Use of operator theory and sub-band filters in the analysis and encoding of signals and images

Le Gui
University of Iowa

Copyright 2009 Le Gui

This dissertation is available at Iowa Research Online: https://ir.uiowa.edu/etd/293

Recommended Citation
https://doi.org/10.17077/etd.w94el458
USE OF OPERATOR THEORY AND SUB-BAND FILTERS IN THE ANALYSIS
AND ENCODING OF SIGNALS AND IMAGES

by

Le Gui

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and Computational Sciences
in the Graduate College of
The University of Iowa

July 2009

Thesis Supervisor: Professor Palle Jorgensen
ABSTRACT

The thesis is motivated by recent advances in signal and image processing, a part of electrical and computer engineering.

In the first part, we begin with a new approach to the mathematical signal processing as used in the digital processing of images. We prove such results in the 2D case, and we explain their use. A key point we explore is the interplay between the two cases, continuous and discrete. Our discrete algorithms present fast matrix-operations to be applied to images in pixel form. This part of the thesis in turn is based on tools from wavelet analysis, and more generally from the theory of operators in Hilbert space.

In the second part, we address encoding and quantization of wavelet coefficients obtained after applying the DWT (mentioned in first part) to 1-D signals. This is the last crucial step in A/D conversion, i.e., analog to digital. By quantization we mean the conversion and encoding of processing-output into bits; bits that in turn are transmitted and fed into a decoder. We isolate and make mathematically precise a particular family of quantizers which are efficient in that they produce error terms of exponential fall-off. We do this with a family of discrete algorithms, each one governed by a quantizer. In Theorems 3.2, 3.5, 3.11, we obtain quite precise a priori estimates.

In the last part, we address the compression of a matrix (a 2-D image) obtained by applying the DWT on an image mentioned in the first part. Embedded Zerotree
Wavelet algorithm is introduced and implemented.

Abstract Approved: ____________________________
Thesis Supervisor

______________________
Title and Department

______________________
Date
USE OF OPERATOR THEORY AND SUB-BAND FILTERS IN THE ANALYSIS
AND ENCODING OF SIGNALS AND IMAGES

by

Le Gui

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Applied Mathematical and Computational Sciences
in the Graduate College of
The University of Iowa

July 2009

Thesis Supervisor: Professor Palle Jorgensen
Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Le Gui

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Applied Mathematical and Computational Sciences at the July 2009 graduation.

Thesis Committee:  
Professor Palle Jorgensen, Thesis Supervisor

Tuong Ton-That

Bor-Luh Lin

Surjit Khurana

William Klink
To my Mom and Grandma especially, and all the rest of my family who have given me so much support......
To Dr. Palle Jorgensen in appreciation of his help and encouragement......
To Joe Frondle for being there for me in so many ways......
Problems worthy of attack prove their worth by hitting back.

—Piet Hein

One cannot expect any serious understanding of what wavelet analysis means without a deep knowledge of the corresponding operator theory.

—Yves Meyer
ACKNOWLEDGMENTS

A dissertation is far more than the finished product one presents for defense in hopes of receiving their doctorate. It is a literary offering-up to which the author alone proudly attaches his/her name in declaration to the world that “I have conquered the mountain so to speak.” This is misleading though, and left uncorrected the dissertation stands as a fraud of sorts, a perpetual rouse of deceit played upon all who read it.

This dissertation is in great part the result of efforts, kindnesses, and encouragements beyond those which I can call my own. I would be forever remiss in failing to recognize all those whose individual and collective signatures can be found within my dissertation, and to whom I am extremely grateful.

First I want to thank my advisor, Dr. Palle Jorgensen, for his assistance and friendship throughout my 6 years at UI, and especially over the last few months. I wish to thank my parents, also my brother and sister, and the rest of my family who have always supported me during my time here. Also Dr. Yi Li who initially recruited me to come to UI, as well as professors Dr. Bor Luh Lin, Dr. Surjit Khurana, Dr. Tuong Ton That who have all helped in so many ways. Finally I want to thank Joe Frondle for all his help over the past several weeks, and also my friends in the math department Kevin, Ben, Mel, Leonita who have made the process much more bearable. This dissertation bears my name but would not have been possible without all of you. Thank you very much.
## TABLE OF CONTENTS

**LIST OF TABLES** .................................................................................. vii

**LIST OF FIGURES** ............................................................................. viii

**CHAPTER**

1 INTRODUCTION ................................................................................ 1

1.1 Overview ....................................................................................... 1
1.2 Scope of Thesis ............................................................................. 2

2 MULTI-RESOLUTION ANALYSIS & WAVELET DECOMPOSITION .... 6

2.1 Fourier Transform and Gabor Transform ................................. 6
   2.1.1 Fourier Transform ............................................................... 6
   2.1.2 Gabor Transform ............................................................... 7
2.2 Wavelets and Wavelet Transform ............................................. 8
   2.2.1 Wavelets ........................................................................... 8
   2.2.2 Some Well-Known Wavelets ............................................ 10
   2.2.3 Continuous Wavelet Transform ....................................... 14
   2.2.4 Discrete Wavelet Transform ............................................ 16
2.3 1-D Multi-Resolution Analysis and Mallat Algorithm ............ 17
   2.3.1 Multi-Resolution Analysis ................................................. 17
   2.3.2 Mallat Algorithm .............................................................. 24
   2.3.3 Connection With Sub-band Filtering Schemes .................. 27
   2.3.4 Mathematical Insight ....................................................... 31
2.4 2-D Multi-Resolution Analysis and Mallat Algorithm .......... 35
   2.4.1 2-D MRA and Mallat Algorithm ..................................... 35
   2.4.2 2-D Image Wavelet Transform ........................................ 36

3 QUANTIZATION ERRORS IN DIFFERENT A/D CONVERSIONS .... 41

3.1 Introduction ................................................................................. 41
3.2 A/ D Conversion .......................................................................... 41
   3.2.1 Sampling ........................................................................... 41
   3.2.2 Quantization ..................................................................... 44
      3.2.2.1 Pulse Code Modulation(PCM) .................................... 45
      3.2.2.2 Sigma-Delta Modulation ........................................... 50
      3.2.2.3 Beta Expansion (Beta Encoder) ................................. 55
3.3 Summary ...................................................................................... 60
# LIST OF TABLES

4.1 3-Level DWT Coefficients .................................................. 74
4.2 Process on the First Dominant Pass ........................................ 78
4.3 Process of the First Subordinate Pass ..................................... 79
4.4 The Second Dominant Pass .................................................. 80
4.5 Process of the Second Subordinate Pass ................................... 81
# LIST OF FIGURES

1.1 Process of DSP .................................................. 1

2.1 Time-Frequency Window of Wavelet Transform .......................... 11

2.2 Haar Wavelet and its Fourier Transform .............................. 12

2.3 Morlet Wavelet: (a) $\psi(t)$; (b) $\hat{\psi}(\omega)$ .................. 12

2.4 Coif4 Wavelet Function and Scaling Function ....................... 15

2.5 An Example of MRA Conception ..................................... 19

2.6 Relation Between $V_j$ and $W_j$ .................................. 20

2.7 The Decomposition of Discrete Signal in Mallat Algorithm ............ 26

2.8 The Reconstruction of Discrete Signal in Mallat Algorithm .......... 26

2.9 Diagram of Decomposition & Reconstruction ......................... 28

2.10 Decomposition & Reconstruction in Terms of Operators [26] ............ 29

2.11 Diagram of Decomposition in Terms of $S_0^*$ and $S_1^*$ .......... 30

2.12 Shape of $S$ and $S^*$ ............................................ 30

2.13 Relation Between $l^2(Z)$ and $L^2(T)$ ............................. 34

2.14 Diagram of 2-D Image Decomposition ............................... 37

2.15 The Sub-band Structure After Applying 3-Level DWT ................. 39

2.16 Decomposition and Reconstruction of a Test Image ................ 40

3.1 Sampling .......................................................... 42

3.2 The Normalized Sinc Function: It Shows the Central Peak at $x = 0$, and Zero-Crossings at the Other Integer Values of $x$. ........... 44

4.1 The Process of Image Processing .................................... 62
4.2 Steps For Coder .......................................................... 66
4.3 2-Level DWT Decomposition ................................. 67
4.4 Spatial Orientation Tree for 3-Level DWT .......... 67
4.5 Decision Tree for Encoding a Coefficient .......... 71
4.6 Scanning Order for Encoding Coefficients ......... 72
4.7 From Top Left to Bottom Right, (1), (2), (3), (4), (5), (6) ......... 77
CHAPTER 1
INTRODUCTION

1.1 Overview

Since the 1960’s, Digital Signal Processing technology (DSP) came into being and has led to the formation of an independent disciplinary system due to the rapid development of the computer and information theory. Now digital signal processing is everywhere in our daily life, from ipod music players to electronic music synthesizers, to the sound cards in our PCs. We also hear about "DSP chips", "oversampling digital filters", "D/A converters" etc. What is DSP? DSP is a process applied to a signal so that useful information contained in the signal can be obtained. The DSP process consists of the acquisition, transformation, extraction, enhancement, transmission, identification, integration and storage etc.

Here is the diagram of this process. See VP 2003.

![Figure 1.1: Process of DSP](image)
• PrF: Pre-filter, which is used to remove the unwanted information;

• A/D: Analog-Digital Converter, which produces a stream of binary numbers;

• DSP: Digital Signal Processor. It can represent a computer or digital hardware;

• D/A: Digital-Analog converter, which is an inverse operation to A/D;

• PoF: Post-filter. It has same function as PrF.

1.2 Scope of Thesis

The thesis is about four areas of mathematics which go into applications dealing with signal and image processing: (1) wavelet algorithms, (2) sampling, (3) quantization, (4) compression. We are always dealing with two kinds of signals, analog and digital, and the conversion of analog to digital, abbreviated A/D.

Our emphasis will be on parts (3) and (4), but we will include the parts of the first two areas of mathematics which will be needed for the new material.

Part (1), wavelet theory, includes both the continuous case and the discrete. But it is the discrete wavelet algorithms which yield matrix algorithms for the processing of signals/images. This entails subdividing signals into bands. This subdivision refers to a mathematical duality, called time frequency. This however is an abstraction of wider generality than a single time variable. First time can be discrete and continuous, and it can have multiple dimensions: a speech signal for example is modeled with time being a multiple of a sample size. In this case, we will have a periodic frequency variable, and we will be using Fourier series with this period.
Hence our first chapter will cover the part of Fourier analysis we will use later in the thesis. A good deal of functional analysis and operators in Hilbert space will be needed as well.

For a fixed period interval, there is a part of interval close to zero, and this sub-interval will be called the low frequency band. We will be introducing operators which "pass" the low frequencies, the low-pass filters. A system of filters used in processing will involve a filter for each of a finite selection of bands. The case of two bands is called "quadrature-mirror."

We show that there is a delicate interplay between the two cases, continuous and discrete wavelet algorithms, and in fact this is the first instance of the two sides, analog vs digital as it is applied in signal processing. But there is one more part of the processing, the conversion of the output numbers into bits to be processed in computer programs (called quantization). Examples include such processing algorithms as MATLAB programs, as they are used in the processing of pixel numbers in an image from an electronic exposure.

Indeed, signals with two variables (2D) are used in the processing of images, and mathematical models for the 2D case, as we show may be obtained from a tensor product procedure applied to the 1D case.

Part (2) sampling is always part of a mathematical model for the processing of signals and for A/D conversion. We will review Shannon’s optimal sampling formula for signals with a fixed and finite frequency band, and we will be studying motivation for, precise definition of, and applications of "oversampling." One of the applications
includes part (3) quantization. (This does not refer to quantum theory in physics, but rather to the use of the word in computer science.)

Part (3) quantization. When an algorithm such as a discrete wavelet/matrix algorithm is applied to a particular analog signal, the output must be converted into bits that are processed in the computer, and hence we are faced with making optimal choices for this part of the A/D conversion; and by "optimal" we refer to such things as run-time, computer memory, and best approximation. Hence we are faced with the design of quantization/encoding schemes that are the best possible in these regards.

A popular tool for quantization goes by the name sigma-delta (\( \Sigma \Delta \)) modulation, and in chapter 3 we develop a scheme for its realization of data conversion within the framework mentioned above.

Wavelet algorithms allow us to subdivide by "frequency" into resolution bands, and modulation is a method for encoding high resolution signals into lower resolution signals. It uses what is also called pulse-density modulation, and it has found use in the design of electronic components, such as analog-to-digital and digital-to-analog converters, frequency synthesizers, and switched-mode power supplies. A circuit which implements this technique can achieve very high resolutions while using low-cost CMOS processes, such as the processes used to produce digital integrated circuits. It is only in recent years that it has come into widespread use with improvements in silicon technology. Almost all current analog integrated circuit vendors offer delta-sigma converters.

Part (4) compression. We will mention the motivation feasibility of image com-
pression. In addition, we will talk about an image compression algorithm—*Embedded Zerotree Wavelet (EZW)*. It is based on a new data structure—*Zerotree* of wavelet coefficients and the fact that the magnitude of wavelet coefficients will decrease if the frequency of wavelet band increases.

The four parts of the thesis rely on the literature in different ways.

In section 2, we present the part of harmonic analysis and wavelet theory that will be needed for our main conclusions. Our references for harmonic and functional analysis are [24], [9], [10], [12]; for wavelets, [3], [1], [4], [6], [18], [23], [31], [20], [21], [30]. In the main body of the thesis, we rely on the current literature as follows: Tools from stochastic integration and estimation [14], [15], [17]; and from selected applications [7], [8], [2], [19], [25], [5], [11], [13], [16], [22], [26], [27], [28], [29].
2.1 Fourier Transform and Gabor Transform

2.1.1 Fourier Transform

Fourier analysis is a field that almost every scientific worker is happy to use as a mathematical tool. At present, Fourier’s ideas and methods are widely used in linear programming, geodesy, as well as telephone, radio, X-ray and other scientific instruments that are difficult to count. It is a platform for development both in basic science and applied scientific research. Therefore, the physicist James Clark Maxwell praised the Fourier analysis as a great epic of mathematics.

Fourier Analysis defines the “Frequency” concept. We can analyze the distribution of signal t in different frequencies.

**Definition 2.1.** If \( f(t) \in L^2(R) \), The Continuous Fourier Transform of \( f(t) \) is the function \( F(\omega) \) defined by letting

\[
F(\omega) = \hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt
\]  

(2.1)

Its Inverse Fourier Transform is defined as:

\[
f(t) = \hat{F}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega)e^{i\omega t}d\omega
\]

(2.2)

\( \hat{f}(\omega) \) is called the Fourier Transform of \( f(t) \). When the function \( f(t) \) is periodic function, \( \hat{f}(\omega) \) is also called Fourier Series of \( f(t) \). Function \( f(t) \) and its Fourier Transform \( \hat{f}(\omega) \) are two different representations of the same signal. \( f(t) \) shows the
time-domain information of this function and hides the frequency-domain information; on the other hand, \( \hat{f}(\omega) \) only shows the frequency-domain information of this function[31], [6].

The Fourier Transform is a tool to transform functions between time-domain and frequency-domain. The essential idea of Fourier Transform is that any periodic function, regardless of how irregular it is, can be represented as the sum of a sequence of sine functions and cosine functions. For non-periodic functions, we can convert them to periodic ones by Average Procedure [3], [6]. Fourier Transform is very good at processing stable signals.

2.1.2 Gabor Transform

For some irregular signals, like speech, music etc, their frequency-domain information changes very fast within a short period of time compared to the regular signals. For these types of signals, Fourier Transform can only tell us different frequency information about the signal. It can’t tell us when the different frequency information will show up. As we know, frequency and the length of period of signals are inversely related, which means we can give more accurate approximation for high frequency signals over a relatively short period of time. For low frequency signals, we want to have a long period of time. Therefore, we need a flexible “time-frequency window”. Fourier Transform doesn’t have time-frequency localization property. Therefore it can’t satisfy our request.

In order to study the frequency character of a signal within the scope of local
time-frequency domain, Gabor came up with a Windowed Fourier Transform in 1946. The idea is that we take the Fourier transform of a function $f(t)$ that is multiplied by a window function $g(t-b)$, for some shift $b$ called the center of the window, where $g(t)$ is a given smooth function with compact support and also called window function.

**Definition 2.2.** Let $g(t) \in L^2(R)$ is a given *Window Function*, Windowed Fourier Transform of $f(t)$ is defined as:

$$G_f(\omega, \tau) = \int_{-\infty}^{+\infty} f(t) g(t-\tau) e^{-i\omega t} dt$$ (2.3)

$G_f(\omega, \tau)$ represents the frequency information of $f(t)$ within local time interval $(\tau - \Delta \tau, \tau + \Delta \tau)$, see [12][10].

Unfortunately, this transform still does not solve the time-frequency localization problem! When the window function $g(t)$ is given, then the size and shape of the window are fixed. Therefore it doesn’t capture short “pulses” accurately, unless a very small window is used. But in that case, low-frequency content cannot be accurately captured. This is also due to the *Uncertainty Principle*: [12], [10], which states that a function cannot be simultaneously concentrated in both physical space and Fourier space.

### 2.2 Wavelets and Wavelet Transform

#### 2.2.1 Wavelets

A better solution to the previous problem in section 2.1.2 is to use wavelets!
**Definition 2.3.** A real or complex-value continuous-time function \( \psi(t) \) is a *Wavelet* or *Mother Wavelet* if it satisfies the following mathematical criteria:

- A wavelet must be square integrable or, equivalently, have finite energy:
  \[
  E = \int_{-\infty}^{+\infty} |\psi(t)|^2 \, dt < \infty
  \]
- A wavelet must satisfy the *Admissibility Condition*[1]:
  \[
  C_\psi = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} \, d\omega < \infty \tag{2.4}
  \]

**Definition 2.4.** A *Wavelet Family* is a collection of functions obtained by shifting and dilating the graph of a wavelet. Specifically, a wavelet family with mother wavelet \( \psi(t) \) consists of functions \( \psi_{a,b}(t) \) (*Analyzing Wavelet*) of the form:

\[
\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \tag{2.5}
\]

where \( a, b \in \mathbb{R}, a \neq 0 \). \( b \) is the location or center of \( \psi_{a,b}(t) \), \( a \) is the scale.

In general, the energy of wavelet function \( \psi(t) \) is accumulated at the origin, the energy of wavelet function \( \psi_{a,b}(t) \) is at the point \((b, 0)\).

Now let’s analyze why using wavelet functions is a better solution to the problem mentioned in section 2.1.2.

- Let’s define the window width of wavelet function \( \psi(t) \) to be \( \Delta t \), the center of window is at \( t_0 \), then the center of corresponding analyzing wavelet \( \psi_{a,b}(t) \) is at \( t_{a,b} = at_0 + b \), the width is \( \Delta t_{a,b} = a \Delta t \).
• Let $\hat{\psi}(\omega)$ be the Fourier Transform of $\psi(t)$, and its center of window in frequency-domain is at $\omega_0$, the width of window is $\Delta \omega$.

• Let $\hat{\psi}_{a,b}(\omega)$ be the Fourier Transform of $\psi_{a,b}(t)$. Then

$$\hat{\psi}_{a,b}(\omega) = a^{\frac{1}{2}} e^{-i\omega b} \hat{\psi}(a\omega)$$

• the center of window in frequency-domain of function $\hat{\psi}_{a,b}(\omega)$ is $\omega_{a,b} = \frac{1}{a}\omega_0$, the width $\Delta \omega_{a,b} = \frac{1}{a} \Delta \omega$.

• From the above, we can see that the width and center of function $\hat{\psi}_{a,b}(t)$ vary as the scale $a$ varies.

• If we call $\Delta t \cdot \Delta \omega$ the area of window corresponding to function $\psi(t)$, the area of window of $\psi_{a,b}(t)$ will not vary as $a$ and $b$ change since

$$\Delta t_{a,b} \cdot \Delta \omega_{a,b} = a \Delta t \cdot \frac{1}{a} \Delta \omega = \Delta t \cdot \Delta \omega$$

• By Uncertainty Principle [12], [10], we know that $\Delta t \cdot \Delta \omega \geq \frac{1}{2}$. It tells us that the window area $\Delta t \cdot \Delta \omega$ will not vary as a result of changes in $a$ and $b$. Therefore, when window area is fixed, there is a corresponding relationship between changes in $\Delta t$ and $\Delta \omega$. If $\Delta t$ is bigger, then $\Delta \omega$ must be smaller, and vice versa. See graphs 2.1.

2.2.2 Some Well-Known Wavelets

Haar Wavelet  
**Haar Wavelet** was introduced by mathematician A. Haar in 1910. It has compactly supported, orthogonal and symmetric properties.
It’s defined as:

\[ h(t) = \begin{cases} 
1 & 0 \leq t < \frac{1}{2} \\
-1 & \frac{1}{2} \leq t \leq 1 \\
0 & \text{otherwise} 
\end{cases} \]

Its Fourier Transform is:

\[ \hat{h}(\omega) = ie^{-i\omega/2} \frac{\sin^2(\omega/4)}{\omega/4} \]

Graphs of \( h(t) \) and \( \hat{h}(\omega) \) are in figure 2.2.

It’s simple definition is helpful for computing wavelet transforms. However, it is not as useful as other wavelets for analyzing continuous signals because it is not continuous. See [3], [31].

**Morlet Wavelet**

**Definition 2.5.** Morlet Wavelet is a complex wavelet. It is defined as:
Figure 2.2: Haar Wavelet and its Fourier Transform

\[ \psi(t) = e^{-t^2/2} e^{i\omega_0 t} \]

Where \( \omega_0 \) tells us where its Fourier Transform is centered, see the figure 2.3.

Its real part is expressed as:

\[ \psi(t) = e^{-t^2/2} \cos \omega_0 t \]

Its graph in time domain and frequency domain are shown in figure 2.3.

Figure 2.3: Morlet Wavelet: (a) \( \psi(t) \); (b) \( \hat{\psi}(\omega) \)
Mexican Hat Wavelet

**Definition 2.6.** It is the second derivative of Gaussian function.

\[ \psi(t) = \frac{2}{\sqrt{3\sqrt{\pi}}} (1 - t^2) e^{-t^2/2} \]

The shape of the function is like a Mexican hat. That is the reason why it is called Mexican Hat. It is useful for detection in computer vision.

**MATLAB** MATLAB is a numerical computing environment and fourth generation programming language. Maintained by The MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. See www.Wiki MATLAB.

MATLAB has Wavelet Toolbox. See www.mathworks.com. It extends the MATLAB technical computing environment with graphical tools and command-line functions for developing wavelet-based algorithms for the analysis, synthesis, denoising, and compression of signals and images. Wavelet analysis provides more precise information about signal data than other signal analysis techniques, such as Fourier.

The Wavelet Toolbox supports the interactive exploration of wavelet properties and applications. It is useful for speech and audio processing, image and video processing, biomedical imaging, and 1-D and 2-D applications in communications and geophysics.

When we start MATLAB, we type in “wavemenu” in the command window.
It will lead us to the “Wavelet Toolbox Main Menu”. From there, we may choose 1-D wavelet transform, 2-D wavelet transform etc.

Another useful MATLAB command is `wavefun`. It is used to plot the wavelet functions. The format is

\[
[\phi, \psi, Xval] = \text{wavefun}(\text{'wavename'}, \text{iter})
\]

where \(\text{iter}\) is the number of computations, and return wavelet function \(\psi\) and scaling function \(\phi\).

For instance, we can plot the \textit{coif}4 wavelet by the following MATLAB commands.

```matlab
iter = 10; wav = 'coif4';
[phi, psi, Xval] = wavefun(wav, iter);
subplot(121);
plot(Xval, psi); title('wavelet function')
subplot(122);
plot(Xval, phi); title('scaling function')
```

Here is graph 2.4 after we implement this code.

### 2.2.3 Continuous Wavelet Transform

The key idea of wavelet transform is to represent any arbitrary function \(f(t)\) as a combination of a set of wavelet functions \(\psi_{a,b}\) in equation (2.5).
Definition 2.7. Let $\psi(t) \in L^2(R)$, $\psi_{a,b}(t)$ is wavelet family defined by equation (2.5). For a given function $f(t) \in L^2(R)$, the CWT or Continuous-time Wavelet Transform of $f(t)$ with respect to a wavelet $\psi(t)$ is defined as: See Rao & Bopardikar 1998, [6].

$$W_f(a, b) = \langle f, \psi_{a,b} \rangle = |a|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} f(t) \psi \left( \frac{t-b}{a} \right) dt$$ (2.6)

A function $f(t)$ can be recovered from its wavelet transform via the resolution of the identity as follows. Given $f, g \in L^2(R)$, we have:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W_f(a,b) W_g(a,b) \frac{dadb}{a^2} = C_\psi \langle f, g \rangle$$ (2.7)

Equation (2.7) is equivalent to the following:

$$f = C_\psi^{-1} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W_f(a,b) \psi_{a,b} \frac{dadb}{a^2}$$ (2.8)
with convergence of the integral in the weak sense. See [6]. The CWT also has some properties like Linearity, Translation, Scaling etc. You may find them in Rao & Bopardikar 1999.

2.2.4 Discrete Wavelet Transform

In section 2.2.3, the wavelet function was defined at scale $a$ and $b$ as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t - b}{a}\right)$$

where scale $a$ and location $b$ are continuous values.

In practice, we need to discretize the values of the dilation and translation parameters, $a$ and $b$. This leads to the discrete wavelet transform (DWT).

Let $a = a_0^m$, $m$ is integer, $a_0 > 1$, $b = nb_0 a_0^m$, $b_0 > 0$, $n$ is integer. This idea represents wavelet transform as mathematical “zoom”. If you choose too big $a_0^m$, we can choose smaller $nb_0 a_0^m$.

**Definition 2.8.** Discrete Wavelet has the form

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \psi\left(\frac{t - nb_0 a_0^m}{a_0^m}\right) \quad (2.9)$$

**Discrete Wavelet Transform** of function $f(t)$ is defined as

$$Wf(m,n) = (f, \psi_{m,n}) = \int_{-\infty}^{+\infty} f(t) \psi_{m,n}(t) \, dt = a_0^{-m/2} \int_{-\infty}^{+\infty} f(t) \psi(a_0^m t - nb_0) \, dt \quad (2.10)$$

Common choices for discrete wavelet parameters $a_0$ and $b_0$ are 2 and 1 respectively. Substituting $a_0 = 2$, $b_0 = 1$ in equation (2.9), we see that discrete wavelet can
be written as:

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \psi\left(t - n2^m\right)$$

or, more compactly as

$$\psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m}t - n) \quad (2.11)$$

Discrete wavelets are commonly chosen to be orthonormal. These wavelets are both orthogonal to each other and normalized to have unit energy. This is expressed as: [1]

$$\int_{-\infty}^{+\infty} \psi_{m,n}(t) \psi_{m',n'}(t) dt = \delta_{m'm} \delta_{nn'} = \begin{cases} 1 & \text{if } m = m', n = n' \\ 0 & \text{otherwise} \end{cases}$$

The Inverse Discrete Wavelet Transform is given by:

$$f(t) = \sum_{m,n} \psi_{m,n}(t) Wf(m,n) \quad (2.12)$$

In next section, we will discuss the Multi-Resolution Analysis.

### 2.3 1-D Multi-Resolution Analysis and Mallat Algorithm

#### 2.3.1 Multi-Resolution Analysis

**Definition 2.9.** A Multi-Resolution Analysis [6] of $L^2(R)$ consists of an infinite nested sequence of subspaces of $L^2(R)$

$$\cdots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \cdots \subset V_j \subset V_{j-1} \subset \cdots$$

with properties
1. $\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R})$

2. $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$

3. $f(t) \in V_j \iff f(2t) \in V_{j-1}$, for all $j \in \mathbb{Z}$

4. $f(t) \in V_0 \iff f(t - k) \in V_0$, for all $k \in \mathbb{Z}$

5. There exists a function $\varphi \in V_0$ such that the collection $\{\varphi(t - k) : k \in \mathbb{Z}\}$ forms an orthogonal basis\(^1\) of $V_0$, i.e.

$$V_0 = \text{span} \{\varphi(t - k)\}$$

$$\langle \varphi(t - m), \varphi(t - n) \rangle = \delta_{m,n}$$

**Definition 2.10.** Function $\varphi(t) \in L^2(\mathbb{R})$ satisfying the condition (5) in MRA definition is called *scaling function*. As with *Wavelet Family* we mentioned before, if we do dilation and shift to *scaling function* $\varphi(t)$, we will get the scaling function family $\varphi_{j,k}(t)$:

$$\varphi_{j,k}(t) = 2^{-j/2} \varphi(2^{-j} t - k) \quad (2.13)$$

**Remark 2.1.** (a) We can denote function $\varphi(t - k)$ as $\varphi_{0,k}(t)$ by equation (2.13).

(b) From condition (3) in MRA definition, we can get:

$$f(t) \in V_j \iff f(2^j t) \in V_0$$

\(^1\) the requirement of orthogonality can be replaced by having a Riesz basis since we can construct an orthogonal basis from Riesz basis. See [6].
If \( \{ \varphi_{0,k}(t) \}_{k \in \mathbb{Z}} \) is an orthogonal basis of \( V_0 \), then \( \{ \varphi_{j,k}(t) \}_{k \in \mathbb{Z}} \) is an orthogonal basis of \( V_j \). In other words, at the different given scale level \( j \), scaling functions \( \{ \varphi_{j,k}(t) \}_{k \in \mathbb{Z}} \) span different subspace \( V_j \).

(c) Scale level \( j \) can be seen as the distance between a person and an object. As \( j \) increases, this person is further away from the object. As a result, the person’s accurate perception of the object decreases, which means less information about the object will be obtained. Vice versa. We can think of \( V_j \) as the approximation of this object at scale level \( j \). Then we can get \( V_j \subset V_{j-1} \), which is consistent with the MRA definition. See the figure 2.5.

![Figure 2.5: An Example of MRA Conception](image)

(d) Since subspaces \( \{ V_j \}_{j \in \mathbb{Z}} \) are contained inside of each other, the orthogonal bases of different spaces \( \{ V_j \}_{j \in \mathbb{Z}} \) are not orthogonal. Even though we have \( \bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}) \), \( \{ \varphi_{j,k}(t) \}_{j,k \in \mathbb{Z}} \) is NOT the orthonormal basis of \( L^2(\mathbb{R}) \).
What is the orthonormal basis of $L^2(R)$? How do we find it? MRA plays an important role of finding it by considering the difference between approximation at different scale levels.

Now, we introduce another sequence of subspaces $\{W_j\}_{j \in \mathbb{Z}}$ satisfying

\begin{align*}
V_{j-1} &= W_j \oplus V_j \quad (2.14) \\
W_j &\perp V_j \quad (2.15) \\
W_j &\perp W_{j-1} \quad (2.16)
\end{align*}

Subspace $W_j$ is also called wavelet space at scale level $j$. See figure 2.6. It shows the relation between $V_j$, $W_j$ and $L^2(R)$.

Figure 2.6: Relation Between $V_j$ and $W_j$

From equation (2.14), (2.15), and (2.16) we have a decomposition of $L^2(R)$:

$$L^2(R) = \bigoplus_{j \in \mathbb{Z}} W_j \quad (2.17)$$

Now, we are thinking that if we can construct a function called $\psi(t)$ in $W_0$, which plays the same role as $\varphi(t)$ in $V_0$, then we can do dilation and translation to $\psi(t)$ to form a sequence of functions $\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k)$
Is \( \{ \psi_{j,k}(t) \}_{j,k \in \mathbb{Z}} \) an orthonormal basis of \( L^2(R) \)? How can such a function be constructed? The following theorems answer these questions.

**Theorem 2.2.** For any orthogonal MRA with scaling function \( \varphi(t) \) satisfying those five conditions in MRA definition, then there exists a function \( \psi(t) \in L^2(R) \) so that \( \{ \psi(t-k) \}_{k \in \mathbb{Z}} \) forms an orthogonal basis of \( W_0 \), and \( \{ \psi_{j,k}(t) \}_{j,k \in \mathbb{Z}} \) forms an orthonormal wavelet basis of \( L^2(R) \).

*Proof.* Proof can be found in [6, 18].

**Theorem 2.3.** For any orthogonal MRA associated with scaling function \( \varphi(t) \), there exists two \( l^2(R) \) sequences of coefficients \( \{ h_k \}_{k \in \mathbb{Z}} \) and \( \{ g_k \}_{k \in \mathbb{Z}} \), called Masking Coefficients and Wavelet Coefficients respectively, such that \( \varphi(t) \) and \( \psi(t) \) can be represented as

\[
\varphi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2t - k) \tag{2.18}
\]

\[
\psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \varphi(2t - k) \tag{2.19}
\]

in \( L^2(R) \). Equations (2.18) and (2.19) are called two-scale equation (or refinement equation) for function \( \varphi(t) \) and \( \psi(t) \) respectively. In addition, \( \{ h_k \}_{k \in \mathbb{Z}} \) and \( \{ g_k \}_{k \in \mathbb{Z}} \) can be calculated by:

\[
h_k = \langle \varphi(t), \varphi(2t - k) \rangle \tag{2.20}
\]

\[
g_k = \langle \psi(t), \varphi(2t - k) \rangle \tag{2.21}
\]

*Proof.* You may find it in [31].
Remark 2.4. Equation (2.18) and (2.19) in frequency domain can be written as

\[ \hat{\phi} (\omega) = m_0 (\frac{\omega}{2}) \cdot \hat{\phi} (\frac{\omega}{2}) \]  

(2.22)

\[ \hat{\psi} (\omega) = m_1 (\frac{\omega}{2}) \cdot \hat{\phi} (\frac{\omega}{2}) \]  

(2.23)

where

\[ m_0 (\omega) = \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} h_k e^{-i\omega k} \]  

(2.24)

\[ m_1 (\omega) = \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} g_k e^{-i\omega k} \]  

(2.25)

\( m_0 \) is called low-pass filter, \( m_1 \) is called high-pass filter.

Theorem 2.5. Let \( \{V_j\}_{j \in \mathbb{Z}} \) be a MRA with scaling function \( \varphi (t) \) and masking coefficients \( \{h_k\}_{k \in \mathbb{Z}} \). The coefficients \( \{g_k\}_{k \in \mathbb{Z}} \) in equation (2.19) can be obtained by

\[ g_k = (-1)^k h_{1-k} \]  

(2.26)

Proof. Proof can be found in [3, 6, 31].

Remark 2.6. Theorem 2.5 tells us a method to construct an orthonormal basis for \( L^2 (\mathbb{R}) \).

- By equation 2.18, we can determine function \( \varphi (t) \);
- By equation 2.20, we can have \( \{h_k\}_{k \in \mathbb{Z}} \);
- By equation 2.26, we can get \( \{g_k\}_{k \in \mathbb{Z}} \);
- By equation 2.19, we can get function \( \psi (t) \);
• By \( \psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t-k) \), we can get \( \{\psi_{j,k}(t)\}_{j,k \in \mathbb{Z}} \), which is an orthonormal basis for \( L^2(R) \).

**Definition 2.11.** For each \( j \in \mathbb{Z} \), define the approximation operator \( P_j \) on function \( f(t) \in L^2(R) \) at scale \( j \) by

\[
P_j f = \sum_k \langle f, \varphi_{j,k} \rangle \varphi_{j,k} = \sum_k c_{j,k} \varphi_{j,k}
\]

(2.27)

Where

\[
c_{j,k} = \langle f, \varphi_{j,k} \rangle
\]

c\(_{j,k}\) is called *Scaling Coefficients*. \( P_j f \) is called an approximation to \( f(t) \) at scale \( j \).

For each \( j \in \mathbb{Z} \), define the detail operator \( Q_j \) on function \( f(t) \in L^2(R) \) at scale \( j \) by

\[
Q_j f = P_{j-1} f - P_j f = \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k} = \sum_k d_{j,k} \psi_{j,k}
\]

(2.28)

where

\[
d_{j,k} = \langle f, \psi_{j,k} \rangle
\]

d\(_{j,k}\) is also called *Wavelet Coefficients*. See [31, 3].

**Remark 2.7.** [18, 3]

(i) The approximation operator \( P_j \) is an orthogonal projection of function \( f(t) \in L^2(R) \) onto subspace \( V_j \), i.e \( V_j = \{ f \mid P_j f = f \} \);

(ii) For all \( f(t) \in C^0_c(R) \), \( \lim_{j \to \infty} \|P_j f - f\|_2 = 0 \). Proof is in [3, 31].

(iii) \( Q_j \) is also an orthogonal projection of function \( f(t) \) onto subspace \( W_j \), i.e \( W_j = \{ f \mid Q_j f = f \} \)

\[
Q_j^2 = P_{j-1}^2 - P_{j-1}P_j - P_jP_{j-1} + P_j^2 = Q_j
\]
(iv) A MRA provides a sequence of approximations $P_j f$ of increasing accuracy to a given function $f(t)$.

Because of equation (2.14) and (2.17), we can decompose a given function $f(t) \in L^2(R)$ in terms of details at all resolution levels as [18, 22]

$$f = \sum_{j=\infty}^{j=-\infty} Q_j f$$  \hspace{1cm} (2.29)

Alternatively, we can start at any level $l$ and use the approximation at scale level $j$ plus all the details at finer resolution:

$$f = P_l f + \sum_{j=l}^{\infty} Q_j f$$  \hspace{1cm} (2.30)

For practical applications, we need to reduce equation (2.30) to a finite sum. We assume that $f \in V_j$ for some $j > l$. Then

$$f = P_j f = P_l f + \sum_{k=l}^{j} Q_k f$$  \hspace{1cm} (2.31)

Equation (2.31) describes the DWT: the original function of signal $f(t)$ gets decomposed into a coarse approximation $P_l f$, and fine details at several scale levels.

2.3.2 Mallat Algorithm

From equations (2.27) and (2.28), we know for any function $f(t) \in V_{j-1}$, we have:

$$f(t) = \sum_k c_{j-1,k} \varphi_{j-1,k}(t)$$  \hspace{1cm} (2.32)

where $c_{j-1,k} = \langle f(t), \varphi_{j-1,k}(t) \rangle$. 
Since $V_{j-1} = V_j \oplus W_j$, function $f(t)$ can be decomposed as:

$$f(t) = \sum_k c_{j,k} \varphi_{j,k}(t) + \sum_k d_{j,k} \psi_{j,k}(t)$$  \hspace{1cm} (2.33)

where

$$c_{j,k} = \langle f, \varphi_{j,k} \rangle = \int_{\mathbb{R}} f(t)2^{-j/2}\varphi(2^{-j}t - k) \, dt$$ \hspace{1cm} (2.34)

$$d_{j,k} = \langle f, \psi_{j,k} \rangle = \int_{\mathbb{R}} f(t)2^{-j/2}\psi(2^{-j}t - k) \, dt$$ \hspace{1cm} (2.35)

By *two-scale equation* (2.18), we get:

$$\varphi(2^{-j}t - k) = \sqrt{2} \sum_n h_n \varphi(2^{-j+1}t - 2k - n)$$

Let $m = 2k + n$

$$\varphi(2^{-j}t - k) = \sqrt{2} \sum_m h_{m-2k} \varphi(2^{-j+1}t - m)$$  \hspace{1cm} (2.36)

Similarly, by Two-Scale Equation (2.19), we get:

$$\psi(2^{-j}t - k) = \sqrt{2} \sum_m g_m \varphi(2^{-i+1}t - m)$$  \hspace{1cm} (2.37)

Now substitute equations (2.36) and (2.37) into equations (2.34) and (2.35) respectively, we have:

$$\begin{cases} 
    c_{j,k} = \sum_m h_{m-2k} \cdot c_{j-1,m} \\
    d_{j,k} = \sum_m g_{m-2k} \cdot d_{j-1,m}
\end{cases}$$ \hspace{1cm} (2.38)

Equation (2.38) shows us that the *Scaling Coefficients* $c_{j,k}$ and *Wavelet Coefficients* $d_{j,k}$ at resolution level $j$ can be obtained from the *Scaling Coefficients* $c_{j-1,k}$ at resolution level $j - 1$. This is called *Mallat Algorithm*. See [3, 6]. This decomposition process is in figure 2.7.
If we consider \( c_0 \) as the original signal coefficient, then we can use equation (2.38) to decompose \( c_0 \) into \( d_1, d_2, \ldots, d_n \) and \( c_n \).

Of course, the reverse process also works. We can reconstruct the signal by

\[
c_{j-1,m} = \sum_k c_{j,k} h_{m-2k} + \sum_k d_{j,k} g_{m-2k}
\]

(2.39)

Figure 2.8 shows the reconstruction of a discrete signal.

The essential idea of \textit{Mallat Algorithm} is that we don’t have to know about \textit{Scaling Function} \( \varphi(t) \) and \textit{Wavelet Function} \( \psi(t) \). We can realize the decomposition and reconstruction of signal \( f(t) \) by only using the \textit{scaling coefficients} \( c_{j,k} \) and \textit{wavelet coefficients} \( d_{j,k} \). Further, they only depend on \textit{Masking Coefficients} \( \{h_k\}_{k \in \mathbb{Z}} \) because of the relation between \( \{h_k\}_{k \in \mathbb{Z}} \) and \( \{g_k\}_{k \in \mathbb{Z}} \) in equation (2.26). That is also the
reason why Mallat Algorithm is also called Fast Wavelet Transform (FWT).

2.3.3 Connection With Sub-band Filtering Schemes

In previous sections, we have talked about the Continuous Wavelet Transform and the Discrete Wavelet Transform for continuous signal $f(t) \in L^2(R)$. In practice, most signals are discrete signals like \( \{c(n)\}_{n \in \mathbb{Z}} \). The idea of Wavelet Transform of discrete signals is the same as filtering schemes in Digital Signal Processing.

In electrical engineering terms, equations (2.38) and 2.39 are the analysis and synthesis steps of a sub-band filtering scheme with exact Reconstruction. In a two-channel sub-band filtering scheme, an incoming sequence is convolved with two different filters, one low-pass and one high-pass. The two resulting sequences are then subsampled, i.e., only the even entries are retained. This is exactly what happens in equation (2.38). Going from one level in a multi-resolution analysis to the next coarser level and the corresponding level of wavelets, and then doing the reverse operation, can therefore be represented by a diagram. Here $h$, $g$ are low-pass filter and high-pass filter respectively and $\overline{(h)}_m = h_{-m}$, $\overline{(g)}_m = g_{-m}$. [6, 3, 22].

Remark 2.8. (i) In figure 2.9, if we denote the down sampling operator (decimation operator) by $D = 2 \downarrow$, then for any given sequence $\{c(n)\}_{n \in \mathbb{Z}}$, we have:

\[
(Dc)_n = \begin{cases} 
  c_n & \text{if } n \text{ is even} \\
  0 & \text{if } n \text{ is odd}
\end{cases}
\]

Down sampling is that we throw away the odd coefficients and renumber the even coefficients.
(ii) the *up sampling operator* is denoted by $U = 2 \uparrow$. It is defined by

$$(Uc)_n = \begin{cases} 
\frac{c(n)}{2} & \text{if } n \text{ is even} \\
0 & \text{if } n \text{ is odd} 
\end{cases}$$

*Up sampling* is that we interweave a zero between each pair of adjacent coefficients and then renumber them.

Hence the two operators $2 \uparrow$ and $h$ in figure 2.9 can be denoted by

$$S_0 = hU$$

Similarly $g$ and $2 \uparrow$ can be denoted by

$$S_1 = gU$$

The adjoint operator of $S_0$ is

$$S_0^* = (hU)^* = D\bar{h}$$

The adjoint operator of $S_1$ is

$$S_1^* = (gU)^* = D\bar{g}$$
Therefore the figure 2.9 becomes figure 2.10. We will explain the reason why $U^* = D$ later.

Figure 2.10: Decomposition & Reconstruction in Terms of Operators [26]

(iii) We can consider decomposition equation (2.38) as transformation rules corresponding to two operators $S_0^*$ and $S_1^*$, i.e. for given a sequence $\{c_k\}_{k \in \mathbb{Z}} \in l^2$, equation (2.38) can be rewritten as

\[
\begin{align*}
(S_0^* c)_m &= \sum_k \bar{h}_{k-2m} c_k \\
(S_1^* c)_m &= \sum_k \bar{g}_{k-2m} c_k
\end{align*}
\]  
(2.40)
where $S_0^*$ and $S_1^*$ are adjoint operators of $S_0$ and $S_1$ which are the following:

$$\begin{align*}
(S_0c)_m &= \sum_k h_{m-2k} c_k \\
(S_1c)_m &= \sum_k g_{m-2k} c_k
\end{align*}$$

(2.41)

Thus figure 2.7 becomes figure 2.11.

(iv) The matrix representations of operators $S_i^*$ and $S_i$ are in the shape of:

[26].

Figure 2.12: Shape of $S$ and $S^*$
These decomposition rules in (2.38) can also be interpreted as infinite matrix-vector products.

For a given discrete signal sequence \( \{c_n\}_{n \in \mathbb{Z}} \), the DWT is

\[
\begin{bmatrix}
    s_1 \\
    s_2 \\
    \vdots \\
    s_{N/2} \\
    d_1 \\
    d_2 \\
    \vdots \\
    d_{N/2}
\end{bmatrix} =
\begin{bmatrix}
    \tilde{h}_0 & \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 & 0 & \cdots & 0 \\
    0 & 0 & \tilde{h}_0 & \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 & 0 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 & 0 & \cdots & 0 \\
    0 & 0 & \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 & 0 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
    c_1 \\
    c_2 \\
    \vdots \\
    c_N
\end{bmatrix}
\]

(v) The transformation rules in (2.41) can be extended that: For any orthonormal basis of \( l^2(Z) \), equation (2.41) can give us a new orthonormal basis for \( l^2(Z) \). See [3].

(vi) The fact that equation (2.41) permutes ONB’s in \( l^2(Z) \) is equivalent to the usual conditions for the QMF’s [31] and they take the form of the following familiar operator relations:

\[
\begin{align*}
    S_i^* S_j &= \delta_{i,j} I \\
    \sum_i S_i S_i^* &= I
\end{align*}
\]

(2.42)

2.3.4 Mathematical Insight

In previous section, we have discussed up sampling \( U \), down sampling \( D \), operators \( S_0 \) and \( S_1 \)(they are also matrices) and slanted matrix etc. in \( l^2(Z) \). Because
of the isomorphism \( l^2(Z) \cong L^2(T) \) [3], we will have the same results in \( L^2(T) \) as those in \( l^2(Z) \).

Now, let’s take a look at the up sampling \( \check{U} \), down sampling \( \check{D} \), operator \( \check{S}_0 \) and \( \check{S}_1 \) in \( L^2(T) \). Recall that the isomorphism \( l^2(Z) \cong L^2(T) \) and choices involved in the isomorphism is Fourier Transform.

For a sequence \( \{c_n\}_{n \in \mathbb{Z}} \in l^2(Z) \), we apply Fourier Transform on \( c_n \), we get a function \( f(z) \) in \( L^2(T) \), i.e. \( f(z) = (Fc)(z) = \sum_{n \in \mathbb{Z}} c_n z^n \). It also satisfies an energy conservation relation known as Parseval’s formula, \( \sum_{n \in \mathbb{Z}} |c_n|^2 = \int_T |f|^2 \, dz \). See MV, Jk 1995 and [3].

Now let’s first discuss the up sampling operator \( \check{U} \) and down sampling operator \( \check{D} \) in \( L^2(T) \).

**Definition 2.12.** *Up Sampling Operator* \( \check{U} \) is defined as

\[
(\check{U}f)(z) = f(z^2) = \sum_{n \in \mathbb{Z}} c_n z^{2n}
\]  

(2.43)

where \( f(z) = \sum_{n \in \mathbb{Z}} c_n z^n \) is Discrete Fourier Transform. The key idea of up sampling \( \check{U} \) is same as operator \( U \) in \( l^2(Z) \).

*Down Sampling Operator* \( \check{D} \) is defined as

\[
(\check{D}f)(z) = \frac{1}{2} \sum_{\omega^2 = z} f(\omega)
\]

The key idea of down sampling \( \check{D} \) in \( L^2(T) \) is same as \( D \) in \( l^2(Z) \).

We also have relations between \( U \) and \( \check{U} \), \( D \) and \( \check{D} \):

\[
\check{D}F = FD
\]
\[ \tilde{U}F = FU \]

where \( f(z) = (Fc)(z) = \sum_{n \in \mathbb{Z}} c_n z^n. \)

**Lemma 2.9.** \( \tilde{D} = \tilde{U}^\ast \), i.e. the adjoint operator of Up Sampling \( \tilde{U} \) is Down Sampling \( \tilde{D}. \)

**Proof.** Claim \( \langle Db, c \rangle = \langle b, Uc \rangle \), indeed,

\[
\langle Db, c \rangle = \sum_{n \in \mathbb{Z}} b_{2n} c_n = \langle b, Uc \rangle, \forall b, c \in l^2(Z).
\]

since \( l^2(Z) \simeq L^2(T) \)

hence \( \langle \tilde{D}f, g \rangle = \langle f, \tilde{U}g \rangle, \forall f, g \in L^2(T) \)

then \( \tilde{D} = \tilde{U}^\ast \)

**Lemma 2.10.** If we define \((\tilde{S}_i f)(z) = m_i(z) f(z^2), i = 0, 1 \) in \( L^2(T), \) \( \tilde{S}_i \) can be identified with \( S_i \) using \( \tilde{SF} = FS \)

**Proof.** Note: from equation (2.43), we have:

\[
\tilde{S}_i f(z) = \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} h_k z^k \cdot \sum_{l \in \mathbb{Z}} c_l z^{2l} = \sum_{m \in \mathbb{Z}} \left( \sum_{l \in \mathbb{Z}} h_{m-2l} c_l \right) z^m
\]

By equations 2.24 on page 22 and 2.43 (note: here \( z = e^{-i\omega} \)), we have:

\[
(\tilde{S}_0 f)(z) = \sum_{k \in \mathbb{Z}} h_k z^k \cdot c_0 z^{2l} = \sum_{m \in \mathbb{Z}} \left( \sum_{l \in \mathbb{Z}} h_{m-2l} c_l \right) z^m
\]

(2.45)
by comparing the equation (2.41) and equation (2.45), we conclude that $\tilde{S} F = F S$.

As for $\tilde{S}_i^* (i = 0, 1)$, they are the adjoint operators of $\tilde{S}_i (i = 0, 1)$. By equation (2.44), we conclude:

$$\tilde{S}_i^* = (m_i \tilde{U})^* = \tilde{U}^* m_i = \tilde{D} m_i$$

which are also equivalent to $S_i^*$ for $i = 0, 1$.

Figure 2.13 is a diagram to show the relation $l^2(Z)$ and $L^2(T)$. From it, we also see that $\tilde{S} F = F S$.

Figure 2.13: Relation Between $l^2(Z)$ and $L^2(T)$. 
2.4 2-D Multi-Resolution Analysis and Mallat Algorithm

2.4.1 2-D MRA and Mallat Algorithm

2-D Multi-Resolution Analysis (MRA) One of the important applications of Wavelet Transform is on image processing, which is a 2 dimensional case. In the previous section, we have seen that 1-D MRA has four elements:

\[ 1 - D \text{ MRA} = \{ \varphi(t), \psi(t); (V_j)_{j \in \mathbb{Z}}, (W_j)_{j \in \mathbb{Z}} \} \]

What about the 2-D MRA? 2-D MRA and 2-D Mallat Algorithm can be extended from 1-D case by using Tensor Product. [3]

Definition 2.13. We say 2-D MRA also has four elements:

\[ 2 - D \text{ MRA} = \{ \Phi(x, y), \Psi(x, y); (V^2_j)_{j \in \mathbb{Z}}, (W^2_j)_{j \in \mathbb{Z}} \} \]

where

\[ V^2_j = V_j \otimes V_j \]

\[ \Phi(x, y) = \varphi(x)\varphi(y) \]

\[ \Psi(x, y) = \{ \psi^1(x, y) = \varphi(x)\psi(y), \psi^2(x, y) = \psi(x)\varphi(y), \psi^3(x, y) = \psi(x)\psi(y) \} \] (2.46)

\[ W^2_j = \underbrace{(V_j \otimes W_j)}_{\text{Horizontal}} \oplus \underbrace{(W_j \otimes V_j)}_{\text{Vertical}} \oplus \underbrace{(W_j \otimes W_j)}_{\text{Diagonal}} \]

From [20, 21] and [3, 6], we know that:
1. An ONB of $V^2_j$ is

$$
\{ \Phi_{j,k_1,k_2} \mid \Phi_{j,k_1,k_2}(x, y) = 2^{-j} \Phi(2^{-j}x - k_1, 2^{-j}y - k_2)
= 2^{-j} \varphi(2^{-j}x - k_1) \varphi(2^{-j}y - k_2), (k_1, k_2) \in \mathbb{Z}^2 \}
$$

2. An ONB of $W^2_j$ is $\{ \psi^e_{j,k,m}(x, y) \mid e = 1, 2, 3; (k, m) \in \mathbb{Z}^2 \}$ in equation (2.46).

3. An ONB of $L^2(\mathbb{R}^2)$ is $\{ \psi^e_{j,k,m}(x, y) \mid e = 1, 2, 3; (j, k, m) \in \mathbb{Z}^3 \}$.

2-D Mallat Algorithm  Using the similar idea to 1-D case, we can get 2-D Mallat Algorithm:

$$
\begin{cases}
    c_{j,k_1,k_2} = \sum_{m_1,m_2} h_{m_1-2k_1} h_{m_2-2k_2} c_{j-1,m_1,m_2} \\
    d_{j,k_1,k_2}^1 = \sum_{m_1,m_2} h_{m_1-2k_1} g_{m_2-2k_2} c_{j-1,m_1,m_2} \\
    d_{j,k_1,k_2}^2 = \sum_{m_1,m_2} g_{m_1-2k_1} h_{m_2-2k_2} c_{j-1,m_1,m_2} \\
    d_{j,k_1,k_2}^3 = \sum_{m_1,m_2} g_{m_1-2k_1} g_{m_2-2k_2} c_{j-1,m_1,m_2}
\end{cases}
$$

(2.47)

2.4.2 2-D Image Wavelet Transform

Since 2-D MRA and Mallat Algorithm come from the tensor product of 1-D cases, we can consider 2-D Image Wavelet Transform as two directions of 1-D Wavelet Transforms: first is along the row direction, then along the column direction. See figure 2.14.

Mathematics Behind This Process
1. An image \( f \) can be represented by a \( M \times N \) matrix; More specifically, it can be written as a column vector whose entries are the row vectors:

\[
\begin{bmatrix}
  f_1 \\
  f_2 \\
  \vdots \\
  f_M
\end{bmatrix}
\]

where \( f_i = (f_{i1}, f_{i2}, \ldots, f_{iN}), \ i = 1 \cdots M \).

2. Then we apply the matrix \( S_i^*, \ i = 0, 1 \) (mentioned in section 2.3.3) to each of the row vector \( f_i \), we get equation 2.48. Note: The top half entries of the resulting column vector on the left of the equation (2.48) are the average information.
the low half entries are the difference.

\[
\begin{bmatrix}
a_{i,1} \\
a_{i,2} \\
\vdots \\
a_{i,N/2} \\
d_{i,1} \\
d_{i,2} \\
\vdots \\
d_{i,N/2}
\end{bmatrix}
= 
\begin{bmatrix}
\tilde{h}_0 & \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 & 0 & \cdots & 0 \\
0 & 0 & \tilde{h}_0 & \tilde{h}_1 & \tilde{h}_2 & \tilde{h}_3 & 0 & \cdots & 0 \\
\vdots \\
\tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 & 0 & \cdots & 0 \\
0 & 0 & \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 & 0 & \cdots & 0 \\
\vdots \\
\vdots
\end{bmatrix}
\begin{bmatrix}
f_{i,1} \\
f_{i,2} \\
f_{i,N}
\end{bmatrix}
\]

(2.48)

3. We collect all the resulting vectors from step 2, and then we have a new matrix \( T \) formed. The left half of matrix \( T \) is the average information, the right half is the difference information along the row direction.

4. We will do step 2 (apply the matrix \( S_i^*, i = 0, 1 \)) to each column vector of the new matrix \( T \).

5. The final resulting matrix is in the form below [30]: Each block has size \( \frac{M}{2} \times \frac{N}{2} \).

\[
f \mapsto \begin{pmatrix}
a^1 & h^1 \\
\vdots & \vdots \\
v^1 & d^1
\end{pmatrix}
\]

where \( a^1 = V_m \otimes V_n \), \( h^1 = V_m \otimes W_n \), \( v^1 = W_m \otimes V_n \), \( d^1 = W_m \otimes W_n \). \( a^1 \) tells us the average information of the original image; \( h^1 \) tells the detail information along the horizontal direction; \( v^1 \) shows the detail information along the vertical direction; \( d^1 \) shows the detail information along the diagonal.
Figure 2.15 gives us the sub-band structure after we apply 3-level DWT to a given image.

Figure 2.15: The Sub-band Structure After Applying 3-Level DWT

From graph 2.15, we see that we get a pyramid structure after decomposition. In this structure, each level has 3 sub-bands.

- $LL_3$ is the sub-band in the lowest frequency extracted by the 2-D scaling function $\varphi(x, y)$;

- $LH_3$, $LH_2$ and $LH_1$ are the sub-bands in the higher frequency along the horizontal direction extracted by the 2-D wavelet function;

- $HL_3$, $HL_2$ and $HL_1$ are the sub-bands in the higher frequency along the vertical direction extracted by 2-D wavelet function;

- $HH_3$, $HH_2$ and $HH_1$ are the sub-bands in the higher frequency along the diagonal direction extracted by 2-D wavelet function.
We use a test image 3-level decomposition and reconstruction with the choice of Daubechies wavelet (db4). See figure 2.16.
CHAPTER 3
QUANTIZATION ERRORS IN DIFFERENT A/D CONVERSIONS

3.1 Introduction

In chapter 2, we discussed the nature of *Wavelet Transform*. The question arises “How do we deal with those wavelet coefficients obtained as the result of *Wavelet Transform*?” This chapter will answer this question.

In chapter 1, we showed the process of DSP. We see that after we have obtained the wavelet coefficients from the *Wavelet Transform* we come to the next step, which is A/D conversion.

3.2 A/D Conversion

A/D conversion is also called Digitization. It is a process of converting analog signals to digital signals, which produces a stream of binary numbers. It has two steps:

- Sampling
- Quantization

3.2.1 Sampling

Sampling is the process of converting analog signals to discrete-time signals. Here is a figure 3.1 showing this process.

In figure 3.1, the dashed line represents the analog signal, the sequence of black dots represents the sampled signal, which is a discrete signal.
Note that we need to reconstruct our analog signal back for our human hearing system understanding after we process, store, retrieve, and transmit the signal. This means we will reconstruct the analog signal based on some sampled values from the signal. Does this scheme work? Is it reliable? Is there any theorem to guarantee this will work? The following theorem will answer all these questions. See [8].

**Bandlimited Functions and Shannon’s Theorem**

**Definition 3.1.** A function \( f \in L^2(R) \) is called *bandlimited* if its Fourier transform has compact support, i.e. \( \hat{f}(\xi) \equiv 0 \) for \( \xi > \Omega \).

\[
\hat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(t) e^{-i\xi t} \, dt
\]

Then \( \hat{f} \) can be represented by its Fourier series

\[
\hat{f}(\xi) = \sum_{n \in \mathbb{Z}} c_n e^{-i n \pi \xi / \Omega} \quad \text{for} \quad |\xi| \leq \Omega
\]

where

\[
c_n = \frac{1}{2\Omega} \int_{-\Omega}^{+\Omega} \hat{f}(\xi) e^{i n \pi \xi / \Omega} = \frac{1}{\Omega} \sqrt{\frac{\pi}{2}} f\left(\frac{n\pi}{\Omega}\right)
\]
then we get
\[ \hat{f}(\xi) = \frac{1}{\Omega} \sqrt{\frac{\pi}{2}} \sum_{n \in \mathbb{Z}} f\left( \frac{n \pi}{\Omega} \right) e^{-i n \xi \pi / \Omega} \chi_{|\xi| \leq \Omega} \]  
\text{(3.1)}

by the inverse Fourier transform, we get
\[ f(t) = \sum_{n=-\infty}^{\infty} f\left( \frac{n \pi}{\Omega} \right) \frac{\sin(\Omega t - n \pi)}{(\Omega t - n \pi)} \]
\[ = \sum_{n=-\infty}^{\infty} f\left( \frac{n \pi}{\Omega} \right) \text{sinc}\left( t - \frac{n \pi}{\Omega} \right) \]  
\text{(3.2)}

Equation (3.2) reflects the famous Shannon’s Theorem. See [6, 7].

**Theorem 3.1.** *(Shannon’s Theorem)* Any band-limited function \( f(t) \in L^2(\mathbb{R}) \) with support interval \( \hat{f} \subset [-\Omega, \Omega] \), (\( \Omega \) is arbitrary ) can be recovered perfectly from its sample values \( f\left( \frac{n \pi}{\Omega} \right) \) on a sufficiently dense grid via:
\[ f(t) = \sum_{n=-\infty}^{\infty} f\left( \frac{n \pi}{\Omega} \right) \text{sinc}\left( t - \frac{n \pi}{\Omega} \right) \]  
\text{(3.3)}

the sampling density is \( \frac{|\text{support } f|}{2\pi} = \frac{\Omega}{\pi} \). It is also called Nyquist frequency. \( \frac{\pi}{\Omega} \) is called Nyquist rate.

In practice, equation (3.3) is not widely used since \( \text{sinc}(t) = \frac{\sin t}{t} \) decays too slowly. It is not well localized, see figure 3.2.

**Oversampling** In order to circumvent this situation, we think of “oversampling”, which makes it possible to write \( f \) as a superposition of functions with faster decay.

We choose \( \lambda > 1 \) and sample the signal at more closely spaced points \( \frac{n \pi}{\lambda \Omega}, n \in \mathbb{Z} \), then equation (3.1) can be replaced by:
\[ \hat{f}(\xi) = \frac{1}{\lambda \Omega} \sqrt{\frac{2}{\pi}} \sum_{n \in \mathbb{Z}} f\left( \frac{n \pi}{\lambda \Omega} \right) e^{-i n \xi \pi / \lambda \Omega} \quad \text{for } |\xi| \leq \lambda \Omega \]
Figure 3.2: The Normalized Sinc Function: It Shows the Central Peak at \( x = 0 \), and Zero-Crossings at the Other Integer Values of \( x \).

Define function \( g \in C^\infty \) by

\[
\hat{g}(\xi) = \begin{cases} 
1 & \text{for } |\xi| \leq \Omega \\
0 & \text{for } |\xi| > \lambda \Omega
\end{cases}
\] (3.4)

Then \( \hat{f}(\xi) \) can be written

\[
\hat{f}(\xi) = \frac{\pi}{\lambda \Omega} \sum_{n \in \mathbb{Z}} f\left(\frac{n\pi}{\lambda \Omega}\right) e^{-i n \xi / \lambda \Omega} \hat{g}\left(\frac{\pi \xi}{\Omega}\right)
\]

Hence

\[
f(t) = \frac{1}{\lambda} \sum_{n = -\infty}^{\infty} f\left(\frac{n\pi}{\lambda \Omega}\right) g\left(\frac{\Omega}{\pi} t - \frac{n}{\lambda}\right)
\] (3.5)

By \( g \) function’s definition, \( g \) is smooth with fast decay; the above series converges absolutely and uniformly. See [8], [13].

3.2.2 Quantization

*Quantization* is the process of converting the infinitely many samples \( \{f\left(\frac{n\pi}{\lambda \Omega}\right)\}_{n \in \mathbb{Z}} \) we got from the first step into a finite collection of finite strings of digits.

Different Quantization Algorithms:
• **Pulse Code Modulation**

Replace each real-valued sample by its truncated *binary expansion*.

• **Sigma-Delta Modulation**

Every sample is replaced by one bit $q_n \in \{-1, 1\}$. It can be obtained by performing an iteration.

• **Beta Encoder**

Replace samples by their truncated beta expansion.

3.2.2.1 Pulse Code Modulation(PCM)

*Pulse Code Modulation (PCM)* is the simplest method to convert analog to digital. The main idea is to replace each sample value by its truncated binary expansion. See [8].

The *binary expansion* of a real number $r \in \{-1, 1\}$ has the form:

$$r = b_0 \sum_{i=1}^{\infty} b_i 2^{-i}$$

(3.6)

where $b_0 \in \{-1, 1\}$ is the sign bit, $b_i \in \{0, 1\}$ are the binary digits of $|r|$.

The sign bit $b_0 = Q_0(r)$ where

$$Q_0(z) = \begin{cases} 
1 & \text{if } z \geq 0; \\
-1 & \text{if } z < 0.
\end{cases}$$

(3.7)

To determine $b_i$, we need to define the second quantizer function $Q_1(z)$ as

$$Q_1(z) = \begin{cases} 
0 & \text{if } z < 1; \\
1 & \text{if } z \geq 1.
\end{cases}$$

(3.8)
Then $b_i$ can be computed using the following algorithm known as *Successive Approximation*:

Given $u_1 = 2|r|, b_1 = Q_1(u_1),$

\[
\begin{align*}
    u_{i+1} &= 2(u_i - b_i); \\
    b_{i+1} &= Q_1(u_{i+1}).
\end{align*}
\] (3.9)

**Advantage of PCM**  PCM has exponential accuracy.

**Theorem 3.2.** If $f(t)$ is original signal, $\bar{f}(t)$ is the recovered signal by using PCM, then we conclude that

\[
\|f(t) - \bar{f}(t)\|_{L^\infty} \leq K2^{-m}
\]

where $m$ is the number of the digits we keep in the truncated version of binary expansion, $K$ is a constant depending on $\lambda$. [8].

*Proof.* In practice, we observe a function only on a finite portion $I = [a, b]$ of the real line $R$. We also consider functions of limited maximum amplitude. i.e. we consider the class $S(\Omega, M, I)$ of all the signals $f(t) \in S(\Omega)$ that take values in $(-M, M)$ when $t \in I$: $|f(t)| < M, \quad t \in I$

It will be sufficient in all of what follows to consider the case where $\Omega = \pi$ and $M = 1$. We denote $S(\pi, 1, I)$ simply by $S$. Also let $x_n$ denote the sample values $x_n := f(\frac{n\pi}{M}) = f(\frac{n}{\lambda}), n \in \mathbb{Z}$, then the equation (3.5) becomes.

\[
f(t) = \frac{1}{\lambda} \sum_{n \in \mathbb{Z}} f(\frac{n}{\lambda}) g(t - \frac{n}{\lambda})
\] (3.10)
We consider those \( g \) such that \( \hat{g} \) is sufficiently smooth and satisfies

\[
\sum_{n \in \mathbb{Z}} G_\lambda(n) < \infty \tag{3.11}
\]

where

\[
G_\lambda(n) := \sup_{t \in [0, \frac{1}{\lambda}]} |g(t - \frac{n}{\lambda})| \tag{3.12}
\]

Given an integer \( m > 0 \) representing the number of digits kept in the truncated version of binary expansion. In addition, given an interval \( I = [a, b] \) in which we would like to reconstruct the signal, define \( \bar{I} := [a - M, b + M] \) where \( M \) depends on \( m \) and \( \lambda \) and is chosen so that

\[
\sum_{n \notin \lambda \bar{I}} G_\lambda(n) < 2^{-m} \tag{3.13}
\]

For any signal \( f(t) \in S \), PCM algorithm tells us that the truncated version sample values are \( \bar{x}_n := b_0(x_n) \sum_{i=1}^{m} b_i(x_n)2^{-i} \).

Thus we get:

\[
|x_n - \bar{x}_n| = |b_0(x_n) \sum_{i \in \mathbb{Z}} b_i(x_n)2^{-i} - b_0(x_n) \sum_{i=1}^{m} b_i(x_n)2^{-i}|
\]

\[
= |\sum_{i=m+1}^{\infty} b_i(x_n)2^{-i}|
\]

\[
\leq \sum_{i=m+1}^{\infty} 2^{-i} = 2^{-m} \tag{3.14}
\]

By the reconstruction formula (3.10), we have

\[
\bar{f}(t) = \frac{1}{\lambda} \sum_{n \in \lambda \bar{I}} \bar{x}_n g(t - \frac{n}{\lambda}) \tag{3.15}
\]
we can decompose the error term $f(t) - \bar{f}(t)$ into two components as:

$$f(t) - \bar{f}(t) = \frac{1}{\lambda} \sum_{n \in \mathbb{Z}} x_n g(t - \frac{n}{\lambda}) - \frac{1}{\lambda} \sum_{n \in \lambda I} x_n g(t - \frac{n}{\lambda})$$

$$= e_1(t) + e_2(t)$$

where

$$e_1(t) = \frac{1}{\lambda} \sum_{n \in \lambda I} (x_n - \bar{x}_n) g(t - \frac{n}{\lambda})$$

$$e_2(t) = \frac{1}{\lambda} \sum_{n \not\in \lambda I} x_n g(t - \frac{n}{\lambda})$$

By sample error bound (3.14), the sum of the sequence $G_\lambda(n)$ is finite by equation (3.11), and $\lambda > 1$, we have

$$|e_1(t)| \leq \sum_{n \in \lambda I} |x_n - \bar{x}_n| |g(t - \frac{n}{\lambda})|$$

$$\leq 2^{-m} \sum_{n \in \lambda I} |g(t - \frac{n}{\lambda})|$$

$$\leq C 2^{-m}$$

where $C$ is the sum of the series of $G_\lambda(n)$ given by equation (3.11).

Similarly, for $e_2(t)$, we notice that signal $f(t) \in S$, where $S$ represents $S(\pi, 1, I)$, which means that all the sample values $x_n$ satisfy $|x_n| < 1$. Then we conclude that

$$|e_2(t)| \leq \sum_{n \not\in \lambda I} |g(t - \frac{n}{\lambda})|$$

$$\leq \sum_{n \not\in \lambda I} G_\lambda(n)$$

$$\leq 2^{-m}$$
Putting these two error terms together, we conclude that: for all \( t \in I \),
\[
|f(t) - \tilde{f}(t)| \leq |e_1(t)| + |e_2(t)|
\]
\[
\leq (1 + C)2^{-m}
\]
\[
= K2^{-m}
\]

Where \( K = 1 + C \), which depends on \( \lambda \). See [8].

**Disadvantage of PCM**  The PCM algorithm lacks robustness!

In practice, we might have imperfect quantizer functions. Let’s say the circuit doesn’t follow the two quantizer functions \( Q_0(z) \) and \( Q_1(z) \) up there. Instead, it follows the other two called \( \tilde{Q}_0(z) \) and \( \tilde{Q}_1(z) \) which are defined as
\[
\tilde{Q}_0(z) = \begin{cases} 
-1, & z \leq \rho \\
1, & z > \rho
\end{cases}
\]
\[ (3.20) \]
\[
\tilde{Q}_1(z) = \begin{cases} 
0, & z \leq 1 + \rho \\
1, & z > 1 + \rho
\end{cases}
\]
\[ (3.21) \]

where \( \rho \) is not known precisely, except that it lies within a certain known tolerance, say \( |\rho| \leq \delta \) where \( \delta > 0 \) is fixed.

Now let’s consider an example. If a given number \( r \in (0, \rho] \), assume \( \rho > 0 \). Let’s see how the sign bit \( b_0 \) changes.

- use quantizer \( Q_0(z) \) by equation (3.7), we get \( b_0 = Q_0(r) = 1 \).
- use the imperfect quantizer \( \tilde{Q}_0(z) \) by equation (3.20), we get the sign bit \( b_0 = \tilde{Q}_0(r) = -1 \).
We found out that the sign bit of \( r \) will not be right! This will make the distortion \(|r - \tilde{r}|\) is at least as large as \(|r|\).

Of course, the above example is not the only trouble the imperfect quantizer functions will cause. For some numbers, we will get other bits \( b_i \) wrong if we use imperfect quantizer functions (3.20) and (3.21). Now we see that if our quantizer changes a little bit, the bit strings will change dramatically. Therefore, this algorithm lacks robustness.

Overall, the accuracy of PCM is of order \( O(2^{-m}) \) given sufficiently large interval \( I \) and sufficiently large number of bit \( m \). But it lacks robustness. When quantizer is not ideal as what we discussed in above example, the accuracy of PCM has no asymptotic decay as \( m \) is increased.

### 3.2.2.2 Sigma-Delta Modulation

In previous section, we have discussed that the PCM algorithm which has exponential accuracy lacks robustness. In this section, we would like to look at the other algorithm called Sigma-Delta Modulation (\( \sum \Delta \) Modulation). Compared with PCM, \( \sum \Delta \) Modulation has less accuracy, but it’s more robust. Here we only discuss the first order \( \sum \Delta \) Modulation. [7].

Let’s still use some of the notations in PCM. \( x_n \) denotes the sample values \( x_n := f(\frac{n}{\lambda}), n \in Z \). For every \( x_n \), \( \sum \Delta \) Modulation produces a bit stream \( q_n \), \( q_n \in \{-1, 1\} \). We also have the auxiliary sequence \( \{u_n\}_{n \in Z} \) as in PCM. \( u_n = 0 \) when \( n < \lambda a \) where \( I = [a, b] \).
Every sample $x_n$ is replaced by one bit $q_n \in \{-1, 1\}$. We can compute $q_n$ by the following:

$$
\begin{align*}
    u_n &= u_{n-1} + x_n + q_n; \\
    q_n &= Q_0(u_{n-1} + x_n).
\end{align*}
$$

(3.22)

Initial condition $u_0$ is arbitrarily chosen in $(-1, 1)$. $Q_0$ is the quantizer defined by equation (3.7).

In this algorithm, we found out that we have one bit assigned to one sample, which corresponds to $\lambda$ bits per Nyquist interval, thus resulting in $|I|\lambda$ bits for the whole signal $f(t)$.

**Disadvantage of $\sum \triangle$ Modulation** The accuracy is worse than that of PCM. It lacks exponential accuracy. Let’s show it by the following.

**Lemma 3.3.** For any signal $f(t) \in S$ and $|u_0| < 1$, the sequence $\{u_n\}_{n \in \mathbb{Z}}$ defined by the iteration (3.22) is uniformly bounded, $|u_n| < 1$ for all $n \in \mathbb{Z}$.

**Proof.** We can use induction to proof it. You may find the proof in [8].

**Lemma 3.4.** Given the sequence $\{u_n\}_{n \in \mathbb{Z}}$ defined in (3.22), for any sequence of number $\{g_n\}_{n \in \mathbb{Z}}$ satisfying $g_n \to 0$, as $n \to \infty$, we have

$$
\sum_{n \in \mathbb{Z}} (u_n - u_{n-1})g_n = \sum_{n \in \mathbb{Z}} u_n(g_n - g_{n+1})
$$

(3.23)

**Proof.** Let’s start from the right hand side. By the summation by part, we know that

$$
\begin{align*}
    u_n &= \sum_{k \leq n} u_k - u_{k-1} \\
    g_k &= \sum_{n \geq k} (g_n - g_{n+1})
\end{align*}
$$
\[ RHS = \sum_{n \in \mathbb{Z}} \sum_{k \leq n} (u_k - u_{k-1})(g_n - g_{n+1}) \]

\[ = \sum_{k \in \mathbb{Z}} (u_k - u_{k-1}) \sum_{n \geq k} (g_n - g_{n+1}) \]

\[ = \sum_{k \in \mathbb{Z}} g_k (u_k - u_{k-1}) = LHS \]

\[ \square \]

**Theorem 3.5.** [7, 2] For signal \( f(t) \in S, \lambda > 1 \), define the sequence \( b_n \) by the iteration (3.22), with \( u_0 \) chosen arbitrarily in \((-1, 1)\). \( g \) is the function defined in (3.4). Then

\[ |f(t) - \frac{1}{\lambda} \sum_{n \in \mathbb{Z}} b_n g(t - \frac{n}{\lambda})| \leq \|g'\|_{L^1} \lambda^{-1} \] (3.24)

**Proof.** By equation 3.10 on page 46, iteration 3.22, lemmas 3.23 and 3.4, we have:

\[ |f(t) - \frac{1}{\lambda} \sum_{n \in \mathbb{Z}} b_n g(t - \frac{n}{\lambda})| = \frac{1}{\lambda} \left| \sum_{n \in \mathbb{Z}} (x_n - b_n) g(t - \frac{n}{\lambda}) \right| \]

\[ = \frac{1}{\lambda} \left| \sum_{n} x_n \left[ g(t - \frac{n}{\lambda}) - g(t - \frac{n+1}{\lambda}) \right] \right| \]

\[ \leq \frac{1}{\lambda} \sum_{n} |g(t - \frac{n}{\lambda}) - g(t - \frac{n+1}{\lambda})| \]

\[ \leq \frac{1}{\lambda} \sum_{n} \int_{t - \frac{n}{\lambda + 1}}^{t - \frac{n+1}{\lambda + 1}} |g'(y)| \, dy \]

\[ = \|g'\|_{L^1} \lambda^{-1} \]

We see that \( \sum \Delta \) Modulation results in \( O(\lambda^{-1}) \) accuracy, which is much worse than PCM. \[ \square \]
Without loss of generality, we can extend this theorem.

Let

\[ D(u, g) = |f(t) - \frac{1}{\lambda} \sum_{n \in \mathbb{Z}} f\left(\frac{n}{\lambda}\right) g(t - \frac{n}{\lambda})| \]

\[ = \frac{1}{\lambda} |\sum_{n \in \mathbb{Z}} u_n [g(t - \frac{n}{\lambda}) - g(t - \frac{n + 1}{\lambda})]| \]

\[ = \frac{1}{\lambda} |\sum_{n \in \mathbb{Z}} u_n (g_n - g_{n+1})| \]

By the Cauchy–Schwartz inequality, we have

\[ D(u, g) \leq \frac{1}{\lambda} \left( \sum_{n \in \mathbb{Z}} u_n^2 \right)^{1/2} \left( \sum_{n \in \mathbb{Z}} |g_n - g_{n+1}|^2 \right)^{1/2} \]

\[ \leq \frac{1}{\lambda} \left( \sum_{n \in \mathbb{Z}} u_n^2 \right)^{1/2} \cdot (\int |g'|^2 dt)^{1/2} \]

Hence

\[ D(u, g) \leq \frac{1}{\lambda} \|u\|_{l^2} \cdot \|g'\|_{L^2} \]  \hspace{1cm} (3.25)

where \( \|g'\|_{L^2} \) is called energy of \( g \).

**Theorem 3.6.** If sequence \( u_n \in l^2 \), \( g \) is same as before, then there exists function \( H \), depending on \( u_n \), satisfying \( H' \in L^2 \) such that

\[ D(u, g) = \int H'(t) g'(t) dt \]

**Proof.** By Riesz Representation Theorem [24], we will get it. \( \square \)

**Remark 3.7.** We offer a comparison between the estimate (3.24) in Theorem 3.5 on one side, and the formula about \( D(u, g) \) in (3.25) on the other.
• (3.24) is an estimate, while $D(u, g)$ offers an exact formula. While (3.24) involves an $L^1$ norm of the first derivative of the function $g$, formula $D(u, g)$ results from an estimate involving the $L^2$ norm instead. As a result, we are in the realm of Hilbert Space, and 'Riesz' lemma applies [24]. Consequently, the exact formula for $D(u, g)$ results.

• While estimate (3.24) requires quantizers $Q$ such that the resulting sequence \( \{u_n\}_{n \in \mathbb{Z}} \) is in $L^\infty$, our formula $D(u, g)$ requires instead that the sequence \( \{u_n\}_{n \in \mathbb{Z}} \) belongs to $l^2$ (a smaller class of quantizers), and hence it applies then only to the corresponding smaller family of quantizers $Q$.

• Nonetheless, there are applications when this more restrictive class is appropriate and we show that in these instances the stronger conclusions are then valid.

**Advantage of $\sum \triangle$ Modulation** For $\sum \triangle$ Modulation, it has *Self Error Correction* property, which means it’s *stable and impervious* to the errors which occurred in the imperfect quantizers.

Suppose that we use imperfect quantizer $\bar{Q}_0(z)$ defined in equation 3.20 on page 49 instead of using $Q_0(z)$.

\[
\bar{Q}_0(z) = \begin{cases} 
-1, & z \leqslant \rho \\
1, & z > \rho 
\end{cases}
\]

Where $\rho$ is not known precisely, except that it lies within a certain known tolerance, say $|\rho| \leqslant \delta$ where $\delta > 0$ is fixed.
This results in the new auxiliary sequence \( \{\tilde{u}_n\}_{n \in \mathbb{Z}} \) which satisfies \( \tilde{u}_n = 0 \) for \( \frac{n}{\lambda} < a \). And the recursion becomes

\[
\begin{align*}
\tilde{u}_n &= \tilde{u}_{n-1} + x_n + \tilde{b}_n; \\
\tilde{b}_n &= \tilde{Q}_0(\tilde{u}_{n-1} + x_n).
\end{align*}
\] (3.26)

The following theorem shows that given any \( \delta > 0 \), an error of at most \( \delta \) in quantizer will not affect too much the distortion bound in (3.24). The accuracy order is still \( O(\lambda^{-1}) \). The difference is only a constant.

**Lemma 3.8.** For any signal \( f(t) \in S \) and a given number \( \delta > 0 \) satisfying \( |\rho| \leq \delta \), the sequence \( \{\tilde{u}_n\}_{n \in \mathbb{Z}} \) defined by the iteration (3.26) is uniformly bounded, for all \( n \in \mathbb{Z} \),

\[
|\tilde{u}_n| < 2 + \delta
\] (3.27)

**Proof.** Use induction to show it. We can also find the proof in [8].

**Theorem 3.9.** Suppose we use imperfect quantizer \( \tilde{Q}_0(z) \) defined by (3.20), with \( |\rho| \leq \delta \), in place of \( Q(z) \). The bit stream \( \tilde{b}_n \) is obtained by iteration (3.26), then the reconstructed signal \( \tilde{f}(t) \) satisfies

\[
|f(t) - \tilde{f}(t)| \leq (2 + \delta)\|g^\prime\|_{L^1} \cdot \lambda^{-1}
\] (3.28)

**Proof.** The idea of proving this theorem is to use summation by part, iteration 3.26, the lemmas 3.8 and 3.4, which is same as before. We will omit it here.

3.2.2.3 Beta Expansion (Beta Encoder)

Naturally we will ask a question now.
Is there an algorithm such that not only has exponential accuracy like PCM but also is robust and impervious to the errors of imperfect quantizers like $\sum \Delta$ Modulation?

The answer is YES! That’s Beta Expansion! Instead of using binary expansion, we utilize representation with respect to a base $\beta \in (1, 2)$. [8]

Given a non-integer $1 < \beta < 2$, $\gamma := \frac{1}{\beta}$. one can express any number $r \in [0, 1]$ as

$$ r = \sum_{i=1}^{+\infty} b_i \beta^{-i} = \sum_{i=1}^{\infty} b_i \gamma^i \quad (3.29) $$

with $b_i \in \{0, 1\}$.

We can compute $b_i$ by

$$ \begin{cases} u_{i+1} = \beta(u_i - b_i) \\ b_{i+1} = Q_1(u_{i+1}) \end{cases} \quad (3.30) $$

Given $u_1 = \beta r$, $b_1 = Q_1(u_1)$.

**Advantages of Beta Encoder**

1. Self-Error Correction

2. Exponential Accuracy

Suppose that we use imperfect quantizer $\tilde{Q}_1(z)$ defined by (3.21), where $\rho$ is not known precisely and may vary. We know it lies within a certain known tolerance, say $|\rho| \leq \delta$ where $\delta > 0$ is fixed. Instead of getting bit stream $b_i$, we have inaccurate bit
stream $\tilde{b}_i$ obtained by

$$\begin{cases}
\tilde{u}_{i+1} = \beta (\tilde{u}_i - \tilde{b}_i) \\
\tilde{b}_{i+1} = \tilde{Q}_1(\tilde{u}_{i+1})
\end{cases} \quad (3.31)$$

Given $\tilde{u}_1 = \beta r, \tilde{b}_1 = \tilde{Q}_1(\tilde{u}_1)$.

Here we still use same notation as before. $x_n := f(\frac{n}{\lambda})$, which is the sample value; $\bar{x}_n := \sum_{i=1}^m b_i \gamma^i$.

**Lemma 3.10.** Let $\delta > 0$ and $x_n \in [0,1]$. Suppose we use quantizer $\tilde{Q}_1$ instead of $Q_1$ with the value $\rho$ possibly varying but always satisfying $|\rho| \leq \delta$. If $\beta$ satisfies

$$1 \leq \beta \leq \frac{2 + \delta}{1 + \delta} \quad (3.32)$$

then for each $m \geq 1$, $\bar{x}_n := \sum_{i=1}^m b_i \gamma^i$ satisfies

$$|x_n - \bar{x}_n| \leq A \gamma^m \quad (3.33)$$

where $A = 1 + \delta$.

**Proof.** Claim that sequence $\tilde{u}_n$ satisfies

$$0 \leq \tilde{u}_n \leq \beta(1 + \delta) \quad (3.34)$$

We can prove it using induction. Assume that (3.34) is true for $n = N$, which means $\tilde{u}_N \leq \beta(1 + \delta)$. We need show that (3.34) is true for $n = N + 1$.

(1) If $b_N = 0$, by assumption $\tilde{u}_N \leq \beta(1 + \delta)$ and iteration (3.31), we have

$$0 \leq \tilde{u}_{N+1} = \beta \tilde{u}_N \leq \beta(1 + \delta)$$
(2) If \( b_N = 1 \), by iteration, assumption and condition (3.32), we have

\[
0 \leq \tilde{u}_{N+1} = \beta(\tilde{u}_n - 1) \leq \beta[\beta(1 + \delta) - 1] \leq \beta(2 + \delta - 1) = \beta(1 + \delta)
\]

Thus we prove the claim.

Note: If we use quantizer \( Q_1 \), which means \( \delta = 0 \), by the bound (3.34), we know the accurate sequence \( u_n \) is uniformly bounded by \( \beta \), which means \( 0 \leq u_n \leq \beta \).

On the other hand, for a fixed \( n \),

\[
|x_n - \tilde{x}_n| = |x_n - \sum_{i=1}^{m} \tilde{b}_i \gamma_i|
\]

\[
= |\gamma \tilde{u}_1 - \sum_{i=1}^{m} \gamma \tilde{u}_i - \gamma \tilde{u}_{i+1}|\]

\[
= |\gamma \tilde{u}_1 - (\sum_{i=1}^{m} \gamma \tilde{u}_i - \sum_{i=1}^{m} \gamma \tilde{u}_{i+1})|\]

\[
= |\gamma \tilde{u}_1 - \gamma \tilde{u}_1 + \gamma^{m+1} u_{m+1}|\]

\[
= \gamma^{m+1} |\tilde{u}_{m+1}|\]

\[
\leq \gamma^{m+1} \beta(1 + \delta)\]

\[
= (1 + \delta) \gamma^m\]

\[
= A \gamma^m
\]

Note: As we mentioned above, if we use accurate quantizer \( Q_1 \), the sample error bound will be

\[
|x_n - \tilde{x}_n| \leq \gamma^m \quad (3.35)
\]

\( \square \)

**Theorem 3.11.** For any signal \( f(t) \in S \), given an interval \( I = [a, b] \) in which we would like to reconstruct the signal, define \( \bar{I} := [a - M, b + M] \) where \( M \) depends on
\( m \) and \( \lambda \) and is chosen so that

\[
\sum_{n \notin \Lambda I} G_\lambda(n) < \gamma^m \tag{3.36}
\]

where \( G_\lambda(n) \) is defined by 3.12 on page 47. Then the beta encoder with \( m \) bits per sample satisfies

\[
|f(t) - \bar{f}(t)| \leq (C + 1)\gamma^m \quad t \in I \tag{3.37}
\]

with \( C = \sum_{n \in \mathbb{Z}} G_\lambda(n) \) only depending on the choice of function \( g \) defined 3.4 on page 44.

Moreover, if we use quantizer \( \tilde{Q}_1 \) given by 3.21 on page 49, with \( \rho \) satisfying \( \rho \leq \delta \), then we still obtain the error bound (3.37) with the different constant \( B \). i.e.

\[
|f(t) - \tilde{f}(t)| \leq B\gamma^m \tag{3.38}
\]

\( \tilde{Q}_1(z) \) where \( B = [(1 + \delta)C + 1] \) with \( C = \sum_{n \in \mathbb{Z}} G_\lambda(n) \).

**Proof.** The idea is the same as in theorem 3.2 on page 46. If we define \( e_1(t) \) as in (3.16), then by (3.35) in lemma (3.10), we get

\[
|e_1(t)| \leq C\gamma^m,
\]

where \( C = \sum_{n \in \mathbb{Z}} G_\lambda(n) \).

Also define \( e_2(t) \) as in (3.17), then by (3.36) and \( |x_n| < 1 \)

\[
|e_2(t)| \leq \gamma^m
\]

Putting together, we have

\[
|f(t) - \tilde{f}(t)| \leq (C + 1)\gamma^m
\]
where \( C = \sum_{n \in \mathbb{Z}} G_\lambda(n) \) is only depending on the choice of function \( g \) defined (3.4).

Similarly, we can prove: if we use imprecise quantizer \( \tilde{Q}_1 \), beta encoder algorithm has self-error correction, which means we still obtain the error bound (3.38) with a different constant \( B = [(1 + \delta)C + 1] \).

\[ \square \]

3.3 Summary

1. PCM: has exponential accuracy \( O(2^{-m}) \), but lacks robustness;

2. Sigma-Delta Modulation: has error correction property, but has less accuracy \( O(\lambda^{-1}) \);

3. Beta Encoder: not only has exponential accuracy \( O(\gamma^m) \), but also has error correction property.
CHAPTER 4
IMAGE COMPRESSION BASED ON WAVELET SPATIAL ORIENTATION TREE

4.1 Introduction

4.1.1 The Necessity and Feasibility of Image Compression

Considering our modern information society with its growing operational requirements of telecommunications, image communication and the capacity of communication networks have become increasingly prominent, especially digital image communication. The huge amount of data is becoming even harder to transport and store. Thus it becomes a big difficulty for us to obtain and use the image information and results in a “bottleneck” problem in the development of image/video communication. In order to solve this problem, simply expanding the storage capacity and increasing the transmission rate is not realistic. Image compression technology is an effective way. Therefore, image compression has become the hottest research spot in image communication. Figure 4.1 is a diagram to show the whole procedure of image processing.

Image compression is always hard due to the large volume of data used to represent an image. To minimize the large amount of data without significantly losing information is the goal of image compression. In other words, image compression aims at using a minimum number of bits representing the images, while ensuring the quality of recovery images. By doing image compression, we can not only save storage space, but also improve the transmission efficiency. Therefore, it is imperative.
On the other hand, image compression is feasible due to the statistical features of images and characteristics of the human visual system. For instance, we know an image can be represented as a matrix, and each entry represents the grey level of one pixel. There are great relevance between pixels along the row direction and column direction. Overall there are significant redundancies in data.

4.1.2 Classifications

There are many types of redundancy in data, like spatial redundancy, visual redundancy etc. Based on different redundancy, image/video compression methods generally fall into two categories:

- **Lossless Compression**
  
  It allows error-free data Reconstruction.

- **Lossy Compression**
  
  It doesn’t preserve the information completely.
For *Lossless Compression*, it includes several methods:

- Run Length Coding.
- Huffman Coding.
- Arithmetic Coding.

For *Lossy Compression*, it includes:

- Predictive Coding.
- Vector Quantization.
- Hierarchical and Progressive Compression.
- Transform Coding.
- Fractal Coding.

Among these methods, *Transform Coding* is more commonly used. It includes:

- Karhunen-Loeve Transform. [27]
- Walsh-Hadamard Transform.
- Discrete Fourier Transform (DFT).
- Discrete Cosine Transform (DCT).
- Discrete Wavelet Transform (DWT).
Among the methods above, DCT has good compression performance and a small amount of computations. Therefore DCT became the core technique in image processing in the last decade. However, DCT gradually exposed its shortcomings.

- It results in the *blocking artifacts* since we need to divide the images into many $8 \times 8$ small blocks before we apply DCT.

- One can’t stop the procedure in the middle of encoding or decoding based on their own need for the quality of image.

Due to these drawbacks, Wavelet Theory and Multi-Resolution Analysis (MRA) start to play a more and more important role in image compression. Wavelet Transform has multi-scale structure. We don’t need to divide images into blocks. This eliminates the *blocking artifacts* caused by block-based DCT. In addition, Wavelet coefficients have the following statistical characteristics, which have been proved to be essential to image compression:

- Time-frequency localization.

- Energy compression.

- Clustered structure of wavelet coefficients in sub-band.

- Similarity of wavelet coefficients in inter-sub-band.

- Decreasing progression of wavelet coefficients from the low-frequency sub-band to high-frequency sub-band.
4.2 Embedded Coding and Spatial Orientation Tree

Nowadays, most high-performance wavelet-based image compression algorithms are Embedded Coding algorithms. See [25].

The so-called embedded coding algorithm is that the encoder orders the bit-streams based on importance and then encodes them. Using an embedded coding, an encoder can always terminate the encoding procedure at any point allowing a target rate or distortion metric to be met exactly. When the target is satisfied, the encoding deceases.

Similarly, given a bit stream, the decoder can stop decoding at any point and can produce reconstructions corresponding to all lower-rate encodings. The larger the bit-stream, the smaller the image distortion.

A typical image coder has the following three steps as shown in figure 4.2.

- **A Transformation**
  Here we mainly use Discrete Wavelet Transform, which we have discussed in Chapter 2.

- **A Quantizer**
  We have talked about many different quantization methods in Chapter 3.

- **Data Compression**
  We will mainly talk about in this chapter.

The Embedded Zerotree Wavelet (EZW) algorithm belongs to Embedded Coding. Before we introduce EZW, let’s first look at the Spatial Orientation Tree.
Suppose we are given an image which can be represented by $M \times N$ matrix. Let’s do the following.

- After we apply two dimension Discrete Wavelet Transform (2D-DWT, we mentioned in Chapter 2) to this matrix once, called 1-level DWT, we will get 4 matrices with size $\frac{M}{2} \times \frac{N}{2}$. Based on the different sub-band, these 4 matrices are called 1-level wavelet coefficients in Low-Low Frequency ($LL_1$), Low-High frequency ($LH_1$), High-Low frequency ($HL_1$), High-High frequency ($HH_1$) respectively.

- If we apply 2-D DWT to wavelet coefficients ONLY in Low-Low frequency ($LL_1$) one more time, called 2-level DWT, we will get 4 more sub-matrices like before, called 2-level wavelet coefficients in $LL_2$, $LH_2$, $HL_2$, $HH_2$ respectively.

- If we keep doing this operation $L$ times, we will get $L$-level wavelet coefficients in $LL_L$, $LH_L$, $HL_L$, $HH_L$ respectively and $3L+1$ matrices (or images) at different scale in total. It’s just like MRA in chapter 2.

Figure 4.3 illustrates 2-level DWT decomposition of an image.
These wavelet coefficients can be seen as a tree structure. Each coefficient is a node. Figure 4.4 shows it. Here are some comments about this tree structure.

- The root is the node in the most low-frequency sub-band, which is at the top left. This root has three nodes which are located at the same spatial location in the three other remaining increasing-frequency sub-bands at the same level.
• All the other nodes are related to four nodes each at the next finer level of similar orientation.

• The coefficient at the coarse level is called the Parent. All coefficients corresponding to the same spatial location at the NEXT finer level of similar orientation are called children.

• For a given parent, the set of all coefficients at ALL finer levels of similar orientation corresponding to the same spatial location are called descendants.

• For a given child, the set of coefficients at ALL coarser levels of similar orientation corresponding to the same spatial location are called ancestors.

• Suppose a given parent is located at \((i, j)\) in the matrix, then its four children are located at \((2i, 2j), (2i + 1, 2j), (2i, 2j + 1), (2i + 1, 2j + 1)\) respectively. The lowest frequency sub-band is at the top left, the highest one is at the bottom right.

4.3 Embedded Zerotree Wavelet Algorithm

The Embedded Zerotree Wavelet algorithm (EZW) belongs to the Embedded Coding. It utilizes the features of wavelet coefficients, such as that wavelet coefficients get smaller as the frequency of their sub-band gets higher, as well as the wavelet coefficient’s self-similarity property that we mentioned in section 4.2. The EZW includes two steps:

2. Successive-Approximation Quantization (SAQ).

4.3.1 Compression of Significance Maps

We have seen that we will get a *Spatial Orientation Tree* if we apply several times DWT to an image. Based on the *Spatial Orientation Tree*, we generate a new data structure called *Zerotree*.

**Definition 4.1.** A wavelet coefficient is called *insignificant* with respect to a given threshold $T$ if $|c_{ij}| < T$, otherwise it is called *significant*.

*Zerotree* is based on the hypothesis that if a coefficient $c_{i,j}$ at a coarse level is *insignificant* with respect to a given threshold $T$ then all of its *descendants* (defined above) are also *insignificant*. All these wavelet coefficients form a *Zerotree*.

**Definition 4.2.** A coefficient $c_{i,j}$ is considered to be an *element* of a *zerotree* for a given threshold $T$ if itself and all its *descendants* are insignificant with respect to threshold $T$. An *element* of a zerotree is a *zerotree root* if it is not the descendant of a previously found zerotree root with respect to threshold $T$, i.e., it is not *predictably insignificant* from the discovery of a zerotree root at a coarser level with respect to the same threshold $T$.

By the definitions above, we see that there will be many zerotrees if the zerotree hypothesis is satisfied. Therefore, we can use a *zerotree root* to represent a whole *zerotree*, thereby reducing the size of the bit stream.

The significance map can be represented by the string of 4 symbols\(^1\):

\(^1\)We use the first three symbols to represent the wavelet coefficients in the finest level.
1. Positive Significant (POS)

A positive coefficient whose magnitude is not less than the given threshold $T$.

2. Negative Significant (NEG)

A negative coefficient whose magnitude is not less than the given threshold $T$.

3. Isolated Zero (IZ)

The coefficient is insignificant but has some significant descendants.

4. Zerotree Root (ZTR)

It is defined above.

The decision tree for encoding a coefficient is shown in Figure 4.5 on page 71.

Now we have two questions. With which coefficient do we need to start our work above? Having begun at this coefficient, in what direction does our work progress?

**Scanning Order** A scanning of coefficients is performed in a way that no child node is scanned before its parent. For the 3-level DWT, the scan starts from the lowest frequency sub-band denoted $LL_3$, which is at the top left, then scans the sub-bands $LH_3$, $HL_3$, $HH_3$, at which point it moves to the finer level 2, which are $LH_2$, $HL_2$, $HH_2$ etc. Overall, scanning follows a “Z” route starting from the top left to the bottom right. Figure 4.6 on page 72 shows the order of scanning.
4.3.2 Successive-Approximation Quantization (SAQ)

The key idea of EZW is to use Successive-Approximation Quantization (SAQ). SAQ means the use a sequence of threshold $T_0, T_1, T_2, \cdots, T_{N-1}$ to determine the significance map, where $T_i = \frac{T_{i-1}}{2}$. The initial threshold $T_0$ is chosen such that $|c_{ij}| < 2T_0$, where $c_{ij}$ represents the wavelet coefficients.

Let’s consider the encoding process and decoding process separately.
4.3.2.1 Encoding Process

There are two separate lists always maintained during the encoding process\(^2\):

1. **The Dominant List**
   
   It contains the *coordinates* of coefficients that have not been found to be significant in the scanning order mentioned in the previous section. The initial value is the coordinates of all the wavelet coefficients.

2. **The Subordinate List**
   
   It contains the *magnitudes* of those coefficients that have been found to be *significant*. Its initial value is empty set.

*The encoding process* is alternated between the following two passes:

1. **A Dominant Pass**

2. **A Subordinate Pass**

\(^2\)For each threshold, each list is scanned once.
**A Dominant Pass**  During a *Dominant Pass*, coefficients with coordinates on the dominant list are compared to the given threshold \( T_i \) to determine their significance. The main idea is to follow the decision tree in figure 4.5 and use one of the four symbols to represent them. If it is significant, consider its sign. Each time if a coefficient is encoded as significant, either positive or negative, its magnitude is appended to the subordinate list, and the coefficient in the wavelet transform array is set to zero so that the significant coefficient does not prevent the occurrence of a zerotree on future dominant passes at smaller thresholds.

**A Subordinate Pass** A *Subordinate Pass* starts after the *dominant pass*. The main idea is to refine those significant coefficients in the *subordinate list*. For a given threshold \( T_i \), a significant coefficient lies in an uncertainty interval \([T_i, 2T_i]\). Now we create two new uncertainty intervals by cutting \([T_i, 2T_i]\) in half. If a magnitude in the subordinate list falls in the lower half interval \([T_i, \frac{3T_i}{2}]\), then we use symbol “1” to encode it. Otherwise, if it falls in the upper half interval \([\frac{3T_i}{2}, 2T_i]\), we use symbol “0” to encode. After a completion of the subordinate pass, the magnitudes on the subordinate list are sorted in decreasing magnitude.

4.3.2.2 Decoding Process

In the *Decoding Process*, the reconstruction values are chosen to be the middle points of the uncertainty intervals.

**Example** Consider the 3-level DWT of an \( 8 \times 8 \) image. The wavelet coefficients are shown in table 4.1.
When $T_0 = 32$:

**Dominant Pass** After observing the wavelet coefficients, we see that the largest coefficient magnitude is 63. Then we can choose the initial threshold to be $T_0 = 32$. Table 4.2 shows the process of the first dominant pass. Here symbol “Z” represents a zero when there are no children.

**Subordinate Pass** Four coefficients are identified during the first dominant pass. Their magnitudes are $(63, 34, 49, 47)$. Prior to the first subordinate pass the uncertainty interval is $[32, 64)$. The first subordinate pass will refine these magnitudes and identify them as being either in interval $[32, 48)$, which will be encoded with the symbol “0” or interval $[48, 64)$, which will be encoded with the symbol “1”. Table 4.3 shows the process of the first subordinate pass.

<table>
<thead>
<tr>
<th>63</th>
<th>-34</th>
<th>49</th>
<th>10</th>
<th>7</th>
<th>13</th>
<th>-12</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>-31</td>
<td>23</td>
<td>14</td>
<td>-13</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>-1</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>3</td>
<td>-12</td>
<td>5</td>
<td>-7</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>-9</td>
<td>-7</td>
<td>-14</td>
<td>8</td>
<td>4</td>
<td>-2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>-5</td>
<td>9</td>
<td>-1</td>
<td>47</td>
<td>4</td>
<td>6</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-3</td>
<td>2</td>
<td>3</td>
<td>-2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>-3</td>
<td>6</td>
<td>-4</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>-4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.1: 3-Level DWT Coefficients
Third Step After we finish the first subordinate pass, we need to reorder these magnitudes in the subordinate list in the decreasing order. Note the correct order will be $(63, 49, 34, 47)$ instead of $(63, 49, 47, 34)$. 39 is moved ahead of 34 because the reconstruction values 56 and 40 are distinguishable from the encoder’s view. However, we can’t move 47 in front of 34 since both of them have the same reconstruction value 40 but 34 is in the lower frequency sub-band, which is more important than 47.

Last Step Set all the significant coefficients zero in the wavelet coefficients array for the purpose of determining if a zerotree exists.

When $T_1 = 16$:

We follow the same steps as above.

Dominant Pass: Table 4.4 shows the result of the process of the second dominant pass.

Subordinate Pass: The subordinate list now contains $(63, 49, 34, 47, 31, 23)$. Before we perform the subordinate pass, we have three uncertainty intervals: $[48, 64)$, $[32, 48)$ and $[16, 31)$. Now we perform the subordinate pass. We create two new uncertainty intervals for each of the three current uncertainty intervals, which are $[48, 64)$, $[56, 64)$, $[32, 40)$, $[40, 48)$, $[16, 24)$, $[24, 31)$.

Third Step: Using the middle points of each of these uncertainty intervals as the reconstruction values and reordering them, we get table 4.5 as the result of the second subordinate pass.
Last Step: Set all the significant coefficients zero in the wavelet coefficients array.

Keep halving the threshold and repeating the above loop until the quality of the image is satisfied. From table 4.3 and 4.5, we see that the reconstruction errors decrease as the threshold decreases.

4.3.3 Implementation of the EZW

We use MATLAB to implement it. Lena.jpg is used as the test image. Figure 4.7 shows six reconstructed images using EZW under different settings. From left to right, their settings are the following:

1. original test image;
2. 20-level Haar Wavelet Transform, 0.5 bit/pixel, PSNR =35.47;
3. 20-level Haar Wavelet Transform, 3 bit/pixel, PSNR =33.83;
4. 20-level db4 Transform, 2.5bit/pixel, PSNR =33.92;
5. 20-level db6 Transform, 2.5bit/pixel PSNR =33.78;
6. 20-level db8 Transform, 2.5bit/pixel, PSNR =33.51.

4.3.4 Summary

EZW algorithm is a simple but very effective image compression algorithm. It overturned a conclusion in information theory that algorithm efficiency is bound to increase the cost of computational complexity. EZW not only has a very high
efficiency but also low computational complexity due to the fact that the magnitude of wavelet coefficients will decrease if the frequency of the wavelet band increases.

Moreover, EZW has taken progressive transmission. We can stop the algorithm anytime in accordance with the requirement of image quality and bit rate. We can design the size of the final threshold to change the image compression rate and image quality.

However, EZW also has its shortcomings. For instance, compared to other algorithms EZW still has low efficiency as a result of repeated image scanning.
<table>
<thead>
<tr>
<th>Sub-band</th>
<th>Coefficient</th>
<th>Symbol</th>
<th>Reconstruction Value w/o Subordinate Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LL_3$</td>
<td>63</td>
<td>POS</td>
<td>48</td>
</tr>
<tr>
<td>$HL_3$</td>
<td>-34</td>
<td>NEG</td>
<td>-48</td>
</tr>
<tr>
<td>$LH_3$</td>
<td>-31</td>
<td>IZ</td>
<td>0</td>
</tr>
<tr>
<td>$HH_3$</td>
<td>23</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>49</td>
<td>POS</td>
<td>48</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>10</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>14</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>-13</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>15</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>14</td>
<td>IZ</td>
<td>0</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>-9</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>-7</td>
<td>ZTR</td>
<td>0</td>
</tr>
<tr>
<td>$HL_1$</td>
<td>7</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>$HL_1$</td>
<td>13</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>$HL_1$</td>
<td>3</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>$HL_1$</td>
<td>4</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>$LH_1$</td>
<td>-1</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>$LH_1$</td>
<td>47</td>
<td>POS</td>
<td>48</td>
</tr>
<tr>
<td>$LH_1$</td>
<td>-3</td>
<td>Z</td>
<td>0</td>
</tr>
<tr>
<td>$LH_1$</td>
<td>-2</td>
<td>Z</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.2: Process on the First Dominant Pass
<table>
<thead>
<tr>
<th>Coefficient Magnitudes</th>
<th>Symbol</th>
<th>Reconstruction Value with Subordinate Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>1</td>
<td>56</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>56</td>
</tr>
<tr>
<td>47</td>
<td>0</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 4.3: Process of the First Subordinate Pass
<table>
<thead>
<tr>
<th>Sub-band</th>
<th>Wavelet Coefficients</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LL_3$</td>
<td>0</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HL_3$</td>
<td>0</td>
<td>ZTR</td>
</tr>
<tr>
<td>$LH_3$</td>
<td>-31</td>
<td>NEG</td>
</tr>
<tr>
<td>$HH_3$</td>
<td>23</td>
<td>POS</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>0</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>10</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>14</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HL_2$</td>
<td>-13</td>
<td>ZTR</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>15</td>
<td>ZTR</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>14</td>
<td>ZTR</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>-9</td>
<td>ZTR</td>
</tr>
<tr>
<td>$LH_2$</td>
<td>-7</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HH_2$</td>
<td>3</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HH_2$</td>
<td>-12</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HH_2$</td>
<td>-14</td>
<td>ZTR</td>
</tr>
<tr>
<td>$HH_2$</td>
<td>8</td>
<td>ZTR</td>
</tr>
</tbody>
</table>

Table 4.4: The Second Dominant Pass
<table>
<thead>
<tr>
<th>Coefficient Magnitudes</th>
<th>Symbol</th>
<th>Reconstruction Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>47</td>
<td>1</td>
<td>44</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>28</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4.5: Process of the Second Subordinate Pass
CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

The mathematics in our thesis is motivated by recent advances in image processing, a part of electrical and computer engineering. In recent years, mathematical tools from signal processing have found new uses and refinements dictated by the digital processing of images; for example, speech in 1D-signals vs transmission of images in pixel form (so 2D) as generated by digital cameras. While there is a rich and time honored theory in the traditional case of 1D signals, images are 2D, and they present new challenges. Already the increase in the dimension presents new mathematical issues: refinements of the mathematical algorithms used earlier.

In the first part of the thesis we prove such results in the 2D case, and we explain their use. A key point we explore is the interplay between the two cases, continuous and discrete. Our discrete algorithms present fast matrix-operations to be applied to images in pixel form. This part of the thesis in turn is based on tools from wavelet analysis, and more generally from the theory of operators in Hilbert space.

In the second part, we address encoding and quantization. This is the last crucial step in A/D conversion, i.e., analog to digital. By quantization we mean the conversion and encoding of processing-output into bits; bits that in turn feed into a computer. We do this with a family of discrete algorithms, each one governed by a
quantizer. We then obtain a priori estimates. We isolate and make mathematically
precise a particular family of quantizers which are efficient in that they produce error
terms of exponential fall-off.

In the last part, we address the compression of a matrix (a 2-D image) obtained
by applying the DWT on an image mentioned in the first part. Embedded Zerotree
Wavelet algorithm is introduced and implemented.

5.2 Future Work

In the future, I would like to do more research about different quantization
algorithms like $\beta$-encoder. In PCM and $\sum \Delta$, truncated binary expansions are used.
However, $\beta$-encoders quantize a real number by computing one of its truncated $\beta$-
expansion where $\beta \in (1, 2)$. Yet most of $\beta$-encoders need to have a priori knowledge
of $\beta$. I would like to investigate if we can improve those $\beta$-encoders without knowing $\beta$.

I would like to investigate JPEG2000, SPIHT (Set Partitioning in Hierarchical
Trees), EZBC (Embedded Zero-Block Coding), or ESCOT (Motion-based Embedded
Sub-band Coding with Optimized Truncation) since they can provide a bit-stream
that can be flexibly adapted to the characteristics of the network and the receiving
device when combined with the wavelet-based video coding.

Wavelet-based financial time series forecasting is also an interesting topic that
I would like to do more research in. Forecasting relies on a predictive function (a
predictor) which uses a known part of a time series to generate a future prediction.
The amount of noise that is present in the known time series region that is used to generate the forecast value will have some influence on how close a forecast value is to the actual future value. On the other hand, wavelet compression can provide a way to estimate the amount of noise in a time series region and furthermore it can be used in de-noising. Combining both of them will be very interesting.
REFERENCES


