]] At the decomposition,

$$a_{j+1}[p] = \sum_{n=-\infty}^{+\infty} h[n-2p]a_j[n] = a_j \star \bar{h}[2p], \tag{4.21}$$

$$d_{j+1}[p] = \sum_{n=-\infty}^{+\infty} g[n-2p]a_j[n] = a_j \star \bar{g}[2p]. \tag{4.22}$$

At the reconstruction,

$$\begin{aligned} a_j[p] &= \sum_{n=-\infty}^{+\infty} h[p-2n]a_{j+1}[n] + \sum_{n=-\infty}^{+\infty} g[p-2n]d_{j+1}[n] \\ &= \check{a}_{j+1} \star h[p] + \check{d}_{j+1} \star g[p]. \end{aligned} \tag{4.23}$$

Proof [8]: Since $\phi_{j+1,p} \in V_{j+1} \subseteq V_j$ and $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_j, we have

$$\phi_{j+1,p} = \sum_{n=-\infty}^{+\infty} \langle \phi_{j+1,p}, \phi_{j,n} \rangle \, \phi_{j,n}. \tag{4.24}$$

With the change of variable $s = 2^{-j}t^{-2}p$, compute

$$\begin{aligned}
\langle \phi_{j+1,p}, \phi_{j,n} \rangle &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2^{j+1}}} \phi\left(\frac{t - 2^{j+1}p}{2^{j+1}}\right) \frac{1}{\sqrt{2^{j}}} \phi^{*}\left(\frac{t - 2^{j}n}{2^{j}}\right) dt \\
& \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) \phi^{*}\left(t - n + 2p\right) dt \\
&= \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi\left(t - n + 2p\right) \right\rangle \\
&= h[n - 2p].
\end{aligned}$$

Then (4.24) becomes

$$\phi_{j+1,p} = \sum_{n=-\infty}^{+\infty} h[n-2p]\phi_{j,n}.$$

Now, computing the inner product of each side with f, we get

$$a_{j+1}[p] = \sum_{n=-\infty}^{+\infty} h[n-2p]a_j[n] = a_j \star \bar{h}[2p].$$

Since $\psi_{j+1,p} \in W_{j+1} \subseteq V_j$ and $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_j, we have

$$\psi_{j+1,p} = \sum_{n=-\infty}^{+\infty} \langle \psi_{j+1,p}, \phi_{j,n} \rangle \, \phi_{j,n}. \tag{4.25}$$

With the change of variable $s = 2^{-j}t^{-2}p$, compute

$$\begin{aligned}
\langle \psi_{j+1,p}, \phi_{j,n} \rangle &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2^{j+1}}} \psi\left(\frac{t - 2^{j+1}p}{2^{j+1}}\right) \frac{1}{\sqrt{2^{j}}} \phi^{*}\left(\frac{t - 2^{j}n}{2^{j}}\right) dt \\
& \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right) \phi^{*}\left(t - n + 2p\right) dt \\
&= \left\langle \frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right), \phi\left(t - n + 2p\right) \right\rangle \\
&= g[n - 2p].
\end{aligned}$$

Then (4.25) becomes

$$\psi_{j+1,p} = \sum_{n=-\infty}^{+\infty} g[n-2p]\phi_{j,n}.$$

Now, computing the inner product of each side with f, we get

$$d_{j+1}[p] = \sum_{n=-\infty}^{+\infty} g[n-2p]a_j[n] = a_j \star \bar{g}[2p].$$

Since $\phi_{j,p} \in V_j$ and $V_j = V_{j+1} \oplus W_{j+1}$, we have

$$\begin{aligned}\phi_{j,p} &= \sum_{n=-\infty}^{+\infty} \langle \phi_{j,p}, \phi_{j+1,n} \rangle \, \phi_{j+1,n} + \sum_{n=-\infty}^{+\infty} \langle \phi_{j,p}, \psi_{j+1,n} \rangle \, \psi_{j+1,n} \\ &= \sum_{n=-\infty}^{+\infty} h[p-2n]\phi_{j+1,n} + \sum_{n=-\infty}^{+\infty} g[p-2n]\psi_{j+1,n}.\end{aligned}$$

Now, computing the inner product with f on both sides of this inequality, we get

$$a_j[p] = \sum_{n=-\infty}^{+\infty} h[p-2n]a_{j+1}[n] + \sum_{n=-\infty}^{+\infty} g[p-2n]d_{j+1}[n] = \check{a}_{j+1} \star h[p] + \check{d}_{j+1} \star g[p].$$

■

Figure (5.4) (source: http://www.ceremade.dauphine.fr/ peyre/wavelet-tour/) illustrates the one-dimensional Fast Wavelet Transform Algorithm.

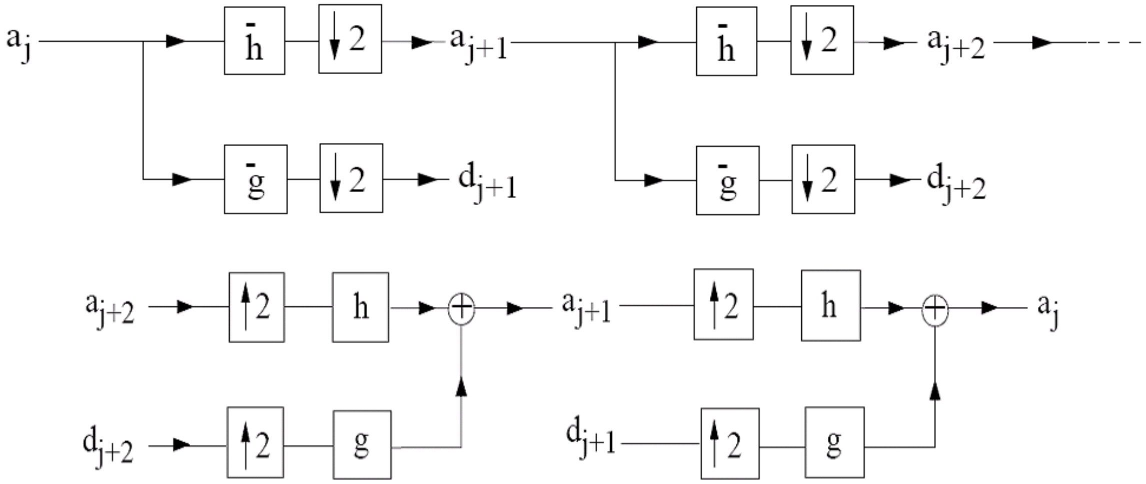

Figure 5.4: *Decomposition and Reconstruction diagrams of the 1-D Fast Wavelet Transform Algorithm.*

Definition 52 (Filter Bank)**.** A *two-channel multirate filter bank* consists of a low-pass filter $\bar{h}[n] = h[-n]$ and a high-pass filter $\bar{g}[n] = g[-n]$ which decomposes any given signal a_j at resolution 2^{-j} into approximation and detail coefficients at coarser resolution 2^{-j-1} as:

$$a_{j+1}[n] = a_j \star \bar{h}[2n] \text{ and } d_{j+1}[n] = a_j \star \bar{g}[2n]. \tag{4.26}$$

A reconstructed signal $\tilde{a}_j$ is obtained by a dual low-pass filter $\tilde{h}$ and a dual high-pass filter $\tilde{g}$ as:

$$\hat{a}_j[n] = \check{a}_{j+1} \star \tilde{h}[n] + \check{d}_{j+1} \star \tilde{g}[n]. \tag{4.27}$$

We are after a perfect reconstruction filter bank, i.e., the one which guarantees $\tilde{a}_j = a_j$. The following theorem sets the criteria for this. Its proof is taken from Mallat's book [8].

Theorem 53 (Vetterli[11][8])**.** The filter bank performs an exact reconstruction for any input signal if and only if

$$\hat{h}^*(\omega+\pi)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega+\pi)\hat{\tilde{g}}(\omega) = 0, \tag{4.28}$$

and

$$\hat{h}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega)\hat{\tilde{g}}(\omega) = 2. \tag{4.29}$$

Proof [8]: Since h and g are real, we have $\hat{\bar{h}}(\omega) = \hat{h}(-\omega) = \hat{h}^*(\omega)$ and $\hat{\bar{g}}(\omega) = \hat{g}(-\omega) = \hat{g}^*(\omega)$. Now, using the Fourier transform property (vii) and (50) in (4.26), we get

$$\begin{aligned} a_{j+1}(2\omega) &= \frac{1}{2}\Big(\hat{a}_j(\omega)\hat{h}^*(\omega) + \hat{a}_j(\omega+\pi)\hat{h}^*(\omega+\pi)\Big). \\ d_{j+1}(2\omega) &= \frac{1}{2}\left(\hat{a}_j(\omega)\hat{g}^*(\omega) + \hat{a}_j(\omega+\pi)\hat{g}^*(\omega+\pi)\right). \end{aligned}$$

On the other hand, using (50) in (4.27) yields

$$\hat{\tilde{a}}_j(\omega) = \hat{a}_{j+1}(2\omega)\hat{\tilde{h}}(\omega) + \hat{d}_{j+1}(2\omega)\hat{\tilde{g}}(\omega).$$

Hence

$$\hat{\tilde{a}}_j(\omega) = \frac{1}{2}\Big(\hat{h}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega)\hat{\tilde{g}}(\omega)\Big)\hat{a}_j(\omega) + \frac{1}{2}\Big(\hat{h}^*(\omega+\pi)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega+\pi)\hat{\tilde{g}}(\omega)\Big)\hat{a}_j(\omega+\pi).$$

For perfect reconstruction, i.e., to ensure $\hat{\tilde{a}}_j(\omega) = \hat{a}_j(\omega)$, we must have

$$\hat{h}^*(\omega+\pi)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega+\pi)\hat{\tilde{g}}(\omega) = 0,$$

and

$$\hat{h}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega)\hat{\tilde{g}}(\omega) = 2$$

which are (4.28) and (4.29). ■

The criteria in the above theorem can be written, in matrix form, as

$$\begin{bmatrix} \hat{h}(\omega) & \hat{g}(\omega) \\ \hat{h}(\omega+\pi) & \hat{g}(\omega+\pi) \end{bmatrix} \times \begin{bmatrix} \hat{\tilde{h}}^*(\omega) \\ \hat{\tilde{g}}^*(\omega) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Then

$$\begin{bmatrix} \hat{\tilde{h}}^*(\omega) \\ \hat{\tilde{g}}^*(\omega) \end{bmatrix} = \begin{bmatrix} \hat{h}(\omega) & \hat{g}(\omega) \\ \hat{h}(\omega+\pi) & \hat{g}(\omega+\pi) \end{bmatrix}^{-1} \times \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \frac{2}{\triangle(\omega)} \begin{bmatrix} \hat{g}(\omega+\pi) \\ -\hat{h}(\omega+\pi) \end{bmatrix} \tag{4.30}$$

where

$$\triangle(\omega) = \hat{h}(\omega)\hat{g}(\omega+\pi) - \hat{h}(\omega+\pi)\hat{g}(\omega). \tag{4.31}$$

The following theorem, proof of which has been taken from Mallat's book [8], states the criteria for perfect reconstruction in a different form.

Theorem 54 ([8])**.** Perfect reconstruction filters satisfy

$$\hat{h}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{h}^*(\omega+\pi)\hat{\tilde{h}}(\omega+\pi) = 2. \tag{4.32}$$

For finite impulse-response filters, there exist $a \in \mathbb{R}$ and $l \in \mathbb{Z}$ such that

$$\hat{g}(\omega) = ae^{-i(2l+1)\omega}\hat{\tilde{h}}^*(\omega+\pi) \text{ and } \hat{\tilde{g}}(\omega) = a^{-1}e^{-i(2l+1)\omega}\hat{h}^*(\omega+\pi). \tag{4.33}$$

Proof [8]: From (4.30), we have that

$$\hat{\tilde{h}}^*(\omega) = \frac{2}{\triangle(\omega)}\hat{g}(\omega+\pi) \text{ and } \hat{\tilde{g}}^*(\omega) = \frac{2}{\triangle(\omega)}\hat{h}(\omega+\pi) \tag{4.34}$$

Therefore,

$$\hat{g}(\omega)\hat{\tilde{g}}^*(\omega) = -\frac{\triangle(\omega+\pi)}{\triangle(\omega)}\hat{\tilde{h}}^*(\omega+\pi)\hat{h}(\omega+\pi) = \hat{\tilde{h}}^*(\omega+\pi)\hat{h}(\omega+\pi), \tag{4.35}$$

since $\triangle(\omega+\pi) = -\triangle(\omega)$ by (4.31). Using (4.35) in (4.29) yields (4.32). If g and h are finite impulse response filters, $\hat{g}(\omega)$ and $\hat{h}(\omega)$ are finite series in $e^{\pm in\omega}$. By (4.31), $\triangle(\omega)$ then is a finite series $e^{\pm in\omega}$. Further by (4.34), $\triangle(\omega)^{-1}$ is also a finite series. Consequently, $\triangle(\omega)$ must be a single term. Since $\triangle(\omega+\pi) = -\triangle(\omega)$, the exponent n must be odd. Therefore, there exist $l \in \mathbb{Z}$ and $a \in \mathbb{R}$ such that

$$\triangle(\omega) = -2ae^{-i(2l+1)\omega}.$$

Using this in (4.34), yields the remaining results of the theorem. ■

The following theorem, whose proof is taken from Mallat's book [8], constructs biorthogonal Riesz bases of $l^2(\mathbb{Z})$.

Definition 55 (Biorthogonal Bases)**.** Two bases $\{h[n-2l], g[n-2l]\}_{l\in\mathbb{Z}}$ and $\{\tilde{h}[n-2l], \tilde{g}[n-2l]\}_{l\in\mathbb{Z}}$ are said to be *biorthogonal* if we have for all $n \in \mathbb{Z}$,

$$\left\langle \tilde{h}[n], h[n-2l] \right\rangle = \delta[l] \text{ and } \left\langle \tilde{h}[n], h[n-2l] \right\rangle = \delta[l] \tag{4.36}$$

and

$$\left\langle \tilde{h}[n], g[n-2l] \right\rangle = 0 \text{ and } \langle \tilde{g}[n], h[n-2l] \rangle = 0 \tag{4.37}$$

Theorem 56. If $h, g, \tilde{h}$ and $\tilde{g}$ are perfect reconstruction filters, and their Fourier transforms are bounded, then $\{\tilde{h}[n-2l], \tilde{g}[n-2l]\}_{l\in\mathbb{Z}}$ and $\{h[n-2l], g[n-2l]\}_{l\in\mathbb{Z}}$ are biorthogonal Riesz bases of $l^2(\mathbb{Z})$.

Proof [8]: By (4.32), perfect reconstruction filters satisfy

$$\frac{1}{2}\left(\hat{h}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{h}^*(\omega+\pi)\hat{\tilde{h}}(\omega+\pi)\right) = 1$$

and

$$\frac{1}{2}\left(\hat{g}^*(\omega)\hat{\tilde{g}}(\omega) + \hat{g}^*(\omega+\pi)\hat{\tilde{g}}(\omega+\pi)\right) = 1.$$

These, on taking the inverse Fourier transform, yield

$$\bar{h} \star \tilde{h}[2l] = \delta[l] \text{ and } \bar{g} \star \tilde{g}[2l] = \delta[l].$$

That is,

$$\sum_{k=-\infty}^{\infty} \tilde{h}[n]\bar{h}[n-2l] \text{ and } \sum_{k=-\infty}^{\infty} \tilde{g}[n]\bar{g}[n-2l],$$

which are exactly the conditions (4.36) for biorthogonality. Again by (4.28), perfect reconstruction filters satisfy

$$\frac{1}{2}\hat{g}^*(\omega)\hat{\tilde{h}}(\omega) + \hat{g}^*(\omega+\pi)\hat{\tilde{h}}(\omega+\pi) = 0$$

and

$$\frac{1}{2}\hat{h}^*(\omega)\hat{\tilde{g}}(\omega) + \hat{h}^*(\omega+\pi)\hat{\tilde{g}}(\omega+\pi) = 0.$$

Taking the inverse Fourier transforms of these equations like before yield the conditions (4.37) for biorthogonality. Since the Fourier transform of each filter is bounded, the existence of Riesz bounds is easily verified. ■

Chapter 6

Wavelet Image Denoising

6.1 Separable Multiresolution Analysis or MRA

Since images are two-dimensional signals $f(x_1, x_2)$, in order to do multiresolution analysis of images, we need to have a two-dimensional MRA. This can be produced in the following way out of a one-dimensional MRA that we have discussed in great detail in the last chapter. Given any wavelet orthonormal basis $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ of $L^2(\mathbb{R})$, we can construct separable wavelet orthonormal bases $\{\psi_{j_1,n_1}(x_1)\psi_{j_2,n_2}(x_2)\}_{(j_1,n_1,j_2,n_2)\in\mathbb{Z}^4}$ of $L^2(\mathbb{R}^2)$. Clearly, $\psi_{j_1,n_1}(x_1)\psi_{j_2,n_2}(x_2)$ provide information at scale 2^{j_1} along x_1 as well as at scale 2^{j_2} along x_2.

Definition 57 (Separable MRA). A *separable two-dimensional MRA* of $L^2(\mathbb{R}^2)$ is which is a countable collection $\{V_j^2\}_{j\in\mathbb{Z}}$ of tensor products $V_j^2 = V_j \otimes V_j$ where $\{V_j\}_{j\in\mathbb{Z}}$ is a one-dimensional MRA of $L^2(\mathbb{R})$.

The space V_j^2 consists of two-dimensional signals

$$f(x_1, x_2) = \sum_{m=-\infty}^{\infty} a[m] f_m(x_1) g_m(x_2) \text{ where } f_m \in V_j,\ g_m \in V_j.$$

Next, we list two-dimensional versions of MRA theorems from last chapter. These results are analogous to their one-dimensional counterparts and follow directly after using basic properties of tensor products.

Theorem 58. [8] There exists a scaling function ϕ such that

$$\left\{ \phi^2_{j,n}(x) := \phi_{j,n_1}(x_1)\phi_{j,n_2}(x_2) = \frac{1}{2^j}\phi\left(\frac{x_1 - 2^j n_1}{2^j}\right)\phi\left(\frac{x_2 - 2^j n_2}{2^j}\right)\right\}_{n=(n_1,n_2)\in\mathbb{Z}^2}$$

is an orthonormal basis of V^2_j.

Let us denote by W^2_j the detail space which is the complement of the coarser resolution approximation space V^2_j in V^2_{j-1}, i.e., $V^2_{j-1} = V^2_j \oplus W^2_j$.

Theorem 59. [8]Let ϕ be a scaling function and ψ be the corresponding wavelet generating a wavelet orthonormal basis of $L^2(\mathbb{R})$. We define three wavelets:

$$\psi^1_{j,n}(x) := \phi_{j,n_1}(x_1)\psi_{j,n_2}(x_2), \psi^2_{j,n}(x) := \psi_{j,n_1}(x_1)\phi_{j,n_2}(x_2), \psi^3_{j,n}(x) := \psi_{j,n_1}(x_1)\psi_{j,n_2}(x_2),$$

and for $1 \leq k \leq 3$ denote

$$\psi^k_{j,n}(x) = \frac{1}{2^j}\psi^k\left(\frac{x_1 - 2^j n_1}{2^j}, \frac{x_2 - 2^j n_2}{2^j}\right).$$

The wavelet family

$$\{\psi^1_{j,n}, \psi^2_{j,n}, \psi^3_{j,n}\}_{n=(n_1,n_2)\in\mathbb{Z}^2}$$

is an orthonormal basis of W^2_j and

$$\{\psi^1_{j,n}, \psi^2_{j,n}, \psi^3_{j,n}\}_{(j,n)=(j,n_1,n_2)\in\mathbb{Z}^3}$$

is an orthonormal basis of $L^2(\mathbb{R}^2)$.

6.2 Two-dimensional Wavelet Transform

A *Fast Wavelet Transform Algorithm* in two dimensions is extended naturally from a one-dimensional version discussed in Section 4.2. Let $n = (n_1, n_2)$. For any given image $f(x) := f(x_1, x_2)$, the decomposition of f at any resolution level 2^{-j} (or, more commonly, at scale 2^j or just scale j, for short) will give us the approximation coefficient $a_j[n] = \langle f, \phi^2_{j,n}\rangle$ and three different detail coefficients, namely, horizontal detail coefficients $d^1_j[n] = \langle f, \psi^1_{j,n}\rangle$ vertical detail coefficients $d_j^2[n] = \left\langle f, \psi_{j,n}^2 \right\rangle$ and diagonal detail coefficients $d_j^3[n] = \left\langle f, \psi_{j,n}^3 \right\rangle$ where $\phi_{j,n}^2(x) := \phi_{j,n_1}(x_1)\phi_{j,n_2}(x_2), \psi_{j,n}^1(x) := \phi_{j,n_1}(x_1)\psi_{j,n_2}(x_2), \psi_{j,n}^2(x) := \phi_{j,n_1}(x_1)\psi_{j,n_2}(x_2)$ and $\psi_{j,n}^3(x) := \phi_{j,n_1}(x_1)\psi_{j,n_2}(x_2)$ are functions on $\mathbb{R}^2$ constructed from a scaling function ϕ and its corresponding wavelet function ψ on $\mathbb{R}$. These different coefficients $a_j[n], d_j^1[n], d_j^2[n]$ and $d_j^3[n]$, upon reconstruction, generate three different frequency subband images, which are denoted by LL_k, LH_k, HL_k and HH_k respectively, where k represents the number of steps of decomposition from the original image. Note that if the original image is of the scale 2^J, the first step decomposition subbands LL_1, LH_1, HL_1 and HH_1 correspond to wavelet coefficients at scale 2^{J+1} or, simply, $J + 1$. The subband LL_1 is exactly the approximation image of f at resolution 2^{-J-1}. The decomposition of LL_1 at scale 1 will now be the decomposition of f at scale $J + 2$. Thus, a J level decomposition of f will result in $3J + 1$ different wavelet coefficients or their corresponding $3J + 1$ different subbands. This is illustrated in Figure (6.1).

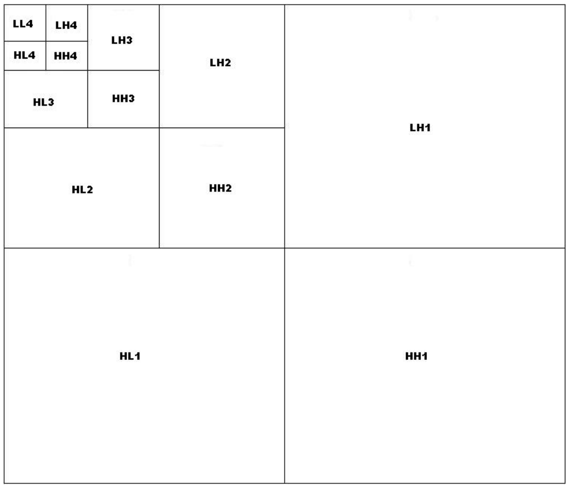

Figure 6.1: *Subbands of two-dimensional wavelet decomposition of an image.*

Given a pair of one-dimensional filters $y[m]$ and $z[m]$, denote $(yz)[n] = y[n_1]z[n_2]$. Recall the notation $\bar{y}[m] = y[-m]$. Also, note $(yz)[2n]$ is the factor 2 subsampling or downsampling of $(yz)[n]$ which is half the size of $(yz)[n]$, obtained by deleting the odd components in both row and column of $y[n]$ and recall the notation $(\check{yz})[n]$ for the factor 2 upsampling of $(yz)[n]$

which is the image twice the size of $(yz)[n]$ and which is obtained by inserting a row of zeros and a column of zeros between pairs of consecutive rows and columns of $(yz)[n]$.

Theorem 60. [8] Let $h[m]$ and $g[m]$ be the conjugate mirror filters associated to the wavelet ψ. At the decomposition,

$$a_{j+1}[n] = a_j \star (\bar{h}\bar{h})[2n],$$

$$d^1_{j+1}[n] = a_j \star (\bar{h}\bar{g})[2n],$$

$$d^2_{j+1}[n] = a_j \star (\bar{g}\bar{h})[2n],$$

$$d^3_{j+1}[n] = a_j \star (\bar{g}\bar{g})[2n].$$

And, at the reconstruction,

$$a_j[n] = \check{a}_{j+1} \star (hh)[n] + \check{d}^1_{j+1} \star (hg)[n] + \check{d}^2_{j+1} \star (gh)[n] + \check{d}^3_{j+1} \star (gg)[n].$$

6.3 MRA of $L^2[0,1]$

Images are two-dimensional finite discrete signals. Finite discrete signals are processed using an MRA of $L^2[0,1]$. The extension of a one-dimensional MRA to a separable two-dimensional MRA has been described in Section 6.1 and the one-dimensional fast wavelet transform algorithm was extended to the two-dimensional case in Section 6.2. With this in mind, we shall develop here a one-dimensional MRA of $L^2[0,1]$ which will also extend to a two-dimensional MRA of $L^2([0,1] \times [0,1])$, which will be the right MRA for images, in an analogous way. Since an MRA is generated with dyadic scaling and translation, we must have signals resized to some power of 2 by extending it with zero insertions. For example, a discrete signal of size $N = 2^l$ will be a signal in $L^2[0,1]$ when we refit it in $[0,1]$ with its entries $\frac{1}{N}$ distance apart. This distance is the original scale of the signal and its reciprocal is the original resolution. Then the finest scale at which we can decompose this signal will be $2/N = 2^{1-l}$ with coarsest resolution $N/2 = 2^{l-1}$. Recall that a finer scale corresponds to a larger size and coarser resolution.

A wavelet basis of $L^2[0,1]$ is obtained by transforming the wavelet basis of $L^2(\mathbb{R})$. The wavelets whose supports are contained in $[0,1]$ remain unchanged but the boundary wavelets whose supports overlap $t = 0$ and $t = 1$ are modified. We describe an MRA of $L^2[0,1]$ developed with the Daubechies wavelet ψ with p vanishing moments which means it is orthogonal to all polynomials of degree $p-1$ or smaller. Let ϕ be the Daubechies scaling function ϕ associated to ψ. Since, according to Theorem 7.7 [8], the size of the support of ϕ and ψ is $2p-1$, we translate ϕ and ψ so that their support is $[-p+1, p]$. The boundary coefficients associated to boundary wavelets are $2p$ in number at all scales. So the maximum scale 2^J must not exceed $(2p)^{-1}$. Then at a scale $2^j \leq (2p)^{-1}$, construct the following scaling functions:

(a) support inside $[0,1]$

$$\phi^{int}_{j,n}(t) = \phi_{j,n}(t),\ p \leq n < 2^j - p.$$

(b) support overlapped to the left of $t = 0$

$$\phi^{int}_{j,n}(t) = \frac{1}{\sqrt{2^j}} \phi^{left}_{n}\left(\frac{t}{2^j}\right),\ 0 \leq n < p.$$

(c) support overlapped to the right of $t = 1$

$$\phi^{int}_{j,n}(t) = \frac{1}{\sqrt{2^j}} \phi^{right}_{2^{-j}-1-n}\left(\frac{t-1}{2^j}\right),\ 0 \leq n < p.$$

The boundary scaling functions ϕ^{left}_n and ϕ^{left}_n are constructed in such a way that $\{\phi^{int}_{j,n}\}_{0 \leq n < 2^{-j}}$ forms an orthonormal basis of V^{int}_j and $\{V^{int}_j\}_{-\infty < j \leq 0}$ forms an MRA of $L^2[0,1]$. Constructing the wavelet functions $\psi^{int}_{j,n}(t)$ analogously, we get $\{\psi^{int}_{j,n}\}_{0 \leq n < 2^{-j}}$ as an orthonormal basis of W^{int}_j, the orthogonal complement of V^{int}_j in V^{int}_{j-1}. Hence, at any scale $2^J \leq (2p)^{-1}$, we obtain

$$[\{\phi^{int}_{J,n}\}_{0 \leq n < 2^{-J}}, \{\psi^{int}_{j,n}\}_{-\infty < j \leq J,\, 0 \leq n < 2^{-j}}]$$

as an orthonormal wavelet basis of $L^2[0,1]$.

6.4 Fast Wavelet Transform Algorithm

We will state without proofs the decomposition and reconstruction theorems of fast discrete algorithm for an MRA of $L^2[0,1]$. This algorithm breaks down decomposition and reconstruction formulas into various cases and replaces usual Daubechies filters h and g by modified filters at the boundary. It is slightly more complicated to implement than other simpler algorithms such as folding and periodic algorithms but requires less computations. Each decomposition or reconstruction of a signal of size N is performed with $O(N)$ operations.

Left boundary coefficient constants $H_{n,l}^{left}$, $G_{n,l}^{left}$, $h_{n,m}^{left}$ and $g_{n,m}^{left}$ satisfy the following equations in relation to the left boundary wavelets ϕ_n^{left} and ψ_n^{left} are supported in $[0, p+n]$.

$$2^{-1/2}\phi_n^{left}(2^{-1}t) = \sum_{l=0}^{p-1} H_{n,l}^{left}\phi_l^{left}(t) + \sum_{m=p}^{p+2n} h_{n,m}^{left}\phi(t-m),$$

$$2^{-1/2}\psi_n^{left}(2^{-1}t) = \sum_{l=0}^{p-1} G_{n,l}^{left}\phi_l^{left}(t) + \sum_{m=p}^{p+2n} g_{n,m}^{left}\phi(t-m).$$

Right boundary coefficient constants $H_{n,l}^{right}$, $G_{n,l}^{right}$, $h_{n,m}^{right}$ and $g_{n,m}^{right}$ satisfy similar equations in relation to the right boundary wavelets ϕ_n^{right} and ψ_n^{right} are supported in $[-p-n, 0]$.

Theorem 61. [8][Cohen, Daubechies, Vial] If $0 \le k < p$,

$$a_j[k] = \sum_{l=0}^{p-1} H_{k,l}^{left} a_{j-1}[l] + \sum_{m=p}^{p+2k} h_{k,m}^{left} a_{j-1}[m],$$

$$d_j[k] = \sum_{l=0}^{p-1} G_{k,l}^{left} a_{j-1}[l] + \sum_{m=p}^{p+2k} g_{k,m}^{left} a_{j-1}[m].$$

If $p \le k < 2^{-j} - p$,

$$a_j[k] = \sum_{l=-\infty}^{\infty} h[l-2k] a_{j-1}[l],$$

$$d_j[k] = \sum_{l=-\infty}^{\infty} g[l-2k] a_{j-1}[l].$$

If $-p \leq k < 0$,

$$a_j[2^{-j}+k] = \sum_{l=-p}^{-1} H_{k,l}^{right} a_{j-1}[2^{-j+1}+l] + \sum_{m=-p+2k+1}^{-p-1} h_{k,m}^{right} a_{j-1}[2^{-j+1}+m],$$

$$d_j[2^{-j}+k] = \sum_{l=-p}^{-1} G_{k,l}^{right} a_{j-1}[2^{-j+1}+l] + \sum_{m=-p+2k+1}^{-p-1} g_{k,m}^{right} a_{j-1}[2^{-j+1}+m],$$

This theorem provides a Fast wavelet transform algorithm to decompose the approximation coefficient a_L at scale 2^L into a finer (resolution) approximation coefficient a_J at a coarser scale 2^J and intermediary detail coefficients $\{d_j\}_{L<j\leq J}$ with $O(N)$ operations.

Theorem 62. [8](Cohen, Daubechies, Vial) If $0 \leq l \leq p-1$,

$$a_{j-1}[l] = \sum_{k=0}^{p-1} H_{k,l}^{left} a_j[k] + \sum_{k=0}^{p-1} G_{k,l}^{left} d_j[k].$$

If $p \leq l \leq 3p-2$,

$$\begin{aligned} a_{j-1}[l] = &\sum_{k=(l-p)/2}^{p-1} h_{k,l}^{left} a_j[k] + \sum_{k=-\infty}^{\infty} h[l-2k] a_j[k] \\ &+ \sum_{k=(l-p)/2}^{p-1} g_{k,l}^{left} d_j[k] + \sum_{k=-\infty}^{\infty} g[l-2k] d_j[k]. \end{aligned}$$

If $3p-1 \leq l \leq 2^{-j+1}-3p$,

$$a_{j-1}[l] = \sum_{k=-\infty}^{\infty} h[l-2k] a_j[k] + \sum_{k=-\infty}^{\infty} g[l-2k] d_j[k].$$

If $-p-1 \geq l \geq -3p+1$,

$$\begin{aligned} a_{j-1}[2^{-j+1}+l] = &\sum_{k=-p}^{(l+p-1)/2} h_{k,l}^{right} a_j[2^{-j}+k] + \sum_{k=-\infty}^{\infty} h[l-2k] a_j[2^{-j}+k] \\ &+ \sum_{k=-p}^{(l+p-1)/2} g_{k,l}^{right} d_j[2^{-j}+k] + \sum_{k=-\infty}^{\infty} g[l-2k] d_j[2^{-j}+k]. \end{aligned}$$

If $-1 \geq l \geq -p$,

$$a_{j-1}[2^{-j+1}+l] = \sum_{k=-p}^{-1} H_{k,l}^{right} a_j[2^{-j}+k] + \sum_{k=-p}^{-1} G_{k,l}^{right} d_j[2^{-j}+k].$$

This theorem reconstructs the original coarser approximation coefficient a_L at scale 2^L from $[a_J, \{d_j\}_{L<j\leq J}]$ by performing an iteration of these formulas for $L < j \leq J$ with $O(N)$ operations.

Figure (6.2) (source: http://www.ceremade.dauphine.fr/ peyre/wavelet-tour/) illustrates the one-dimensional Fast Wavelet Transform Algorithm.

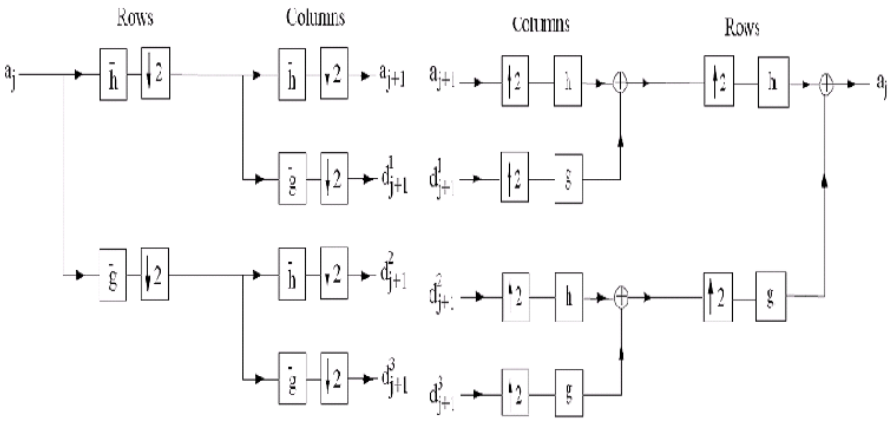

Figure 6.2: *Decomposition and Reconstruction diagrams of the 2-D Fast Wavelet Transform Algorithm.*

6.5 Noise Variance estimator

Let $X[n] = F[n] + W[n], 0 \leq n < N$ be the observed noisy signal where $W[n] \sim N(0, \sigma^2 I_N$ is a white noise with mean zero and variance σ^2 and $F[n]$ is the unknown true signal. When F is piecewise smooth, a robust estimator of σ^2 is calculated from the median of the finest scale wavelet coefficients [2].

A signal F with N discrete samples has the original scale N^{-1} and the finest scale it can be decomposed at is $2N^{-1} = 2^l$, say, with the corresponding finest resolution being

2^{-l}. The noisy signal X of size N is decomposed into $N/2$ wavelet coefficients at the finest scale 2^l. Thus, a robust estimator of σ^2 is calculated from the median of these coefficients $\{\langle X, \psi_{l,m}\rangle\}_{0\leq m\leq N/2}$. The median of these is the middle coefficient $\langle X, \psi_{l,N_0}\rangle$ of rank $N/4$. Also, we saw that if W is normally distributed with mean zero and variance σ^2, then $\langle W, \psi_{l,m}\rangle$ are independent Gaussian random variables of variance σ^2. If F is smooth over the support of $\psi_{l,m}$, $\langle F, \psi_{l,m}\rangle$ is small and $\langle X, \psi_{l,m}\rangle \approx \langle W, \psi_{l,m}\rangle$. In order to estimate the white noise variance we use the following fact:

Fact[8]: If M is the median of the absolute value of p independent Gaussian random variables with mean zero and variance σ_0^2, then one can show that $E(M) \approx 0.6745\sigma_0^2$.

Assuming F is smooth enough, we neglect the influence of F and have $\langle X, \psi_{l,m}\rangle \approx \langle W, \psi_{l,m}\rangle$. Then the variance σ^2 of the noise W is estimated from the median M_x of $\{|\langle X, \psi_{l,m}\rangle|\}_{0\leq m\leq N/2}$ by computing $\tilde{\sigma} = \frac{M_x}{0.6745}$.

6.6 Wavelet Thresholding

Let $X[n] = F[n] + W[n], n = (n_1, n_2), 0 \leq n_1, n_2 < \sqrt{N}$ be the noisy image data where $F[n]$ is the true image and $X[n]$ is the noisy image. The denoised image is an estimate of the true image $F[n]$. The main idea of image denoising is to first obtain the noise details with the help of noisy wavelet coefficients $\langle X, \psi_{j,n}\rangle$. But

$$\langle X, \psi_{j,n}\rangle = \langle F, \psi_{j,n}\rangle + \langle W, \psi_{j,n}\rangle .$$

The bigger coefficients come from wavelet profiles highly correlated to $X[n]$ and are dominated by true image details $\langle F, \psi_{j,n}\rangle$. Smaller coefficients, on the other hand, come from poorly correlated profiles and are dominated by noise details $\langle W, \psi_{j,n}\rangle$. So, in order to remove noise details, we need to remove the wavelet coefficients smaller than a threshold value. The threshold value is to be chosen so that we have a good balance of retaining true image details and losing noise details. Section 4.4 discussed that a good global threshold value is $T = \sigma\sqrt{2\ln N}$.

As discussed in Section 6.5, we estimate the noise variance from the observed noisy image data. If the original scale of an image is 2^{J-1}, we first calculate the median of the finest scale diagonal detail coefficients $d_J^3[n]$ corresponding to the highest diagonal subband HH_1. Then the estimated noise variance is

$$\tilde{\sigma} = \frac{median(|d_J^3[n]|)}{0.6745}. \tag{4.1}$$

Let us consider a square image for convenience which has equal horizontal and vertical size. The support of this image is normalized to $[0,1] \times [0,1]$ with N samples spaced by $N^{-1} = 2^L$ in either direction. Let the scale parameter which varies from 2^L at the original scale up to 2^1 at the coarsest scale be set at $2^J < 1$. After boundary modifications of discrete wavelets $\psi_{j,m}^k[n]$ and $\phi_{J,m}[n]$, suppose the wavelet basis for the decomposition at scale 2^J is

$$\mathcal{B} = \left[\{\psi_{j,m}^k[n]\}_{1\le k\le 3, L<j\le J, 0\le m<2^j},\ \{\phi_{J,m}^2[n]\}_{0\le m<2^{-J}} \right].$$

A thresholding estimator in this wavelet basis can be written as

$$\begin{aligned}\tilde{F} = &\sum_{m=0}^{2^{-J}} \rho_T\left(\langle X, \phi_{J,m}^2\rangle\right)\phi_{J,m}^2 + \sum_{j=L+1}^{J}\sum_{m=0}^{2^{-j}} \rho_T\left(\langle X, \psi_{j,m}^1\rangle\right)\psi_{j,m}^1 \\ &+ \sum_{j=L+1}^{J}\sum_{m=0}^{2^{-j}} \rho_T\left(\langle X, \psi_{j,m}^2\rangle\right)\psi_{j,m}^2 + \sum_{j=L+1}^{J}\sum_{m=0}^{2^{-j}} \rho_T\left(\langle X, \psi_{j,m}^3\rangle\right)\psi_{j,m}^3,\end{aligned}$$

where ρ_T is either a hard thresholding or a soft thresholding, displayed in Figure (4.1) in Section (4.4).

To reconstruct a denoised image at original scale $2^L = 1/N$ from the thresholding estimator, we reconstruct $a_{L,m}$ from $[a_{J,m}, \{d_{j,m}\}_{L<j\le J}]$ by performing an iteration of these formulas for $L < j \le J$ with $O(N)$ operations and computing

$$\tilde{F}_{denoised} = \sum_{m=0}^{2^{-L}} a_{L,m}\phi_{L,m}^2 = \sum_{m=0}^{N} \langle X, \phi_{L,m}^2\rangle \phi_{L,m}^2.$$

A global threshold, as we have seen in Section 4.4, is given by $T = \tilde{\sigma}\sqrt{2\ln N}$ where the noise variance estimate $\tilde{\sigma}$ is computed as in (4.1). However, for wavelet coefficients, there is usually room for improvement. For example, for images of size $N^2 = 512^2$ pixels, the universal threshold is $T = \tilde{\sigma}\sqrt{2\ln 512^2} \approx 5\tilde{\sigma}$. But hard-thresholding estimators are computed by choosing $T = 3\tilde{\sigma}$ because it increases the SNR value which, in turn, significantly improves the visual quality of the image. For soft-thresholding estimators which tend to have a larger risk, we choose $T = 3\tilde{\sigma}/2$ so that we could have nearly the same risk as hard-thresholding estimators.

Global thresholding is a very simple scheme of denoising and it works effectively in most cases. However, many other more complex algorithms of thresholding have been developed to improve the denoising effect which can be both level-dependent and orientation-dependent. Another threshold we would like to consider in this report is called Bayes Shrink which is an orientation-dependent thresholding scheme proposed by B. Yu, S. G. Chang and M. Vetterli in [10].

Bayes Shrink is a data-driven threshold computed in a Bayesian framework assuming the generalized Gaussian distribution as the prior distribution of the wavelet coefficients. Bayes Shrink thresholding scheme sets different thresholds for every subband, i.e., (T_{BS}) actually consists of three different threshold values for three different detail orientations, namely, horizontal, vertical and diagonal. Let us now describe how to calculate Bayes threshold for a fixed orientation. Suppose all the wavelet coefficients for a fixed orientation, say, vertical, are given by $\{V_k\}, k = 1, 2, ..., M$. We have noisy image data as

$$X[n] = F[n] + W[n],$$

where $F[n]$ is true image data and $W[n]$ is noise data whose variance can be estimated as explained in Section (6.5) and as computed in (4.1). Thus, we have

$$\hat{\sigma}_W^2 = \frac{median(|d_J^3[n]|)}{0.6745},$$

where $d_J^3[n]$ are diagonal detail coefficients corresponding to the highest diagonal subband HH_1. The variance of the noisy image, on the other hand, can be estimated by

$$\hat{\sigma}_X^2 = \frac{1}{M}\sum_{k=1}^{M} V_k^2.$$

Then, we compute

$$\hat{\sigma}_F^2 = \sqrt{max\{(\hat{\sigma}_X^2 - \hat{\sigma}_W^2), 0\}}.$$

Then the Bayes Shrink threshold for this vertical orientation is computed by

$$T_{BS} = \begin{cases} max\{|V_k|\}, & \text{if } \hat{\sigma}_F = 0 \\ \dfrac{\hat{\sigma}_W^2}{\hat{\sigma}_F}, & \text{otherwise.} \end{cases}$$

Note that $\hat{\sigma}_F = 0$, we remove all the wavelet coefficients by setting the threshold as the maximum coefficient. Also, it is to be expected that Bayes Shrink threshold should produce better denoising results than the Global threshold schemes as it takes into account the prior distribution of the wavelet coefficients and the variability of the coefficients due to orientation. This will be verified by experimentation in Section 6.7 later.

6.7 Image denoising via wavelet transform

With all the required theory developed in previous sections, we will, in this section, exemplify the computational implementation of image processing with actual images. For this purpose, we are using the software "Wavelet Toolbox" available with MATLAB7.7.0 (R2008B). Some built-in MATLAB codes as well as their results will be copied verbatim from the MATLAB command window itself.

Based on the theory, we will begin with an original image. We have chosen "Lena" for our illustration which is easily available on the web. Next we corrupt this image with some noise. Some common noises are "Salt and pepper", "Gaussian", "Poisson" and "Speckle". These are done using the following MATLAB commands.

```
f=imread (‘lena.png’);
g1=imnoise (f,‘salt & pepper’,.05); %.05 is the noise density.
%and affects approximately .05*numel(I) pixels.
g2=imnoise (f,‘poisson’); %generates Poisson noise from
```

```
%the data instead of adding artificial noise to the data.
g3=imnoise (f,‘speckle’,.04); %g3 = f+n*f, where n is uniformly
%distributed random noise with mean 0 and variance .04.
```

Some noisy images are displayed in Figure (6.3).

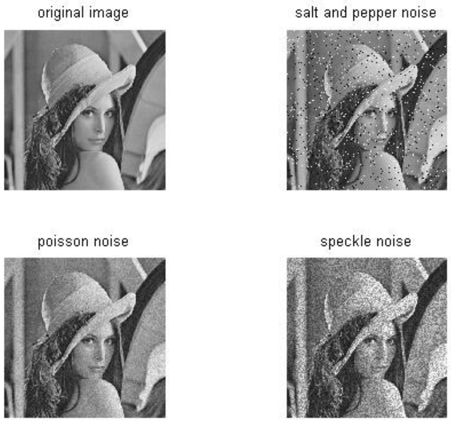

Figure 6.3: *Original lena image and noisy images with additive “salt and pepper noise” with noise density .05, “Poisson noise” generated from the image itself and multiplicative “speckle noise” with uniform noise variance .04.*

Since white Gaussian noise is the most common, we will illustrate our denoising procedure for an image corrupted with additive Gaussian noise with mean zero and variance .01. Next thing is to obtain a global wavelet threshold which is obtained using the following MATLAB command.

```
 [thr,sorh,keepapp] = ddencmp (‘den’,‘wv’,x); %‘den’ means
 %denoising and ‘wv’ means the filterbank is wavelets
```

In the output, “thr” gives a global positive threshold, “sorh” gives the default value ‘s’ which means soft thresholding and “keepapp” gives the default vale 1 which means the approximation coefficients are kept, not thresholded. The other option for “sorh” is ‘h’ for hard thresholding and the other option for “keepapp” is 0 for thresholding approximation coefficients. Since, as we will see, thresholding approximation coefficients removes important features of an image, we always keep approximation coefficients and only threshold detail coefficients.

We will also illustrate the denoising procedure using Bayes soft thresholding explained in Section (6.6). Since it is a level dependent thresholding, unlike "thr" in global thresholding, "bayesthr" is going to be a matrix of size 3 by the level of decomposition. All the columns are equal and the three entries of each column give the thresholds for horizontal, vertical and diagonal detail coefficients at each level respectively.

Next thing is to choose a filterbank of wavelets and the level of decomposition. In MATLAB, we do this by specifying the mother wavelet ψ. Figure (5.3) gives examples of some mother wavelets. We will observe the denoising process of the image of "Lena" using a number of different wavelets, namely, Daubechies 4, Daubechies 1, Haar, Symmlet 4 and Biorthogonal 6.8. These have been selected from a vast pool of wavelets available in MATLAB. Looking at the summary tables, we see that denoising results are almost the same for these different wavelets. These wavelets have different shapes and sizes. Due to these variations, they will capture different local features of the image. Since these subtle local features vary from image to image, we cannot conclude superiority of one wavelet over another in general. Thus, the use of different wavelets are more for illustration purposes than for comparison purposes.

Note that the level of decomposition of an image depends on the original size or scale of that image. For example, an image of size 512×512 is at original scale 2^{-9}. At decomposition level 3, the approximated image is going to be at scale 2^{-6} which is already too small to be seen clearly. The maximum decomposition level for this image is 8 at which the approximated image is going to be at scale 2^{-1}. The decomposition structure is illustrated with the image "A.tif" using Haar wavelets at decomposition level 3 in Figure (6.4) generated by the following MATLAB commands. MATLAB functions such as "wave2gray" and "wavefast" are not built-in MATLAB functions and have been taken from the book "Digital

Image Processing using MATLAB” [3].

```
f=imread (‘A.tif’);

subplot (2,2,1),imshow (f);title(‘original image’);

x=g;

wavelet=‘haar’; %filterbank is Haar wavelets; other options,

% for example, are ‘db4’, ‘sym8’, ‘coif3’, ‘bior6.8’.

level=1; %level 2 decomposition

[c,s]=wavefast (f,level,wavelet);

subplot (2,2,2),wave2gray (c,s,8);title (‘ ‘‘Haar" Level 1’);

level=2;

[c,s]=wavefast (f,level,wavelet);

subplot (2,2,3),wave2gray (c,s,8);title (‘ ‘‘Haar" Level 2’);

level=3;

[c,s]=wavefast (f,level,wavelet);

subplot (2,2,4),wave2gray (c,s,8);title (‘ ‘‘Haar" Level 3’);
```

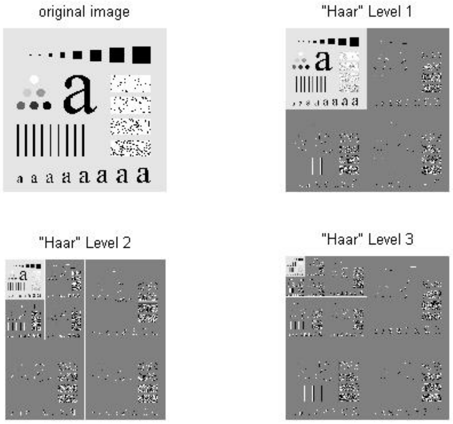

Figure 6.4: *Original image and Level 1, 2 and 3 decomposed structures using Haar wavelets.*

Once we obtain approximation and detail coefficients from the decomposition, we leave the approximation coefficients alone and choosing a threshold type (either global soft threshold or global hard threshold or Bayes soft threshold), kill off noise by thresholding the detail coefficients. Then a reconstruction out of these new detail coefficients and approximation coefficients will produce a denoised image. MATLAB commands for this procedure is given below.

```
>> f=imread (‘lena.png’);
>> g=imnoise (f,‘gaussian’,0,.01);
>> subplot (4,2,1),imshow (f);title(‘original image’);
>> subplot (4,2,2),imshow (g);title (‘noisy image’);
>> x=g;
>> [thr,sorh,keepapp] = ddencmp (‘den’,‘wv’,x);% ‘den’ means
% denoising and ‘wv’ means the filterbank is wavelets
>> wavelet=‘db4’; %filterbank is Daubechies 1 wavelets; other options,
% for example, are ‘haar’, ‘sym4’, ‘db1’, ‘bior6.8’.
>> level=1; %level 1 decomposition
>> thresholdtype=‘s’; %soft threshold; other option is ‘h’
>> keepapp=1; %approximation coefficients are kept, not thresholded;
% other option is 0
>> xdsk = wdencmp (‘gbl’,x,wavelet,level,thr,thresholdtype,keepapp);
>> xdsk=uint8 (xdsk);
>> subplot (3,2,3), imshow (xdsk);title(‘soft denoised, app. coeff. kept’);
>> [C,S]=wavedec2 (g,level,wavelet);
>> st=(S (1,1)^2)+1;
>> zer=length (st:1:length (C));
>> A=C (1:st-1);
>> B=zeros (1,zer);
>> bayesC=[A,B];
>> var=length (C)-S (size (S,1)-1,1)^2+1;
>> sigmahat=median (abs (C (var:length (C))))/0.6745;
>> for jj=2:size (S,1)-1
    % for the H detail coefficients
    coefh=C (st:st+S (jj,1)^2-1);
    thrHjj=bayes (coefh,sigmahat);
    bayesC (st:st+S (jj,1)^2-1)=sthresh (coefh,thrHjj);
    st=st+S (jj,1)^2;
% for the V detail coefficients
    coefv=C (st:st+S (jj,1)^2-1);
    thrVjj=bayes (coefv,sigmahat);
    bayesC (st:st+S (jj,1)^2-1)=sthresh (coefv,thrVjj);
    st=st+S (jj,1)^2;
% for Diag detail coefficients
    coefd=C (st:st+S (jj,1)^2-1);
    thrDjj=bayes (coefd,sigmahat);
    bayesC (st:st+S (jj,1)^2-1)=sthresh (coefd,thrDjj);
    st=st+S (jj,1)^2;
>> end
>> bayesthr=zeros(3,size(S,1)-2);
>> for jj=2:size (S,1)-1
    bayesthr(1,jj-1)=thrHjj;
    bayesthr(2,jj-1)=thrVjj;
    bayesthr(3,jj-1)=thrDjj;
>> end
>> xb=waverec2 (bayesC,S,wavelet);
>> xb=uint8 (xb);
>> subplot (3,2,5), imshow (xb);title (‘Bayes soft denoised’);
```

Using different thresholds and different levels of decomposition, the denoising effect can be compared. The comparison can be based on several versions of error between a noisy image and its denoised version. We will do some comparison using maximum entrywise difference (Maxabs), mean square error (MSE) and signal-to-noise ratio (SNR) which have been discussed in Section (2.5) using the following MATLAB commands which, in turn, uses previously written MATLAB functions "compare11.m" and "wpsnr.m" that computes MSE and SNR respectively.

```
>> ff=im2double (f);
>> xddsk=im2double (xdsk);
>> maxabssk=max(max(abs(ff-xddsk)));
>> msesk=compare11 (ff,xddsk);
>> snrsk=wpsnr (ff,xddsk);
```

Table (6.1) summarizes the denoising results for Noisy Lena with Daubechies 4 wavelets at decomposition level 1 obtained in Figure (6.5). The global threshold for hard and soft thresholding is 133.3075 and Bayes Shrink thresholds for horizontal, vertical and diagonal coefficients are $132.0946, 81.979$ and 188.8918 respectively.

It is observed that by thresholding approximation coefficients we lose important features of the image and significantly reduce the quality of the image. In fact, SNR values for denoised images obtained by thresholding approximation coefficients are much less than the noisy image itself.

Tables (6.2) and (6.3) below summarizes the denoising results for Noisy Lena with Daubechies 4 wavelets at decomposition levels 2 and 3 obtained in Figures (6.6) and (6.7) respectively. Since we have the same filterbank of Daubacies 4 wavelets, we have the same global threshold and Bayes thresholds for decompositions at different levels.

From the summary tables, we can compare the denoising effects for different levels of decomposition. Note that hard and soft global thresholding do not perform well with the decomposition at level 3. This reflects the fact that thresholding the decomposed coefficients at level 3 removed some essential features of the image. However, Bayes soft thresholding is consistently improving the quality of the denoised image with each higher level decomposition.

Table 6.1: *Summary of denoising results for Noisy Lena with* **Daubechies 4 at level 1**

	Soft threholding with app. coeff. kept	Soft thresholding all coeff.	Hard thresholding with app. coeff. kept	Hard thresholding all coeff.	Bayes soft thresholding
Maxabs	0.3216	0.5373	0.3373	0.4471	0.3098
MSE	0.0530	0.2548	0.0530	0.093	0.053
SNR	35.3768	17.9892	35.3768	28.2256	35.3771

Table 6.2: *Summary of denoising results for Noisy Lena with* **Daubechies 4 at level 2**

	Soft threholding with app. coeff. kept	Hard thresholding with app. coeff. kept	Bayes soft thresholding
Maxabs	0.3804	0.3569	0.3255
MSE	0.0419	0.0411	0.0373
SNR	35.4222	35.6923	36.6216

Table 6.3: *Summary of denoising results for Noisy Lena with* **Daubechies 4 at level 3**

	Soft threholding with app. coeff. kept	Hard thresholding with app. coeff. kept	Bayes soft thresholding
Maxabs	0.4588	0.4118	0.2824
MSE	0.049	0.0443	0.0347
SNR	32.9875	34.3918	37.6334

Table (6.4) summarizes the denoising results for Noisy Lena with Haar wavelets at decomposition level 2 obtained in Figure (6.8). The global threshold for hard and soft thresholding is 133.3075 and Bayes soft thresholds for horizontal, vertical and diagonal coefficients are $156.8834, 72.8668$ and 120 respectively. Note that soft thresholding is not effective in this case as it is reducing the SNR value of the noisy image. So, it verifies our description in Section (6.6) computationally that soft thresholding is generally weaker than hard thresholding and that we need to decrease the soft threshold in order to get the same denoising effect as that of the hard threshold.

Table (6.5) summarizes the denoising results for Noisy Lena with Symmlet 4 wavelets at decomposition level 2 obtained in Figure (6.9). The global threshold for hard and soft

Table 6.4: *Summary of denoising results for Noisy Lena with* **Haar at level 2**

	Soft threholding with app. coeff. kept	Hard thresholding with app. coeff. kept	Bayes soft thresholding
Maxabs	0.5294	0.5294	0.4431
MSE	0.0502	0.0486	0.0432
SNR	34.9354	35.4333	36.6617

thresholding is 133.3075 and Bayes soft thresholds for horizontal, vertical and diagonal coefficients are $236.5832, 97.8961$ and 120.0934 respectively.

Table 6.5: *Summary of denoising results for Noisy Lena with* **Symmlet 4 at level 2**

	Soft threholding with app. coeff. kept	Hard thresholding with app. coeff. kept	Bayes soft thresholding
Maxabs	0.4275	0.3608	0.3059
MSE	0.0419	0.0411	0.373
SNR	35.4049	35.6728	36.5901

Table (6.6) summarizes the denoising results for Noisy Lena with Biorthogonal 6.8 wavelets at decomposition level 2 obtained in Figure (6.10). The global threshold for hard and soft thresholding is 133.3075 and Bayes soft thresholds for horizontal, vertical and diagonal coefficients are $61.4229, 52.5046$ and 110.0743 respectively.

Table 6.6: *Summary of denoising results for Noisy Lena with* **Biortho 6.8 at level 2**

	Soft threholding with app. coeff. kept	Hard thresholding with app. coeff. kept	Bayes soft thresholding
Maxabs	0.4157	0.3176	0.2824
MSE	0.0406	0.0399	0.0378
SNR	35.6836	35.8594	36.603

Summary tables of denoising results with different wavelets reveal that Bayes Shrink thresholding performs consistently the best as it has the highest SNR value. Hard and soft thresholding schemes set the same global threshold for all three different orientations of detail, namely, horizontal, vertical and diagonal. On the other hand, Bayes Shrink thresholding scheme is more sophisticated as it sets different thresholds for these different orientations. Thus, Bayes Shrink thresholds work better than global hard and soft threholds.

Let us now look at denoising results for a different kind of image. Figure (6.11) shows an image which comes from Affymetrix microarray for wheat (Molly genotype) at one day after Hessian fly attack. We believe that the image we have observed is corrupted with noise and would like to denoise this image using the filterbank of Daubechies 4 wavelets at decomposition levels $1, 2$ and 3. Since we cannot know what the original image is, denoising results are summarized with figures only.

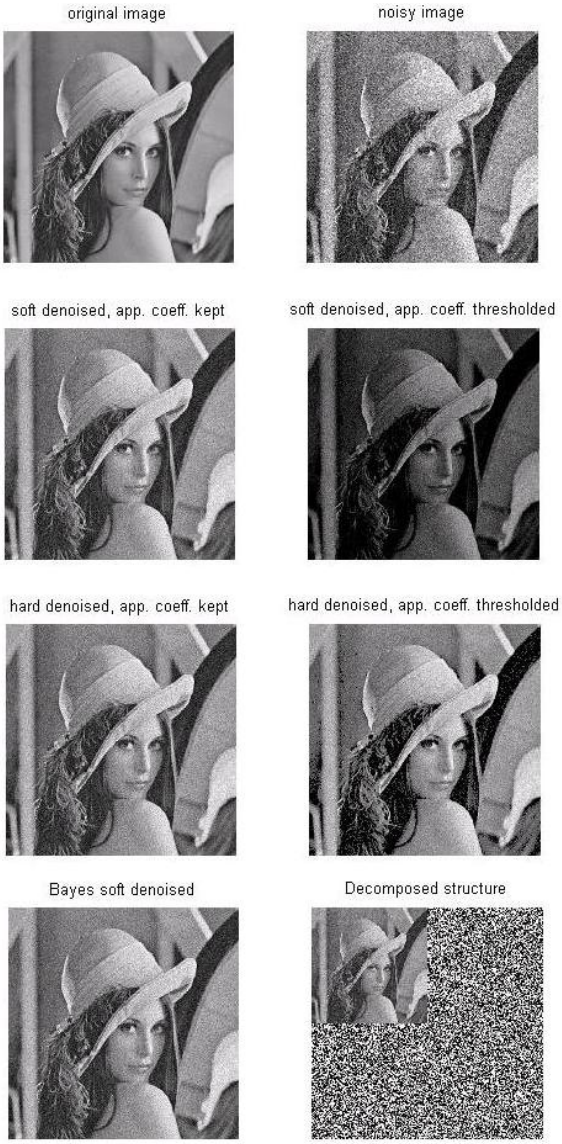
original image
noisy image
soft denoised, app. coeff. kept
soft denoised, app. coeff. thresholded
hard denoised, app. coeff. kept
hard denoised, app. coeff. thresholded
Bayes soft denoised
Decomposed structure

Figure 6.5: *Original Lena, Noisy Lena(SNR=35.2193), and several denoised versions of it with wavelets* **<u>Daubechies 4 at level 1</u>** *(Global threshold=133.3075 and Bayes thresholds: H=132.0946, V=81.979, D=188.8918): Soft-denoised keeping app. coeff.(SNR=35.3768), Soft-denoised thresholding app. coeff.(SNR=17.9892), Hard-denoised keeping app. coeff.(SNR=35.3768), Hard-denoised thresholding app. coeff.(SNR=28.2256) and Bayes soft-denoised (SNR=35.3771).*

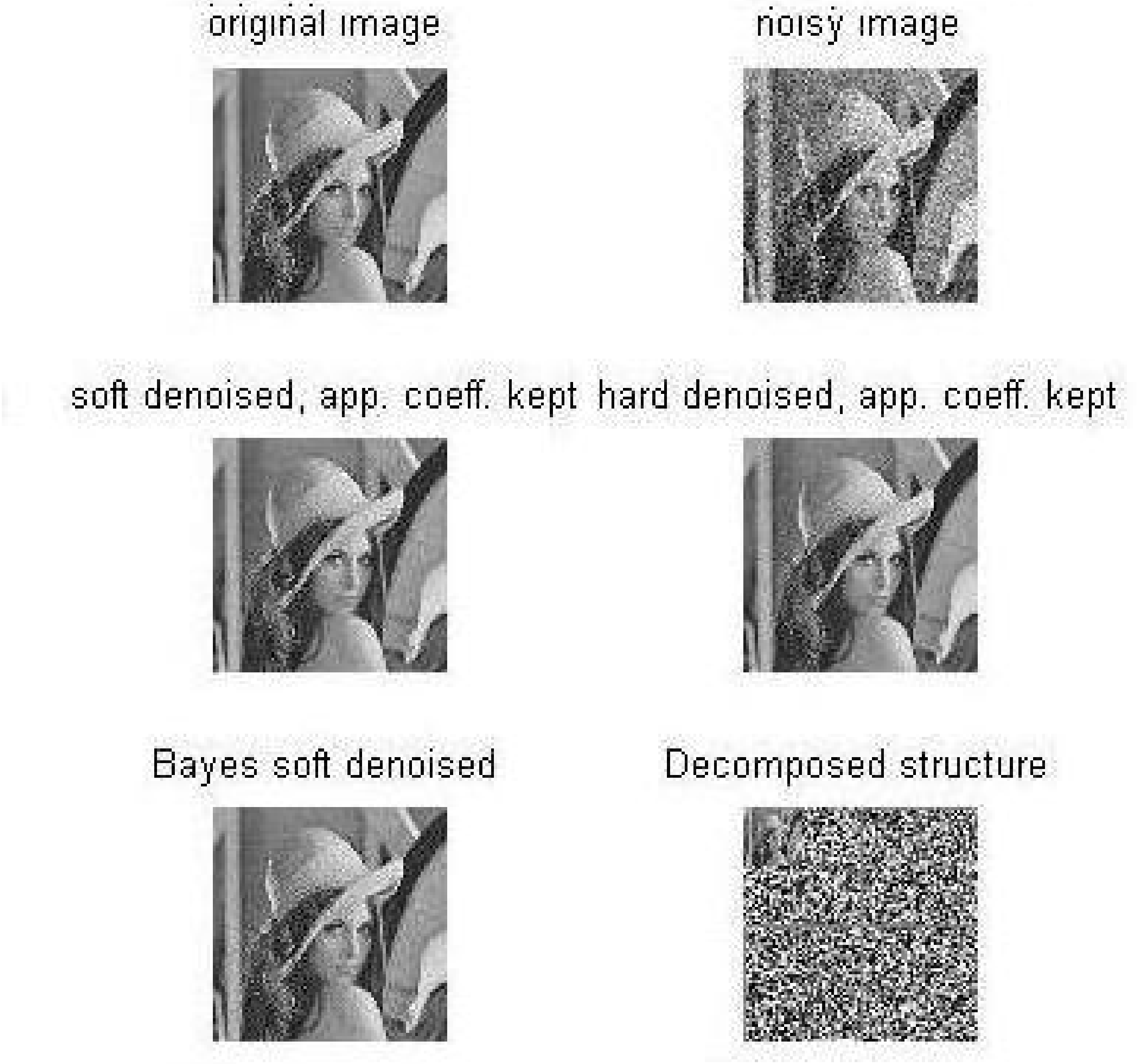

Figure 6.6: *Original image, noisy image(SNR=35.2193), and several denoised versions of it with wavelets* **<u>Daubechies 4 at level 2</u>** *(Global threshold=133.3075 and Bayes thresholds: H=132.0946, V=81.979, D=188.8918): Soft-denoised keeping app. coeff.(SNR=35.4222), Hard-denoised keeping app. coeff.(SNR=35.6923) and Bayes soft-denoised (SNR=36.6216).*

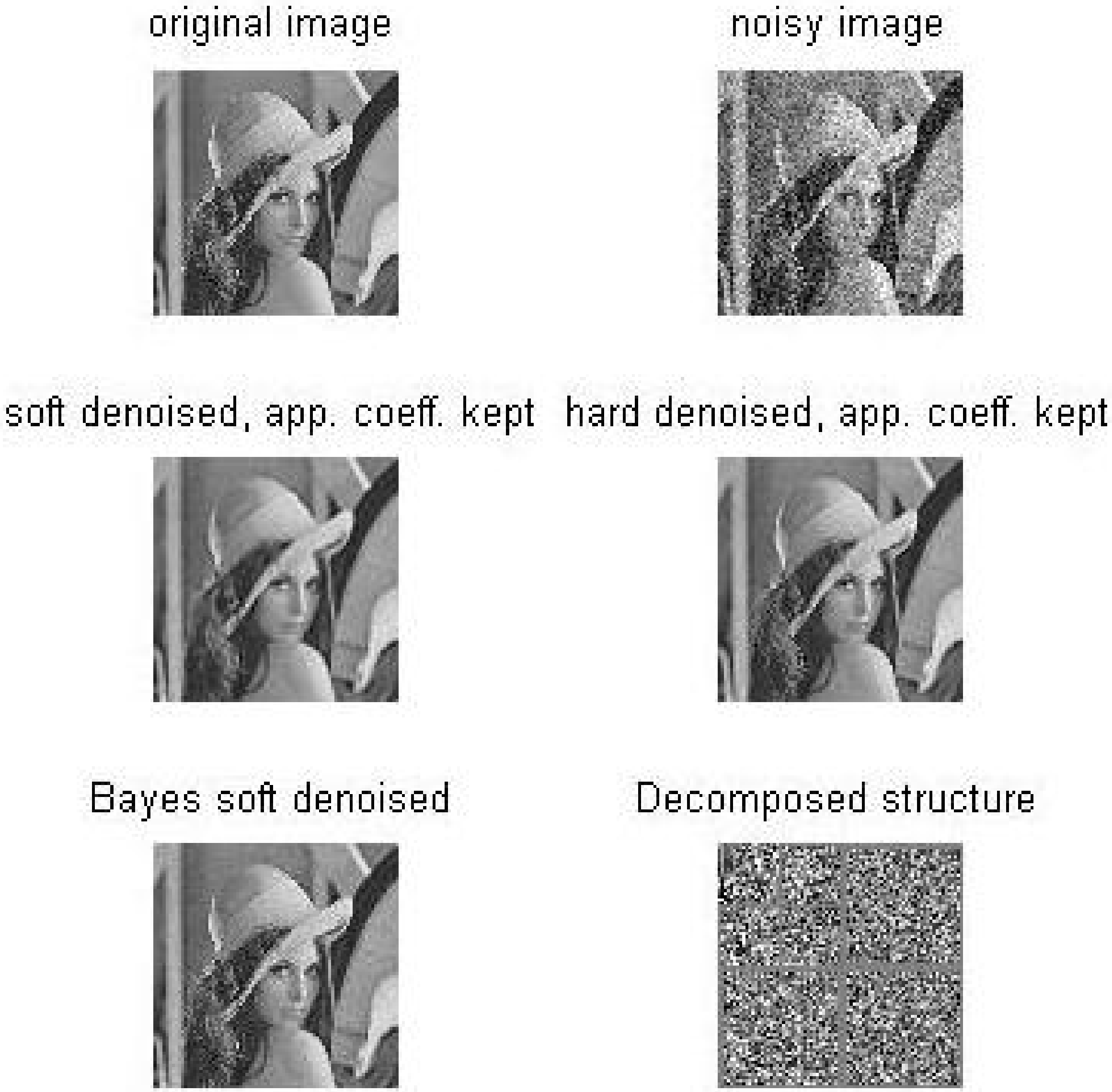

Figure 6.7: *Original Lena, Noisy Lena(SNR=35.2193), and several denoised versions of it with wavelets* **<u>Daubechies 4 at level 3</u>** *(Global threshold=133.3075 and Bayes thresholds: H=132.0946, V=81.979, D=188.8918): Soft-denoised keeping app. coeff.(SNR=32.9875), Hard-denoised keeping app. coeff.(SNR=34.3918) and Bayes soft-denoised (SNR=37.6334).*

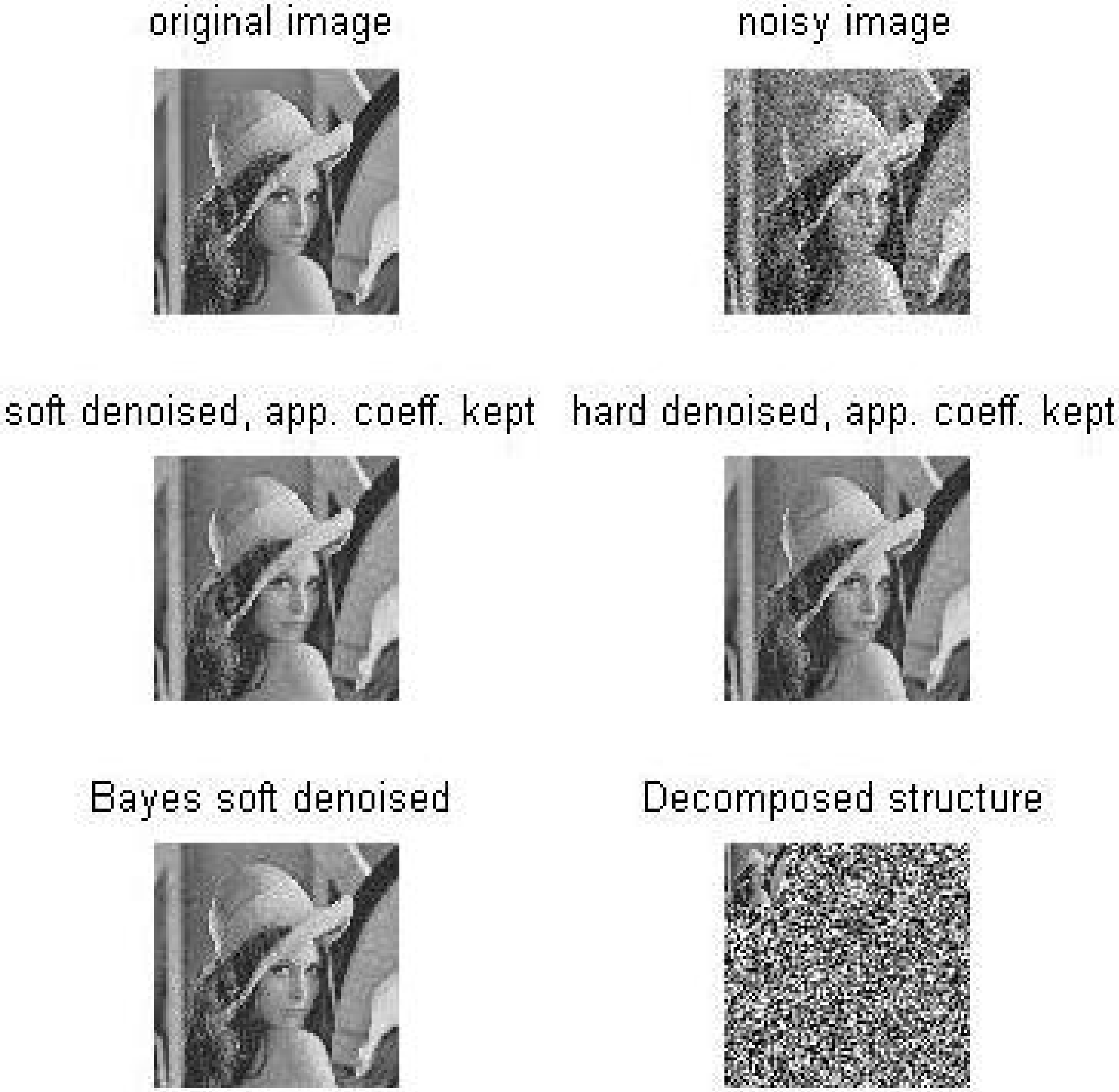

Figure 6.8: *Original Lena, Noisy Lena(SNR=35.2193), and several denoised versions of it with wavelets* **<u>Haar at level 2</u>** *(Global threshold=133.3075 and Bayes thresholds: H=156.8834, V=72.8668, D=120): Soft-denoised keeping app. coeff.(SNR=34.9354), Hard-denoised keeping app. coeff.(SNR=35.4333) and Bayes soft-denoised (SNR=36.6617).*

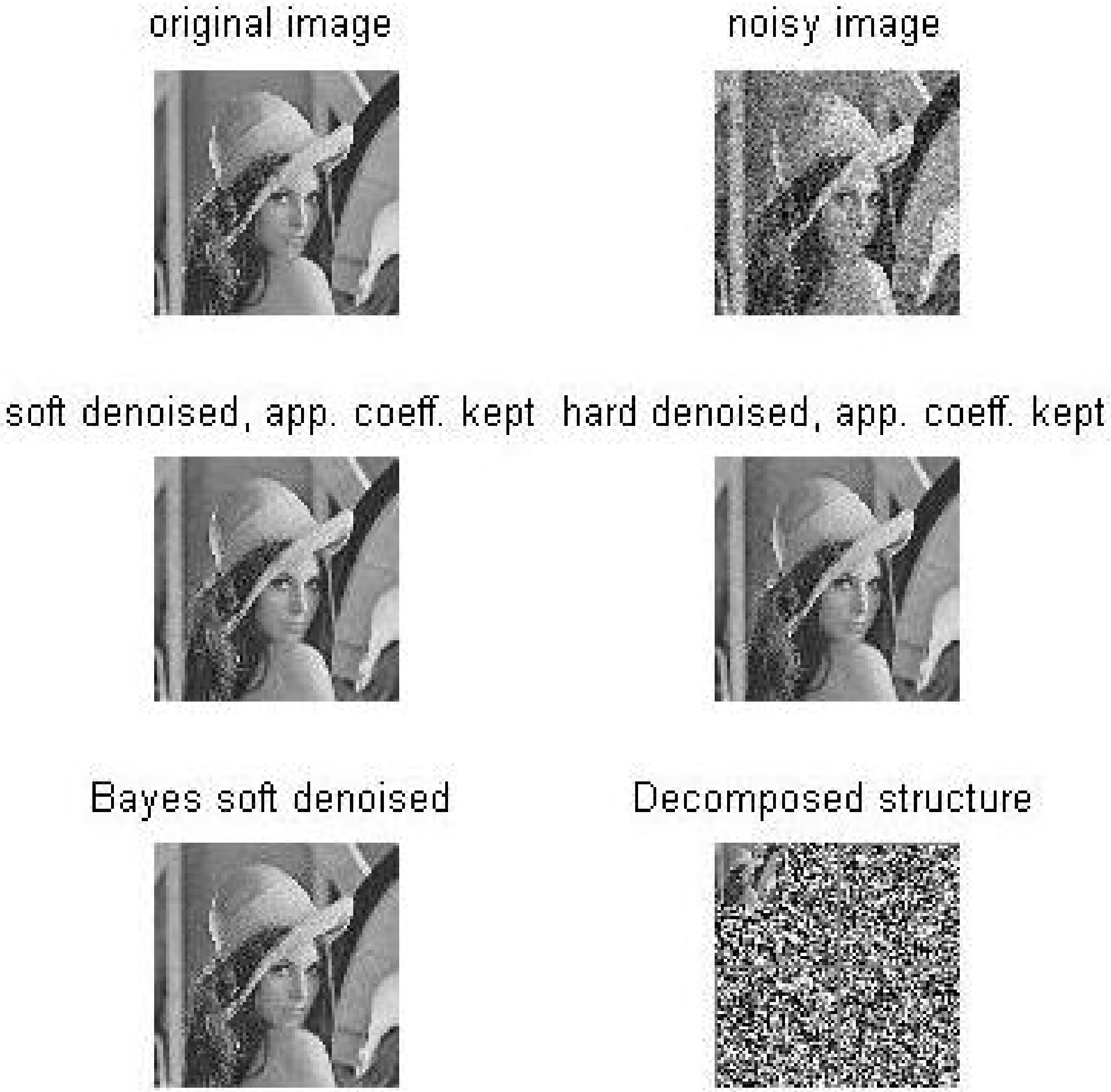

Figure 6.9: *Original Lena, Noisy Lena(SNR=35.2193), and several denoised versions of it with wavelets* **Symmlet 4 at level 2** *(Global threshold=133.3075 and Bayes thresholds: H=236.5832, V=96.8961, D=120.0934): Soft-denoised keeping app. coeff.(SNR=35.4049), Hard-denoised keeping app. coeff.(SNR=35.6728) and Bayes soft-denoised (SNR=36.5901).*

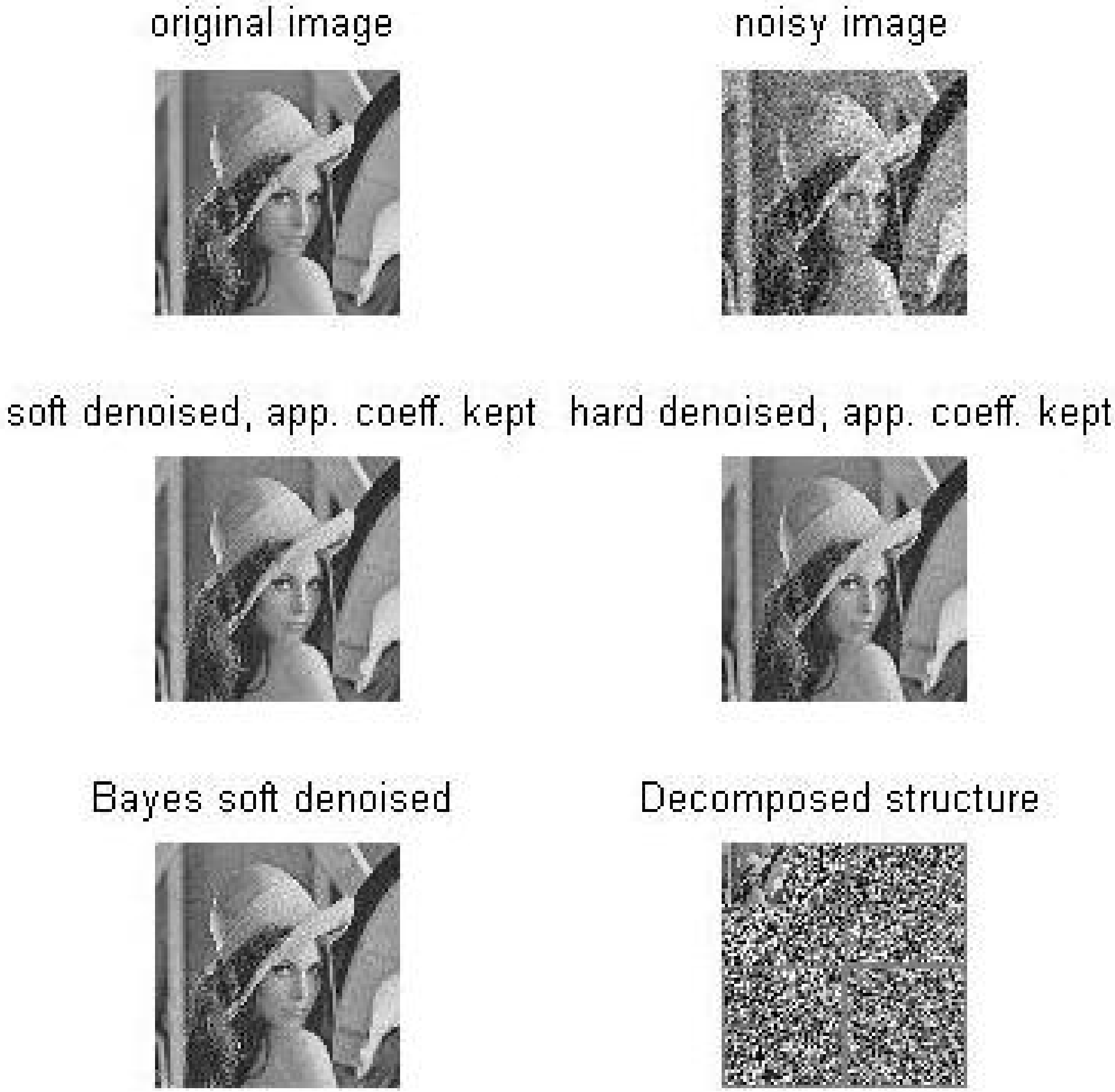

Figure 6.10: *Original image, noisy image(SNR=35.2193), and several denoised versions of it with wavelets* **Biorthogonal 6.8 at level 2** *(Global threshold=133.3075 and Bayes thresholds: H=61.4229, V=52.5046, D=110.0743): Soft-denoised keeping app. coeff.(SNR=35.6836), Hard-denoised keeping app. coeff.(SNR=35.8594) and Bayes soft-denoised (SNR=36.603).*

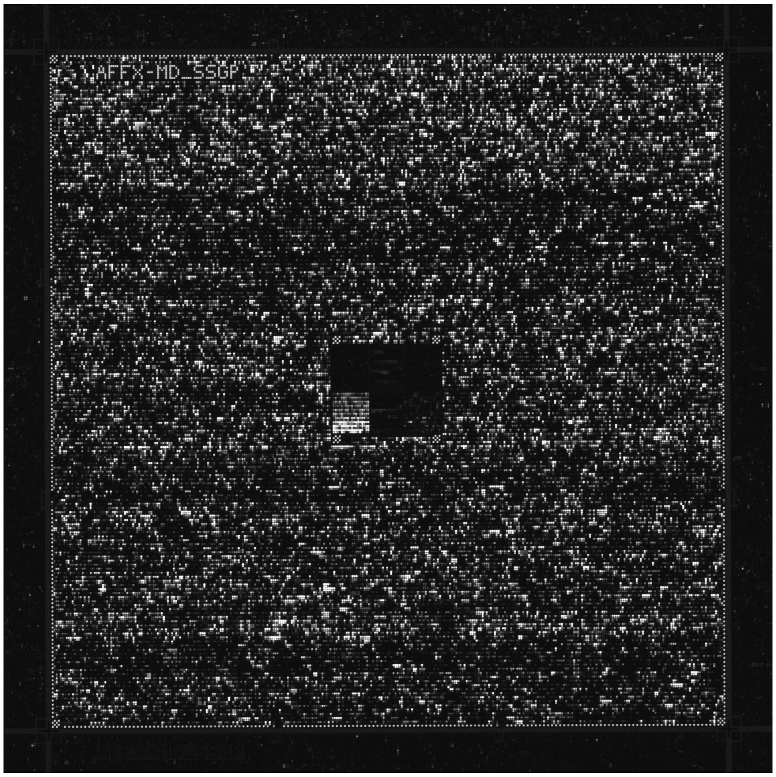

Figure 6.11: *Image of Wheat (Molly genotype) at one day after Hessian fly attack*

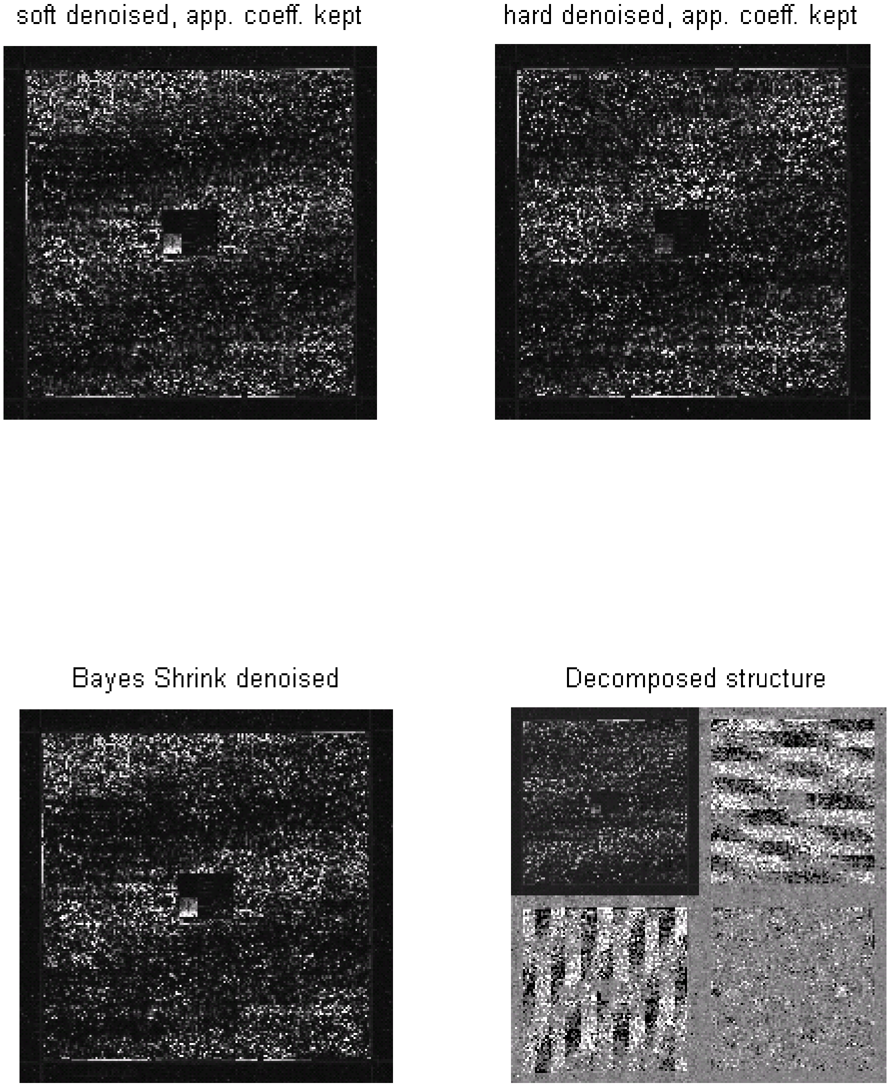
soft denoised, app. coeff. kept
hard denoised, app. coeff. kept
Bayes Shrink denoised
Decomposed structure

Figure 6.12: *Several denoised versions of Wheat with wavelets* **Daubechies 4 at level 1** *(Global threshold=23.8623 and Bayes thresholds: H=0.4710, V=0.4666, D=1.9850).*

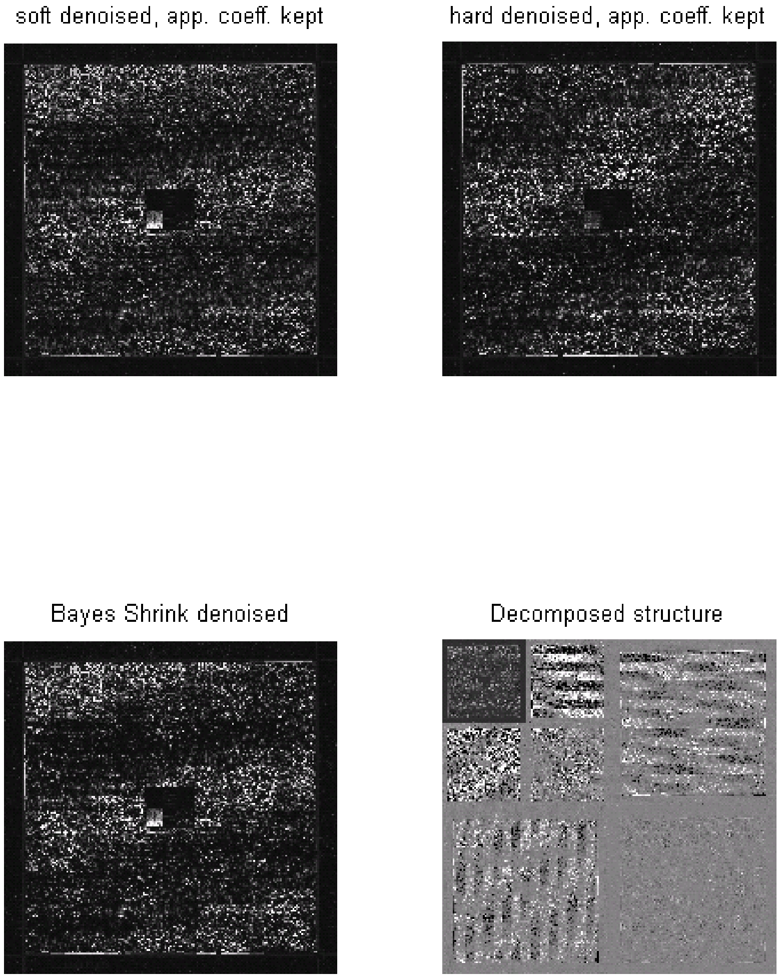
soft denoised, app. coeff. kept
hard denoised, app. coeff. kept
Bayes Shrink denoised
Decomposed structure

Figure 6.13: *Several denoised versions of Wheat with wavelets* **Daubechies 4 at level 2** *(Global threshold=23.8623 and Bayes thresholds: H=0.4710, V=0.4666, D=1.9850).*

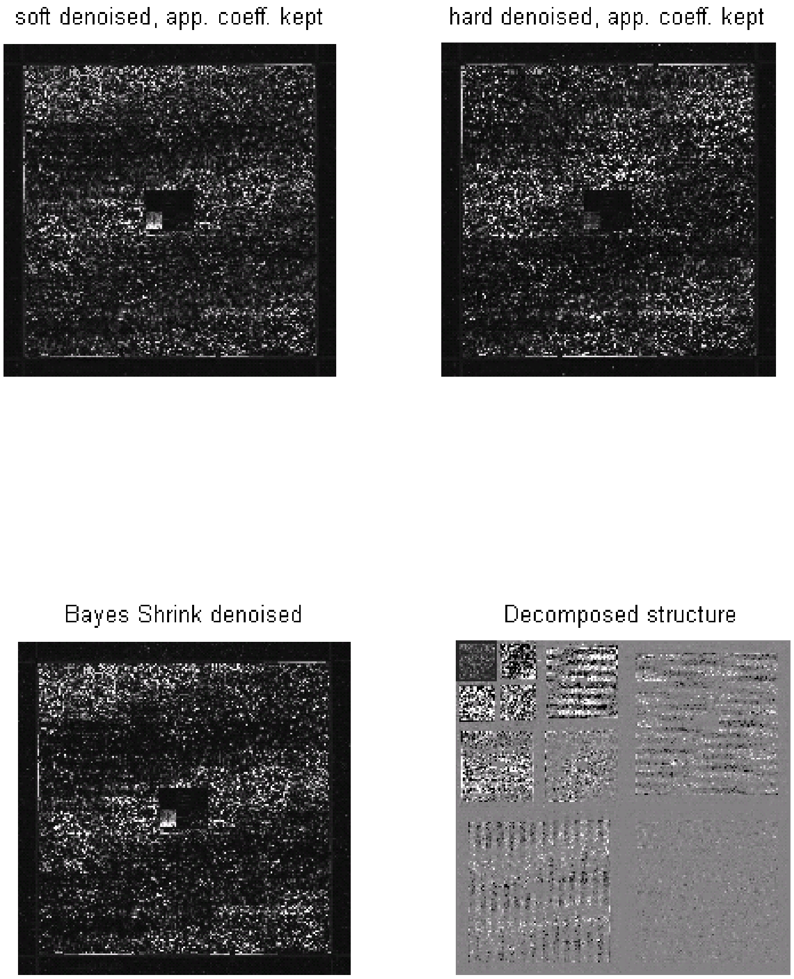
soft denoised, app. coeff. kept
hard denoised, app. coeff. kept
Bayes Shrink denoised
Decomposed structure

Figure 6.14: *Several denoised versions of Wheat with wavelets* **<u>Daubechies 4 at level 3</u>** *(Global threshold=23.8623 and Bayes thresholds: H=0.4710, V=0.4666, D=1.9850).*

Conclusion

The main work in this report has been to demonstrate an implementation of two image processing techniques, namely, “Bayes classification” and “Wavelet denoising” along with the related theories. Although examples of actual images being processed has been dealt with in detail, a considerable amount of effort has also been put toward developing a self-contained theory on these techniques.

Before Bayesian classification is exemplified, some theory on Bayesian statistics and Bayesian estimator has been presented. Having developed some related theory as well as other suitable concepts, Bayes classification method has been described and illustrated with an example. A satellite image with four easily identifiable classes, namely, sand, settlement, vegetation and road has been chosen for this purpose. Using some training samples of each class from this image, a Bayes recognition system has been developed. This system is checked for stability with independent samples and then applied to a randomly selected region of the image for classification. The procedure for developing a Bayes recognition system as well as the implementation of the system has been illustrated with plenty of figures. The results of classification by the developed recognition system have been summarized in a table. All necessary MATLAB codes that have been used to generate this illustrated example are presented.

The other image processing technique that has been studied is image denoising via wavelet thresholding. Since this technique dealt with wavelet theory, some background in Fourier Analysis was deemed inevitable to describe what wavelets are and how they work. A good amount of effort has been put toward a detailed review of some Fourier Analysis results and their application in solving some selected problems in Stephen Mallat’s book titled *“A wavelet tour of signal processing - the sparse way”*[8]. This book has been the primary source of material as many statements, theorems, proofs as well as figures in this report have been taken from this book whose online source is http://www.ceremade.dauphine.fr/ peyre/wavelet-tour/. After developing a wavelet theory required for our image denoising technique, implementation of this technique on noisy images using MATLAB has been described with several illustrations. An original image of “Lena” has been used for this purpose. The original image has been corrupted with some randomly generated Gaussian white noise. Three different thresholding schemes, namely, soft threholding, hard thresholding and Bayes Shrink thresholding has been selected for comparison. Among these three, it is seen that Bayes Shrink thresholding consistently performs the best. This is expected since it takes into account the prior distribution of wavelet coefficients and the variability of these coefficients in three different orientations, namely, horizontal, vertical and diagonal, and assigns three different thresholds accordingly as opposed to a common threshold for all the orientations in the cases of Global soft and hard thresholding. Also, a number of different wavelets have been used in the denoising process. We can already find a huge pool of commonly-used wavelets in MATLAB which differ in shape and size. Due to this variation, different wavelets capture different local features of an image. Since local features are subtle characteristics and they vary from image to image, we do not gain any insight by comparing different wavelets. Thus, the observation of the denoising effect that we have done using a number of select wavelets is merely for the sake of illustration and not for comparing between these wavelets. However, we can compare the denoising results with Daubechies 4 wavelets for different levels of decomposition. The results show that the higher the decomposition level, the better is the denoising with Bayes Shrink thresholding scheme. This is due to the fact that more subtle features of the image can be read by more wavelet coefficients at higher levels.

Bibliography

[1] S. M. Berman, *Sojournes and extremes of stochastic processes*, Wadsworth, 1989.

[2] D. Donoho and I. Johnstone, *Ideal spatial adaptation via wavelet shrinkage*, Biometrika **81(3)** (1994), 425–455.

[3] Woods R. E. Gonzalez, R. C. and S. L. Eddins, *Digital image processing using matlab*, Prentice Hall, 2004.

[4] Gudmund R. Iversen, *Bayesian statistical inference*, Sage University Paper series on Quantitative Applications in the Social Sciences (1984), 07–043.

[5] Peter M. Lee, *Bayesian statistics: An introduction*, Oxford, 1989.

[6] S Mallat, *An efficient image representation for multiscale analysis*, Proc.Machine Vision Conference (1987).

[7] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of l^2*, Trans.Amer.Math.Soc. **315** (1989), 69–87.

[8] Stephane Mallat, *A wavelet tour of signal processing, the sparse way*, 3rd ed., Elsevier, 2009.

[9] Y. Meyer, *Wavelets and operators: Advanced mathematics*, Cambridge University Press, 1992.

[10] B. Yu S. G. Chang and M. Vetterli, *Adaptive wavelet thresholding for image denoising and compression*, IEEE Transactions on Image Processing **9** (2000), 1532–1546.

[11] M. Vetterli, *Filter banks allowing perfect reconstruction*, Signal Proc. **10(3)** (1986), 219–244.

[12] Gerhald Winkler, *Image analysis, random fields and dynamic monte carlo methods*, Springer, 1995.