Spring 2000

Efficient algorithms and implementations for signal processing

Changbai Xiao

EFFICIENT ALGORITHMS AND IMPLEMENTATIONS FOR SIGNAL PROCESSING

by

Changbai Xiao, Ph.D.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING

LOUISIANA TECH UNIVERSITY

May 2000

UMI Number: 9967578

UMI®

UMI Microform9967578

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

May 9, 2000
Date

We hereby recommend that the dissertation prepared under our supervision

by Changbai Xiao

entitled Efficient Algorithms and Implementations for Signal Processing

be accepted in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy in Applied Computational Analysis and Modeling

Like Zou
Supervisor of Dissertation Research

Richard J. Greechie
Head of Department

Applied Computational Analysis & Modeling
Department (ACAM)

Recommendation concurred in:

Like Zou

Richard J. Greechie

Louis E Roemer

Ben CHOI

Advisory Committee

Approved:
Director of Graduate Studies

Approved:
Terry McConathy
Director of the Graduate School

Leslie K Guice
Dean of the College

ABSTRACT

A scheme is presented to regain a finite number of lost samples from a Nyquist-rate-sampled band-limited signal f of finite energy by replenishing new sample values of the same number. The result can also be viewed as the solution to a special non-uniform sampling problem.

A scheme is also presented to recover a band-limited function f of finite energy from its sampling values on real sequences with an accumulation point. The result given here can also be viewed as an approach to the extrapolation problem of determination a band-limited function in terms of its given values on a finite interval. An error estimate is also obtained.

The existence of two kinds of frames, Weyl-Heisenberg frames and affine frames, is studied. The conditions given in this dissertation improve the known conditions and, in addition, are easy to verify.

A parallel algorithm for the two-dimensional forward fast wavelet transform is developed and implemented on the AP1000 multiprocessor system. The algorithm is carefully analyzed before implementation. Experiments are performed on different input sizes on different numbers of processors. The results from the experiments coincide with the theoretical analysis. The parallel algorithm gains expected speedup on the mesh architecture. Further work is suggested.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Li-He Zou, my advisor, for his consistent help, encouragement, and professional guidance.

My thanks to Dr. Richard Greechie for his help in many aspects during my study in Louisiana Tech University.

My thanks to Dr. Barry Kurtz, Dr. Chinhyun Kim, and Dr. Mike Meehan for teaching me computer science.

My thanks to all the people at Louisiana Tech who have been helpful to my studying and teaching, especially Mr. Daniel W. Erickson, Dr. Richard Gibbs, Dr. Raja F. Nassar, Dr. Huaijin Gu, Dr. Chaoqun Liu, Dr. Weizhong Dai, Dr. Louis Roemer, Ms. Margaret A. Dunn, Mr. Daniel L. Schales, Ms. Joyce Bryan, and Ms. Janie M. Ainsworth.

The love and assistance from my wife, Tingting Li; my son, Peiwang Xiao; my daughter, Angela Beihong Xiao; and my parents have been a great inspiration throughout the project.

NOTATION

In this dissertation, **Z** and **R** denote the set of integers and the set of real numbers, respectively.

$L^2(\mathbf{R})$ denotes the Hilbert space of all complex-valued square-integrable functions on the real line **R** with the inner product

$$\langle f, g \rangle = \int_R f(t)\overline{g(t)}dt. \tag{1}$$

The Lebesgue measure of a set $E \subset \mathbf{R}$ is denoted by $|E|$.

A function (signal) $f(t)$ is said to be of finite energy and W-band-limited if $f \in L^2(\mathbf{R})$ and

$$\hat{f}(u) = 0 \qquad (u \notin [-W, W]) \tag{2}$$

i.e.,

$$f(t) = \frac{1}{2\pi}\int_{-W}^{W} \hat{f}(w)e^{iwt}dw \qquad (t \in \mathbf{R}). \tag{3}$$

where

$$\hat{f}(w) = \int_{-\infty}^{\infty} f(t)e^{-itw}dt \qquad (w \in \mathbf{R}) \tag{4}$$

is the Fourier transform of f.

$\mathbf{B}_W$ denotes the set of all W-band-limited functions of finite energy.

CHAPTER 1

INTRODUCTION

An important problem in signal processing is to reconstruct a signal from its sample values taken at non-uniformly spaced sampling points. Irregular sampling arises in many scientific fields, such as geophysics. astronomy, oceanography, medical imaging, spectroscopy, and speech processing. The concentration of this dissertation is to develop algorithms for signal recovery from non-uniform samples and to study frames and wavelets that are closely tied to signal processing and signal reconstruction. In this introductory chapter, some background on related fields is presented.

1.1 The Irregular Sampling Problem

The sampling problem is one of the standard problems in signal analysis. Since a signal $f(x)$ cannot be recorded in its entirety, it is sampled at a sequence $\{x_n : n \in \mathbf{Z}\}$. Then the question arises how f can be reconstructed or at least approximated from the samples $f(x_n)$.

In most applications, it is the case that the signal is band-limited and is of finite energy.

If the samples are equally spaced, then the famous sampling theorem by Shannon, Whittaker, Kotel'nikov, and others provides an explicit reconstruction. The Shannon, Whittaker, and Kotel'nikov sampling theorem [3, 27, 34, 41, 46] ensures that the finite energy W-band-limited signal $f(t)$ can be represented in terms of its uniformly sampled values $f(n/2W)_{n\in\mathbf{Z}}$ as

$$f(t) = \sum_{n=-\infty}^{\infty} f(\frac{n}{2W})\mathrm{sinc(2Wt - n)} \tag{1.1}$$

where

$$\mathrm{sinc(x)} = \frac{\sin(\pi \mathrm{x})}{\pi \mathrm{x}} \tag{1.2}$$

In practice, the samples are more likely non-uniformly spaced; this makes the irregular sampling problem a very important issue in signal processing. the irregular sampling problem was studied by many researchers [6, 18, 35, 36, 49]. The iterative reconstruction methods, spline interpolations, are employed to reconstruct signals from their non-uniform samples.

In Chapter 2, we present a scheme to regain a finite number of lost samples from a Nyquist-rate-sampled band-limited signal f of finite energy by replenishing new sample values of the same number. The algorithm is constructive and the result can also be viewed as the solution to a special non-uniform sampling problem. The advantage of the scheme is that it is programmable; therefore, it is easy to implement the algorithm in computer languages.

It is often important to recover a W-band-limited function f of finite energy from its sampling values at a convergent sequence of different points $x_n, n = 1, 2, ...,$ with at least one limit point $a \in \mathbf{R}$. In Chapter 3, a scheme is presented to deal with this form of irregular sampling problem. The result given in Chapter 3 can also be viewed as an approach to the extrapolation problem of determining a band-limited function in terms of its given values on a finite interval. An error estimate is also obtained.

1.2 Frames

Given a complex Hilbert space $\mathbf{H}$, it is often advantageous to find a family $\{h_n\}_{n\in\mathbf{Z}} \subset \mathbf{H}$ such that every element $h \in \mathbf{H}$ can be written as

$$h = \sum_n c_n h_n \tag{1.3}$$

for some complex scalars c_n. This kind of expansion is obviously available if the family $\{h_n\}_{n\in\mathbf{Z}}$ is chosen to be an orthonormal basis of $\mathbf{H}$. However, the requirement of orthogonality and the basis property is so restrictive that it is sometimes difficult to find an orthonormal basis for practical purposes. As an alternative, a generalization known as frames was introduced which has the property [10, 11, 26] that the expansion (1.3) is always possible.

Definition 1.2.1 *A set of functions $\{h_n\}_{n\in\mathbf{Z}}$ in a Hilbert space $\mathbf{H}$ is a frame if there exist constants $A, B > 0$ so that for all $h \in \mathbf{H}$,*

$$A\|h\|^2 \le \sum_n |\langle h, h_n\rangle|^2 \le B\|h\|^2. \tag{1.4}$$

The constants A and B are called the frame bounds.

Frames were introduced by Duffin and Schaeffer [15] for use in non-harmonic Fourier analysis. An excellent treatment of non-harmonic Fourier analysis can be found in Young's research [48]. Since then, there have been many contributions by Daubechies, Grossmann, Meyer, and others [10, 11, 26] developing the wavelet theory, emphasizing applications of frames in signal processing and finding conditions on the existence of frames for a Hilbert space $\mathbf{H}$.

Frames are closely tied to the irregular sampling problem. One of the existing reconstruction methods is to reconstruct signals from non-uniform samples using frames as the base technique [37].

Theorem 1.2.1 ([Marvasti]) *Suppose that*

$$A\|f\|_2^2 \leq \sum_{j=1}^{r} |f(t_j)|^2 \leq B\|f\|_2^2 \tag{1.5}$$

with $r \geq 2W+1$ *and assume* $\gamma < \frac{2}{A+B}$, *Then* f *can be reconstructed iteratively by*

$$f_0 = 0, f_{n+1} = f_n + \gamma S(f - f_n). \tag{1.6}$$

And $\lim_{n\to\infty} f_n = f$ *in* $L^2(\mathbf{R})$ *norm for* $f \in \mathbf{B}_W$, *where* Sx *is defined as*

$$Sx(t) = \sum_{j=1}^{r} x(t_j) sinc_W(t - t_j). \tag{1.7}$$

We are not dealing directly with the reconstruction algorithms using frames but providing schemes to form new frames. In chapter 4, the existence of two kinds of frames, Weyl-Heisenberg frames and affine frames, is studied. Precisely, we provide conditions on a function $g \in L^2(\mathbf{R})$ such that the regular Weyl-Heisenberg system $\{e^{2\pi imbx}g(x - na)\}_{m,n\in\mathbf{Z}}$ or the semi-irregular Weyl-Heisenberg system $\{e^{2\pi imbx}g(x - a_n)\}_{m,n\in\mathbf{Z}}$ with an arbitrary real sequence $\{a_n\}_{n\in\mathbf{Z}}$ forms a Weyl-Heisenberg frame for $L^2(\mathbf{R})$. Also, conditions are given on $g \in L^2(\mathbf{R})$ such that the affine system $\{a^{-n/2}g((x - mba^n)a^{-n})\}_{m,n\in\mathbf{Z}}$ forms an affine frame. The conditions given in this dissertation improve the known conditions and, in addition, are easy to verify.

1.3 Wavelets

The Fourier transform has been used as a powerful tool in application domains such as signal processing for many years. Nevertheless, it suffers from certain limitations. The Fourier transform of a signal f can only provide information of the frequency content of f; it provides little information concerning time-localization of the signal f. Small frequency changes in th Fourier transform will produce changes everywhere in the time domain. The Fourier transform provides poor time locality.

The Wavelet transform is a new tool that overcomes the weakness of the Fourier transform. A time-domain function (signal) can be transformed into a representation that is localized not only in the frequency domain (like the Fourier transform) but also in the time domain.

Wavelets are used in many fields such as chemical engineering [40], sub-band coding [1, 38], signal and image processing and compression [32, 44], and other applications [4, 7, 17, 39].

Functions can be represented using wavelets in a compact way. For instance, functions with discontinuities or sharp spikes usually take fewer wavelet terms than the sine-cosine basis functions that are used in Fourier analysis to achieve a comparable approximation. Therefore, in application domains such as data compression, wavelet transforms are more powerful than Fourier transforms.

The wavelet transform is used to decompose data or signals into different frequency components (*wavelets*). The original signal can be studied in terms of simpler wavelets. Informally, a wavelet is a "little wave;" the wavelet technique is to represent the "big wave" (signal) in terms of a set of well-chosen "little waves" so that the signal can be thoroughly studied through the study of simpler "little waves." Information about the original signal can therefore be extracted from the "little waves" which are actually functions built up with the translations and dilations (or modulations) of a signal function called the *mother wavelet* (sometimes it is also called the *analyzing wavelet*). Figure 1.1 shows the famous order-4 Daubechies mother wavelet.

Once the mother wavelet ψ is fixed, the translations and dilations of the mother wavelet $\{\psi((x-b)/a) : a > 0, b \in \mathbf{R}\}$ form the wavelet family which is used as the basis in representing other signals or functions. Different mother wavelets generate different wavelets. Different applications require different wavelets. Depending on requirements, one can choose among smooth wavelets, compactly supported wavelets, or wavelets with simple mathematical expressions, etc. Different wavelet families make different trade-offs between smoothness and compactness of basis functions.

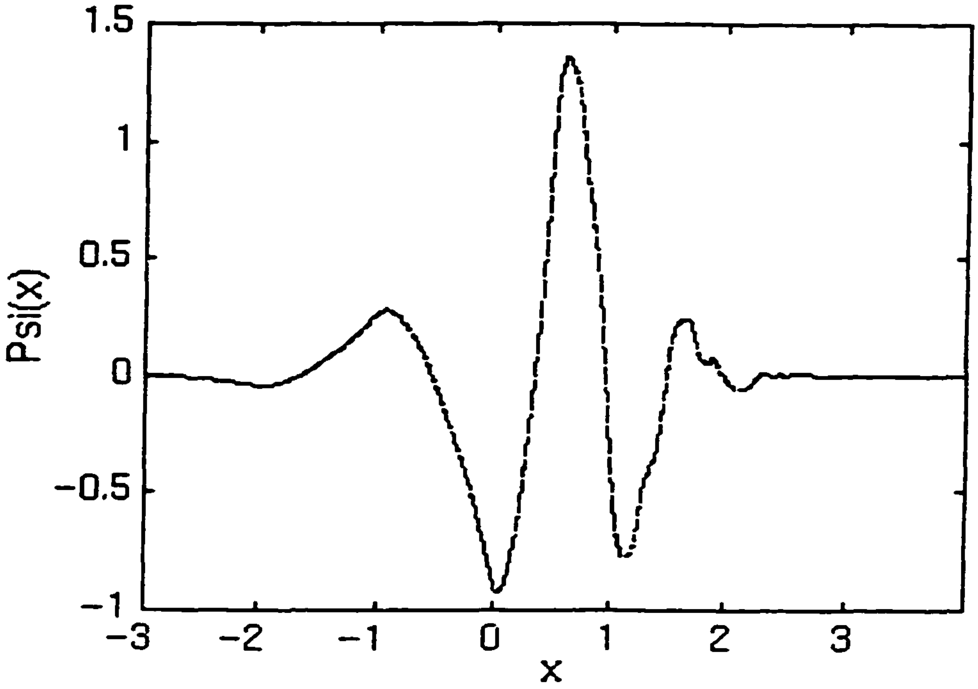

Figure 1.1 : Daubechies order-4 mother wavelets

Another advantage of the wavelet transform over the Fourier transform lies in the fact that the *fast wavelet transform (FWT)* is faster than the widely used *fast Fourier transform (FFT)*. It is well known that the computational complexity of the FFT is $O(n \cdot log(n))$ for an n-point transform. For the FWT, the computational complexity is $O(n)$ for an n-point transform.

Despite its favorable computational complicity, FWT becomes a time-consuming task when the data size becomes large. For example, video processing applications require processing of large amount of data within a specific time frame. To make a movie visually smooth, the interval between video frames should be short enough, thereby demanding a high image processing speed to compress and decompress the video data. Therefore, it is important to develop applicable parallel FWT algorithms.

The inherent parallelism of two-dimensional FWT is another important indication that parallel FWT algorithms deserve more attention.

In the FWT parallel algorithm we designed, p processors are applied to input data of size n. The mesh formulation is used, and the n input data is equally partitioned into $\sqrt{p} \times \sqrt{p}$ sub matrices. Each of the processors is assigned to process one of the sub-matrices. Each processor computes row-wise and column-wise 1-D FWT alternatively. Before each 1-D FWT computation, each processor communicates with a number of other processors (the number depends on the choice of wavelets) to get the data needed to perform the corresponding computation. The detailed algorithm and analysis will be presented in Section 5.2 and Section 5.3.

CHAPTER 2

RECONSTRUCTION OF BAND-LIMITED SIGNALS WITH LOST SAMPLES AT ITS NYQUIST RATE

2.1 The Problem

Band-limited signals of finite energy must be reconstructed from their sample values in many scientific and engineering problems. The Shannon, Whittacker, and Kotel'nikov sampling theorem [3, 27, 34, 41, 46] ensures that a finite energy band-limited signal uniformly samples at or above its Nyquist rate can be uniquely determined by its sample value. However, in practical digital recording of the sample values of a given signal, there is always a possibility of loss of samples. The lost samples have to be recovered some way if the signal is to be reconstructed completely. When a band-limited signal of finite energy is sampled beyond its Nyquist rate, various techniques developed by several authors [35, 36, 37] can be employed to restore a finite number of lost samples in terms of the remaining known samples. When a band-limited signal of finite energy is uniformly sampled exactly at its Nyquist rate, each signal sample is independent of every other sample value. With some of its samples lost, the signal cannot be uniquely determined only from the knowledge of the remaining samples. Here, we give a remedial measure and present a scheme for regaining a finite number of lost samples from a Nyquist-rate-sampled band-limited signal of finite energy by replenishing new sample values of the same number. The result can also be viewed as the solution to a special non-uniform sampling problem [19, 28, 38].

2.2 A Lemma

In order to properly state and prove our main result for the chapter, it is necessary to establish the following lemma, which is itself of interest from the mathematical point of view.

Lemma 2.2.1 *Let $N > 0$ be an integer. If $m_1, ..., m_N, k_1, ..., k_N$ are integers, $a_1, ..., a_N$ are real numbers within the open interval $(0, 1)$. Then the matrix*

$$S(N) = [s_{ij}]_{N\times N} = [(m_i + a_i - k_j)^{-1}]_{N\times N}$$

$$= \begin{bmatrix} \frac{1}{m_1+a_1-k_1} & \frac{1}{m_1+a_1-k_2} & \cdots & \frac{1}{m_1+a_1-k_N} \\ \frac{1}{m_2+a_2-k_1} & \frac{1}{m_2+a_2-k_2} & \cdots & \frac{1}{m_2+a_2-k_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{m_N+a_N-k_1} & \frac{1}{m_N+a_N-k_2} & \cdots & \frac{1}{m_N+a_N-k_N} \end{bmatrix} \tag{2.1}$$

is non-singular, i.e., the determinant of $S(N)$

$$det(S(N)) \neq 0 \tag{2.2}$$

if and only if

$$m_i + a_i \neq m_j + a_j \qquad (i \neq j, 1 \leq i, j \leq N) \tag{2.3}$$

and

$$k_i \neq k_j \qquad (i \neq j, 1 \leq i, j \leq N) \tag{2.4}$$

Proof. Using mathematical induction, we first prove that

$$det(S(N)) = A_1 A_2 ... A_N \tag{2.5}$$

with

$$A_n = \frac{\prod_{i=1}^{n-1}(k_i - k_n) \cdot \prod_{j=1}^{n-1}(m_n + a_n - m_j - a_j)}{\prod_{i=1}^{n}(m_i + a_i - k_n) \cdot \prod_{j=1}^{n-1}(m_n + a_n - k_j)} \qquad (2 \leq n \leq N) \tag{2.6}$$

and

$$A_1 = \frac{1}{m_1 + a_1 - k_1} \tag{2.7}$$

(a) If $N = 1$ (2.5) follows directly from (2.1).

(b) Suppose (2.5) is true for $N = K - 1$, i.e.,

$$det(S(K-1)) = A_1 A_2 ... A_{K-1} \tag{2.8}$$

Applying the elementary operations to $det(S(K))$, we get

$$det(S(K)) = det(([m_i + a_i - k_j)^{-1}]_{K\times K})$$

$$= \frac{1}{\prod_{j=1}^{K}(m_j + a_j - k_K)} \cdot \begin{bmatrix} \frac{m_1+a_1-k_K}{m_1+a_1-k_1} & \cdots & \frac{m_1+a_1-k_K}{m_1+a_1-k_{K-1}} & 1 \\ \cdots & \cdots & \cdots & \cdots \\ \frac{m_{K-1}+a_{K-1}-k_K}{m_{K-1}+a_{K-1}-k_1} & \cdots & \frac{m_{K-1}+a_{K-1}-k_K}{m_{K-1}+a_{K-1}-k_{K-1}} & 1 \\ \frac{m_K+a_K-k_K}{m_K+a_K-k_1} & \cdots & \frac{m_K+a_K-k_K}{m_K+a_K-k_{K-1}} & 1 \end{bmatrix}$$

$$= \frac{1}{\prod_{j=1}^{K}(m_j + a_j - k_K)} \cdot$$

$$\begin{bmatrix} \frac{(k_1-k_K)(m_K-m_1+a_K-a_1)}{(m_1+a_1-k_1)(m_K+a_K-k_1)} & \cdots & \frac{(k_{K-1}-k_K)(m_K-m_1+a_K-a_1)}{(m_1+a_1-k_{K-1})(m_K+a_K-k_{K-1})} & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \frac{(k_1-k_K)(m_K-m_{K-1}+a_K-a_{K-1})}{(m_{K-1}+a_{K-1}-k_1)(m_K+a_K-k_1)} & \cdots & \frac{(k_{K-1}-k_K)(m_K-m_{K-1}+a_K-a_{K-1})}{(m_{K-1}+a_{K-1}-k_1)(m_K+a_K-k_{K-1})} & 0 \\ \frac{m_K+a_K-k_K}{m_K+a_K-k_1} & \cdots & \frac{m_K+a_K-k_K}{m_K+a_K-k_{K-1}} & 1 \end{bmatrix}$$

$$= A_K \cdot det(S(K-1)). \tag{2.9}$$

It follows from (2.9) and (2.8) that

$$det(S(K)) = A_K A_{K-1} \ldots A_1 \tag{2.10}$$

It means that (2.5) is also true for $N = K$ if it is supposed true for $N = K-1$. Thus, based on the principle of the mathematical induction, (2.5) is true for every integer $N \geq 1$.

For the "if" part of the lemma, we need to prove that (2.2) can be deduced from (2.3) and (2.4). It is easy to see that

$$m_i + a_i - k_j \neq 0 \qquad (1 \leq i, j \leq N) \tag{2.11}$$

because the m_i's and k_j's are integers while the a_i's are decimals within $(0, 1)$. If (2.3) and (2.4) are true, then we can conclude from (2.11), (2.7), and (2.6) that

$$A_n \neq 0 \qquad (1 \leq n \leq N) \tag{2.12}$$

Therefore, (2.2) follows from (2.5) and (2.12).

For the "only if" part of the lemma, we need to deduce (2.3) and (2.4) from (2.2). From (2.5), we know (2.2) ensures (2.12). And now (2.12) and (2.6) imply (2.3) and (2.4).

2.3 The Solution

Let $S = \{n_j : 1 \leq j \leq N\}$ be a finite set of N integers. When a W-band-limited signal f of finite energy is sampled at its Nyquist rate, if the set of samples

$\{f(n/2W) : n \in \mathbf{Z} - S\}$ is known while the samples $\{f(n_j/2W) : 1 \leq j \leq N\}$ are lost, Then the problem at hand is to recover the missing samples so that the signal f can be rebuilt by means of (1.1). We realize this by way of adding N new samples other than $\{f(n/2W) : n \in \mathbf{Z}\}$.

Take arbitrarily N different points $y_1, ..., y_N$ from the set $\mathbf{R} - \{n/2W : n \in \mathbf{Z}\}$. For $1 \leq j \leq N$, Let

$$k_j = n_j - n_1, \quad m_j = [2Wy_j - n_1] \quad and$$

$$a_j = 2Wy_j - n_1 - m_j \tag{2.13}$$

where $[x]$ denotes the largest integer less than x. It follows from (2.13) that

$$2Wy_i - n_j = m_i + a_i - k_j \qquad (1 \leq i, j \leq N). \tag{2.14}$$

Let U be the matrix

$$U = [\mathrm{sinc}(2\mathrm{W}\mathrm{y}_\mathrm{i} - \mathrm{n}_\mathrm{j})]_{N \times N} = [\mathrm{sinc}(\mathrm{m}_\mathrm{i} + \mathrm{a}_\mathrm{i} - \mathrm{k}_\mathrm{j})]_{\mathrm{N}\times\mathrm{N}}. \tag{2.15}$$

Since $\{y_j : 1 \leq j \leq N\} \bigcap \{n/2W : n \in \mathbf{Z}\} = \emptyset$, we know the a_j's defined in (2.13) satisfy

$$0 < a_j < 1 \qquad (1 \leq j \leq N). \tag{2.16}$$

It can also be verified that the integers $m_1, ..., m_N, k_1, ..., k_N$ and the real numbers $a_1, ..., a_N$ defined in (2.13) satisfy (2.3) and (2.4).

Applying the lemma in the last section, we get

$$\begin{aligned} det(U) &= det(((-1)^{m_i - K_j}\mathrm{sin}(\pi \mathrm{a}_\mathrm{i})/(\mathrm{m}_\mathrm{i} + \mathrm{a}_\mathrm{i} - \mathrm{k}_\mathrm{j}))_{\mathrm{N}\times\mathrm{N}}) \\ &= (-1)^{m_1 + ... + m_N - k_1 ... - k_N} \cdot \prod_{i=1}^{N} \sin(\pi a_i) det(S(N)) \\ &\neq 0 \end{aligned} \tag{2.17}$$

Let $\vec{f}$ be the N-dimensional vector of the lost samples arranged in increasing order of index, i.e.,

$$\vec{f} = (f(n_1/2W), ..., f(n_N/2W))^T \tag{2.18}$$

where x^T denotes the transpose of the vector x. Let

$$\vec{f_0} = (f(y_1), ..., f(y_N))^T \tag{2.19}$$

and $\vec{h_0}$ be the vector of linear combinations of the samples $\{f(n/2W) : n \in \mathbf{Z} - S\}$ as follows

$$\vec{h_0} = \begin{bmatrix} \sum_{n \in \mathbf{Z}-S} f(n/2W)\mathrm{sinc}(2\mathrm{Wy}_1 - \mathrm{n}) \\ \cdot \\ \cdot \\ \cdot \\ \sum_{n \in \mathbf{Z}-S} f(n/2W)\mathrm{sinc}(2\mathrm{Wy}_N - \mathrm{n}) \end{bmatrix} \tag{2.20}$$

Now the problem is to represent f in terms of $\vec{f_0}$ and $\vec{h_0}$.

Theorem 2.3.1 *If $y_j \in \mathbf{R} - \{n/2W : n \in \mathbf{Z}\}$ $(1 \leq j \leq N)$, then*

$$\vec{f} = U^{-1}(\vec{f_0} - \vec{h_0}) \tag{2.21}$$

where U is the matrix defined in (2.15).

Proof. Take $t = y_i$ in (1.1) for $i = 1, ..., N$, respectively, we get

$$\begin{aligned} f(y_i) &= \sum_{j=1}^{N} f(n_j/2W)\mathrm{sinc}(2\mathrm{Wy}_i - \mathrm{n}_j) \\ &\quad + \sum_{n \in \mathbf{Z}-S} f(n/2W)\mathrm{sinc}(2\mathrm{Wy}_i - \mathrm{n}). \end{aligned} \tag{2.22}$$

By the definition of $\vec{f}$, $\vec{f_0}$, $\vec{h_0}$ and U, we can write the system of linear equations (2.22) in the form

$$U\vec{f} = \vec{f_0} - \vec{h_0}. \tag{2.23}$$

Equation (2.23) reveals that the matrix U is non-singular. Thus, we have

$$\vec{f} = U^{-1}(\vec{f_0} - \vec{h_0}). \tag{2.24}$$

We obtain (2.21), and the proof is complete.

2.4 A Non-Uniform Sampling Problem

The result obtained in the above section can be viewed as the solution to the special non-uniform sampling problem of reconstructing constructively a finite energy W-band-limited signal $f(t)$ from its Nyquist-rate-sampled values $\{f(n/2W) : n \in \mathbf{Z} - S\} \bigcup \{f(y_i) : 1 \leq i \leq N\}$, where N is a positive integer, S is a set of N integers, and $y_i (1 \leq i \leq N)$ are arbitrarily chosen numbers other than $n/2W (n \in \mathbf{Z})$.

Corollary 2.4.1 *The signal $f(t)$ is uniquely determined and can be represented by it samples at the non-uniform sampling set*

$$Y = \{n/2W : n \in \mathbf{Z} - S\} \bigcup \{y_i : 1 \leq i \leq N\} \tag{2.25}$$

CHAPTER 3

RECONSTRUCTION OF BAND-LIMITED FUNCTIONS FROM VALUES ON REAL SEQUENCES WITH AN ACCUMULATION POINT

3.1 The Problem

Let $f : \mathbf{R} \longrightarrow \mathbf{C}$ be an W-band-limited complex-valued function of finite energy on the real line $\mathbf{R}$, i.e., $f \in L^2(\mathbf{R})$ and $\hat{f}(w) = 0$ outside $[-W, W]$, where $W > 0$ and

$$\hat{f}(w) = \int_{-\infty}^{\infty} f(t)e^{-itw}dt \qquad (w \in \mathbf{R}) \tag{3.1}$$

is the Fourier transform of f. We have

$$f(t) = \frac{1}{2\pi}\int_{-W}^{W} \hat{f}(w)e^{iwt}dw \qquad (t \in \mathbf{R}). \tag{3.2}$$

By the Paley-Wiener theorem [5, pp. 103], f can be viewed as the restriction of the entire function of exponential type at most W:

$$f(z) = \frac{1}{2\pi}\int_{-W}^{W} \hat{f}(w)e^{iwz}dw \qquad (z \in \mathbf{C},\ \textit{the complex plane}) \tag{3.3}$$

to the real line R (so a band-limited function is always continuous). Theoretically, the uniqueness theorem for analytic functions implies that f can be wholly determined by its values at any sequence of different (interpolating) points with accumulation points in R. The goal of this chapter is to present a scheme to recover a W-band-limited functior. f of finite energy from its sampling values at a convergent sequence of different points $x_n, n = 1, 2, ...,$ with limit $a \in \mathbf{R}$.

This can be viewed as an approach to the (irregular) sampling problem of band-limited functions which asks under which conditions and how a band-limited function can be rebuilt if it is known only at a discrete set of points. Because of its great importance in information theory, signal processing and other application fields, a significant body of work has been carried out on (regular and irregular) sampling problems [3, 6, 19, 42]. We note that sampling theorems in literature demand that the sampling points are dense enough and well scattered for the regular case or "relatively well scattered" for the irregular case on the whole line R. The discrete set of sampling points is equally spaced for the classical Shannon, Whittaker, and Kotel'nikov (regular) sampling theorem, and some kind of density (e.g., δ-density used in [19]) for the sampling set is required for irregular sampling problems. Our case is quite different; this kind of requirement is not demanded here. This is one characteristic of the scheme presented in this chapter. Because our sampling (interpolating) points $x_n, n = 1, 2, \ldots$ converge to a limit a, all but finitely many sampling points will be within an interval, say $[\alpha, \beta]$. Thus, our result can also be viewed as an approach to the extrapolation problem of determining a band-limited function in terms of its given values on an interval $[\alpha, \beta] \subset R$ [43]. So, another characteristic of the recovery scheme presented here is that it can be used to rebuild the function from its value on any non-empty interval. As the referees pointed out, this scheme has drawbacks: to calculate the first n coefficients of the series, one needs to solve n linear equations in n unknowns, and the stability cannot be ensured. From this point of view, it would be proper to view our scheme more as an extrapolation method than as a sampling method.

3.2 The Solution

Suppose $\{x_n : n = 1, 2, ...\} \subset R, x_j \neq x_k (j \neq k)$, and $x_n \to a(n \to \infty)$. If $\{f(x_n) : n = 1, 2, ...\}$ are known, we hope to reconstruct f from $\{f(x_n) : n = 1, 2, ...\}$.

Since we can consider $f_a(x) = f(x + a)$ with the sampling points $\{x_n - a, n = 1, 2, ...\}$, we may assume $a = 0$ without loss of generality. Furthermore, we may assume $\{x_n\}$ to be monotone decreasing because otherwise we can consider a monotone decreasing convergent subsequence of $\{x_n\}$ or a monotone decreasing convergent subsequence of $\{-x_n\}$ and $f^-(x) = f(-x)$.

Theorem 3.2.1 *Let $f : \mathbf{R} \to \mathbf{C}$ be a W-band-limited function of finite energy, $\{x_n : n = 1, 2, ...\}$ be a monotone decreasing sequence and $x_n \to 0 (n \to \infty)$; then*

$$\phi_m(t) = \sum_1^m E_m(k) e^{ikWt/m} e^{-iWt} \tag{3.4}$$

converges to $f(t)$ uniformly on each compact subset $S \subset R$ when $m \to \infty$, where the coefficients $E_m(k)'s$ are chosen so that the interpolation equations

$$f(x_n) = \phi_m(x_n) \qquad (1 \leq n \leq m) \tag{3.5}$$

are satisfied, i.e., $\{E_m(k) : k = 1, ..., m\}$ is the solution of the system of linear equations

$$f(x_n) = \sum_1^m E_m(k) e^{ikWx_n/m} e^{-iWx_n} \qquad (1 \leq n \leq m). \tag{3.6}$$

Proof. For simplicity, we only prove the theorem for $x_n = 1/n (n = 1, 2, ...)$. The general case can be proven similarly.

Under the above assumption, (3.6) becomes

$$f(1/n) = \sum_1^m E_m(k) e^{ikW/(mn)} e^{-iW/n} \qquad (1 \leq n \leq m). \tag{3.7}$$

It is easy to verify that the determinant of coefficients for (3.7) is non-singular, so the system of linear equations (3.7) has a unique solution $\{E_m(k) : k = 1, ..., m\}$.

Take real numbers r_1 and r_2 such that

$$0 < r_1 < r_2 < 1 \textit{ and } 0 < \frac{2r_1(1+r_2)^2}{r_2 - r_1} < 1. \tag{3.8}$$

For each $m \geq 1$, the function

$$g_m(z) = z^m \cdot \frac{1}{2\pi} \int_{-W}^{W} \hat{f}(w) z^{mw/W} dw \quad (z \in \mathbf{C}) \tag{3.9}$$

and the polynomial

$$p_m(z) = \sum_{1}^{m} E_m(k) z^k \qquad (z \in \mathbf{C}) \tag{3.10}$$

are analytic on $D_2 = \{z : |z-1| < r_2\} \supset D_1 = \{z : |z-1| < r_1\}$.

Take a positive integer M such that the complex numbers

$$z_{m,n} = e^{iW/(mn)} \qquad (1 \leq n \leq m) \tag{3.11}$$

are in D_1 when $m > M$. It follows from (3.9), (3.2), (3.7), and (3.10) that

$$\begin{aligned} g_m(z_{m,n}) &= (z_{m,n})^m \cdot \frac{1}{2\pi} \int_{-W}^{W} \hat{f}(w)(z_{m,n})^{mw/W} dw \\ &= e^{iW/n} \cdot \frac{1}{2\pi} \int_{-W}^{W} \hat{f}(w) e^{iw/n} dw \\ &= e^{iW/n} f(1/n) \\ &= e^{iW/n} \sum_{k=1}^{m} E_m(k) e^{ikW/(mn)} e^{-iW/n} \\ &= p_m(z_{m,n}). \end{aligned} \tag{3.12}$$

Thus $p_m(z)$ is the interpolating polynomial of degree at most m with the values $\{g_m(z_{m,n}), n = 1, \cdots, m\}$ at the points $\{z_{m,n}, n = 1, ..., m\}$. By the Hermite theorem [14, p.68], we have for $m > M$ and $z \in D_1$

$$g_m(z) - p_m(z) = \frac{1}{2\pi i} \int_{C_2} \frac{(z - z_{m,1})(z - z_{m,2}) \cdots (z - z_{m,m}) g_m(t)}{(t - z_{m,1})(t - z_{m,2}) \cdots (t - z_{m,m})(t - z)} dt, \tag{3.13}$$

where $C_2 = \{z : |z-1| = r_2\}$ is the boundary of D_2. It follows directly from (9) that

$$\begin{aligned} |g_m(t)| &\leq \frac{(1+r_2)^m}{2\pi} \int_{-W}^{W} |\hat{f}(w)|(1+r_2)^{m|w|/W} dw \\ &\leq (1+r_2)^{2m} \frac{1}{2\pi} \int_{-W}^{W} |\hat{f}(w)| dw \end{aligned}$$

$$= A(1+r_2)^{2m} \qquad (m > M, t \in C_2) \tag{3.14}$$

with the constant

$$A = \frac{1}{2\pi}\int_{-W}^{W} |\hat{f}(w)|dw \leq \frac{1}{2\pi}(\int_{-W}^{W} |\hat{f}(w)^2|dw)^{1/2} \cdot (2W)^{1/2} \leq \infty \tag{3.15}$$

based on the Cauchy-Schwarz inequality and the fact that f is of finite energy. Furthermore, we can verify for $m > M$, $1 \leq n \leq m$, $t \in C_2$ and $z \in D_1$ that

$$|z - z_{m,n}| \leq 2r_1, \quad |t - z_{m,n}| \geq r_2 - r_1, \quad |t - z| \geq r_2 - r_1. \tag{3.16}$$

Combining (3.14), (3.15), (3.16) and (3.13), we obtain

$$|g_m(z) - p_m(z)| \leq (\frac{2Ar_2}{r_2 - r_1})(\frac{2r_1(1+r_2)^2}{(r_2 - r_1)})^m. \tag{3.17}$$

Thus it follows from (3.8) and (3.17) that

$$lim_{m\to\infty}|g_m(z) - p_m(z)| = 0 \tag{3.18}$$

uniformly for $z \in D_1$.

For any fixed compact subset $S \subset R$, take a positive number T such that $S \subset [-T, T]$. It is not difficult to find a positive integer $N > M$ such that $|e^{iWt/m} - 1| < r_1$ for all $m > N$ and $t \in [-T, T]$; this means

$$z(m,t) = e^{iWt/m} \in D_1 \tag{3.19}$$

for all $m > N$ and $t \in [-T, T]$. Thus it follows from (3.18) and (3.19) that

$$lim_{m\to\infty}|g_m(z(m,t)) - p_m(z(m,t)| = 0 \tag{3.20}$$

uniformly for $t \in S$. But from (9) and (2), we have

$$g_m(z(m,t)) = e^{iWt} \cdot \frac{1}{2\pi}\int_{-W}^{W} \hat{f}(w)e^{iwt}dw = e^{iWt}f(t). \tag{3.21}$$

and, from (3.10) and (3.4), we have

$$p_m(z(m,t)) = \sum_1^m E_m(k)e^{ikWt/m} = e^{iWt}\phi_m(t). \tag{3.22}$$

Thus (3.20), (3.21), and (3.22) tell us that

$$lim_{m\to\infty}|\phi_m(t) - f(t)| = 0 \tag{3.23}$$

uniformly for $t \in S \subset [-T,T]$.

3.3 Error Estimate

We deal with the special case for $\{x_n\} = \{1/n\}$ because we can handle the general case similarly.

Theorem 3.3.1 *The assumptions are the same as those in theorem 1 with* $1/n$ *replacing* $x_n(n = 1, 2, ...)$. *For any* $0 < \lambda < 1$ *and* $T > 0$, *we can find a positive integer* N *such that, for any* $m > N$, *we have the error estimate*

$$|\phi_m(t) - f(t)| < \frac{22\sqrt{\pi}}{9\pi}\sqrt{WE}\lambda^m. \quad (t \in [-T,T]), \tag{3.24}$$

where E *is the energy of* f, *i.e.,*

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi}\int_{-W}^{W} |\hat{f}(w)|^2 dw. \tag{3.25}$$

Proof. Take $r_1 = \frac{\lambda}{9+2\lambda}$ and $r_2 = \frac{1}{2}$, then

$$0 < r_1 < r_2 < 1 \text{ and } \frac{2r_1(1+r_2)^2}{r_2 - r_1} = \lambda. \tag{3.26}$$

It follows from (3.21), (3.22), (3.17), (3.26), and (3.15) that there is a positive integer N such that, for any $m > N$, and $t \in [-T,T]$,

$$\begin{aligned} |\phi_m(t) - f(t)| &\le \frac{2r_2A}{r_2 - r_1}\lambda^m \\ &\le \frac{2(9+2\lambda)}{9}\sqrt{\frac{WE}{\pi}}\lambda^m \\ &< \frac{22\sqrt{\pi}}{9\pi}\sqrt{WE}\lambda^m. \end{aligned} \tag{3.27}$$

CHAPTER 4

ON THE EXISTENCE OF WEYL-HEISENBERG AND AFFINE FRAMES IN $L^2(\mathbf{R})$

4.1 The Problem

In some applications in quantum mechanics (e.g., the theory of coherent states) and in signal analysis, the natural choice for the Hilbert space $\mathbf{H}$ is $L^2(\mathbf{R})$, and the following three kinds of expansions for $f \in L^2(\mathbf{R})$ are often useful:

1. for some fixed $g \in L^2(\mathbf{R}), a, b \in \mathbf{R}$,

$$f(x) = \sum_{m,n} a_{mn} \left(e^{2\pi imbx} g(x - na)\right), \tag{4.1}$$

2. for some fixed $g \in L^2(\mathbf{R}), a > 1$ and $b > 0$,

$$f(x) = \sum_{m,n} b_{mn} \left(e^{2\pi imba^n x} a^{n/2} g(a^n x)\right); \tag{4.2}$$

3. for some fixed $g \in L^2(\mathbf{R}), a > 1$ and $b > 0$,

$$f(x) = \sum_{m,n} d_{mn} \left(a^{-n/2} g((x - mba^n)a^{-n})\right). \tag{4.3}$$

We are therefore interested in finding conditions on the function g such that the set of functions leading to the expansion in (4.1), (4.2); or (4.3) forms a frame. The set $\{e^{2\pi imbx} g(x - na)\}_{m,n\in\mathbf{Z}}$ used in expansion (4.1), generated from the translations and modulations of a function g, is referred to as a Weyl-Heisenberg (W-H), Gabor, or windowed Fourier system. If a Weyl-Heisenberg system forms a frame for $L^2(\mathbf{R})$, it is called a W-H frame. The functions $a^{-n/2}g((x-mba^n)a^{-n}), m, n \in \mathbf{Z}$, used in expansion (4.3), are often called wavelets, which arise as translations and dilations of a function g. If the set $\{a^{-n/2}g((x-mba^n)a^{-n})\}_{m,n\in\mathbf{Z}}$ forms a frame for $L^2(\mathbf{R})$, it is called an affine frame.

In this dissertation, we generalize the sufficient conditions of existence found by Daubechies and Walnut [10, 27]. Our proofs are elementary, and the conditions we find are easy to verify. In Section 2 we deal with the Weyl-Heisenberg case. The affine cases associated with the expansions (4.2) and (4.3) are handled together in Section 3. Finally, in Section 4 we consider semi-irregular W-H frames of the form $\{e^{2\pi imbx}g(x-a_n)\}_{m,n\in\mathbf{Z}}$ or $\{e^{2\pi ia_nx}g(x-mb)\}_{m,n\in\mathbf{Z}}$ with an arbitrary real sequence $\{a_n\}_{n\in\mathbf{Z}}$.

4.2 The Weyl-Heisenberg Case

4.2.1 Definitions and Some Known Results

To study the set of functions $\{e^{2\pi imbx}g(x-na)\}_{m,n\in\mathbf{Z}}$, it is convenient to introduce the following operators:

- Modulation : $\mathbf{E}_a f(x) = e^{2\pi iax}f(x)$ $\qquad (a \in \mathbf{R})$,
- Translation : $\mathbf{T}_b f(x) = f(x-b)$ $\qquad (b \in \mathbf{R})$,

Definition 4.2.1 *Given $g \in L^2(\mathbf{R})$ and $a, b > 0$, we say that (g, a, b) generates a W-H (Weyl-Heisenberg) frame for $L^2(\mathbf{R})$ if $\{(\mathbf{E}_{mb}\mathbf{T}_{na}g)\}_{m,n\in\mathbf{Z}}$ is a frame for $L^2(\mathbf{R})$. The function g is the W-H generator. The numbers a and b are the time and frequency step size.*

Definition 4.2.2 *The amalgam space $W(L^\infty, \ell^1)$ is the collection of all functions g such that*

$$\|g\|_{w,a} = \sum_n \|g \cdot \chi_{[na,(n+1)a)}\|_\infty < \infty$$

for some $a > 0$.

Existence conditions for W-H frames can be found in the research of Daubechies [10, 12], the research of Daubechies, Grossmann, and Meyer [13], or the research-tutorial of Heil and Walnut [27]. Among those results, we state the following:

Theorem 4.2.1 [10, 27] *Let $g \in L^2(\mathbf{R})$ and $a > 0$ be such that*

(1) There are constants A and B such that

$$0 < A < \sum_n |g(x-na)|^2 \leq B < \infty \quad a.e., \tag{4.4}$$

(2) $lim_{b\to 0} \sum_{k\neq 0} \beta(k/b) = 0$, where

$$\beta(s) = esssup_{x\in\mathbf{R}} \left|\sum_n g(x-na)\overline{g(x-na-s)}\right|. \tag{4.5}$$

Then there exists $b_0 > 0$ such that (g, a, b) generates a W-H frame for $L^2(\mathbf{R})$ for each $0 < b \leq b_0$.

Theorem 4.2.2 ([27]) *If $g \in W(L^\infty, \ell^1)$ satisfies condition (1) of Theorem 4.2.1 for some $a > 0$, then there is a $b_0 > 0$ such that (g, a, b) generates a W-H frame for $L^2(\mathbf{R})$ for all $0 < b \leq b_0$.*

4.2.2 Conditions for W-H Generators

We now state and prove our theorems about conditions on a function $g \in L^2(\mathbf{R})$ such that (g, a, b) generates a W-H frame for $L^2(\mathbf{R})$.

Theorem 4.2.3 *Let $g \in L^2(\mathbf{R})$ and $a, b > 0$ be such that condition (1) of Theorem 4.2.1 is satisfied. Assume there exists a constant $0 < \tilde{A} < A$ such that*

$$\sum_{k \neq 0} \sum_{n} |g(x - na)g(x - na - k/b)| \leq \tilde{A} \quad a.e. \tag{4.6}$$

Then (g, a, b) generates a W-H frame for $L^2(\mathbf{R})$ with the frame bounds $(A - \tilde{A})/b$, $(B + \tilde{A})/b$.

Proof. First, assume that f is continuous and compactly supported. This guarantees that all subsequent interchanges of summation and integration are justified. Define the $1/b$-periodic function

$$F_n(t) = \sum_k f(t - k/b)\, \overline{g(t - na - k/b)}.$$

Using the techniques of Theorem 4.1.5 in [27], in particular, the fact that $\{b^{1/2} e^{-2\pi imbt}\}_{m \in Z}$ is an orthonormal basis for $L^2[0, 1/b]$ and the Plancherel formula, one can show that

$$\begin{aligned}
&\sum_n \sum_m |\langle f, \mathbf{E}_{mb}\mathbf{T}_{na}g\rangle|^2 \\
&= b^{-1} \int_{\mathbf{R}} |f(t)|^2 \sum_n |g(t - na)|^2 \, dt \\
&\quad + b^{-1} \sum_{k \neq 0} \int_{\mathbf{R}} \overline{f(t)} f(t - k/b) \sum_n g(t - na)\, \overline{g(t - na - k/b)} \, dt \\
&= b^{-1}[(*) + (**)].
\end{aligned} \tag{4.7}$$

Applying the Cauchy-Schwarz inequality twice, we have

$$\begin{aligned}
|(**)| &\leq \sum_{k \neq 0} \sum_n \int_{\mathbf{R}} |f(t)| \left(|\mathbf{T}_{na}g|\, |\mathbf{T}_{na+k/b}g|\right)^{1/2} |\mathbf{T}_{k/b}f| \left(|\mathbf{T}_{na}g|\, |\mathbf{T}_{na+k/b}g|\right)^{1/2} dt \\
&\leq \sum_{k \neq 0} \sum_n \left(\int_{\mathbf{R}} |f(t)|^2 |\mathbf{T}_{na}g|\, |\mathbf{T}_{na+k/b}g| \, dt\right)^{1/2} \left(\int_{\mathbf{R}} |f(t)|^2 |\mathbf{T}_{na}g|\, |\mathbf{T}_{na-k/b}g| dt\right)^{1/2} \\
&\leq \left(\sum_{k \neq 0} \sum_n \int_{\mathbf{R}} (|f(t)|^2 |\mathbf{T}_{na}g|\, |\mathbf{T}_{na+k/b}g| \, dt\right)^{1/2} \left(\sum_{k \neq 0} \sum_n \int_{\mathbf{R}} |f(t)|^2 |\mathbf{T}_{na}g|\, |\mathbf{T}_{na-k/b}g| \, dt\right)^{1/2} \\
&= \left(\int_{\mathbf{R}} |f(t)|^2 \sum_{k \neq 0} \sum_n |\mathbf{T}_{na}g|\, |\mathbf{T}_{na+k/b}g| dt\right)^{1/2} \left(\int_{\mathbf{R}} |f(t)|^2 \sum_{k \neq 0} \sum_n |\mathbf{T}_{na}g|\, |\mathbf{T}_{na+k/b}g| \, dt\right)^{1/2} .
\end{aligned}$$

Thus it follows from (4.6) that

$$|(**)| \leq \tilde{A}\|f\|^2. \tag{4.8}$$

Making use of (4.7), (4.8), and inequality (4.4), we obtain

$$\sum_n \sum_m |\langle f, \mathbf{E}_{mb}\mathbf{T}_{na}g\rangle|^2 \leq b^{-1}(B + \tilde{A})\|f\|^2 \tag{4.9}$$

and

$$\sum_n \sum_m |\langle f, \mathbf{E}_{mb}\mathbf{T}_{na}g\rangle|^2 \geq b^{-1}(A - \tilde{A})\|f\|^2. \tag{4.10}$$

We have shown that inequalities (4.9) and (4.10) are satisfied for any function f in $C_c(\mathbf{R})$, the space of continuous and compactly supported functions on $\mathbf{R}$. Since $C_c(\mathbf{R})$ is dense in $L^2(\mathbf{R})$, it follows that (4.9) and (4.10) hold for any function $f \in L^2(\mathbf{R})$. Hence, (g, a, b) generates a W-H frame with $(A-\tilde{A})/b$, $(B+\tilde{A})/b$ as the frame bounds.

Theorem 4.2.4 *Let $g \in L^2(\mathbf{R})$ and $a > 0$ be such that*

$$0 < A_1 \leq \sum_n |g(x - na)| \leq B_1 < \infty \quad a.e. \tag{4.11}$$

If the series $\sum_n |g(x - na)|$ converges uniformly for almost all $x \in [0, a]$, then there exists $0 < b_0 \leq 1/a$ such that (g, a, b_0) generates a W-H frame for $L^2(\mathbf{R})$.

Proof. First we prove that there exist constants $A, B > 0$ such that g satisfies condition (1) of Theorem 4.2.1. By hypothesis, there exists a subset V of $[0, a]$ with $|V| = 0$ such that the series $\sum_n |g(x - na)|$ converges uniformly on $[0, a] \setminus V$, and such that $A_1 \leq \sum_n |g(x - na)| \leq B_1$ for all $x \in [0, a] \setminus V$. So we can find an integer $N_0 > 0$ such that

$$\sum_{|n| \leq N_0} |g(x - na)| \geq A_1/2 \quad \textit{for}\, x \in [0, a] \setminus V.$$

Therefore, for any $x \in [0, a] \setminus V$ there must be an integer N_x such that $|N_x| \leq N_0$ and

$$|g(x - N_x a)| \geq \frac{A_1}{2(2N_0 + 1)}.$$

If we set $A = \left(\frac{A_1}{2(2N_0+1)}\right)^2$, then for any $x \in [0, a] \setminus V$, we have

$$0 < A \leq |g(x - N_x a)|^2 \leq \sum_n |g(x - na)|^2. \tag{4.12}$$

Since $\sum_n |g(x - na)|$ is an a-periodic function, inequality (4.12) holds for almost all $x \in \mathbf{R}$. On the other hand, it is obvious that

$$\sum_n |g(x - na)|^2 \leq \left(\sum_n |g(x - na)|\right)^2 \leq B_1^2 = B < \infty,$$

from which condition (1) of Theorem 4.2.1 is ensured.

Second, by Theorem 4.2.3 we only need to prove that for any $\tilde{A} < A$ there exists a constant b_0 such that $0 < b_0 \leq 1/a$ and

$$\sum_{k \neq 0} \sum_n |g(x - na)||g(x - na - k/b_0)| \leq \tilde{A} \quad a.e. \tag{4.13}$$

Let $\varepsilon_0 = \tilde{A}/(2B_1)$ and let the integer $N_1 > 0$ be so large that

$$\sum_{|n| \geq N_1} |g(x - na)| \leq \varepsilon_0 \tag{4.14}$$

for all $x \in [0, a] \setminus V$. If we take $b_0 = 1/(2N_1 a)$, then for any $|n| < N_1$ we have $|n + 2kN_1| \geq N_1$ for all integers $k \neq 0$. Therefore, for $x \in [0, a] \setminus V$ we have

$$\begin{aligned} \sum_{k \neq 0} |g(x - na - k/b_0)| &= \sum_{k \neq 0} |g(x - na - 2kN_1 a)| \\ &\leq \sum_{|n| \geq N_1} |g(x - na)| \leq \varepsilon_0. \end{aligned} \tag{4.15}$$

Thus, for any $x \in [0, a] \setminus V$, we have

$$\begin{aligned} &\sum_{k \neq 0} \sum_n |g(x - na)||g(x - na - k/b_0)| \\ &= \sum_{k \neq 0} \sum_{|n| \geq N_1} |g(x - na)||g(x - na - k/b_0)| \\ &\qquad + \sum_{k \neq 0} \sum_{|n| < N_1} |g(x - na)||g(x - na - k/b_0)| \\ &= \sum_{|n| \geq N_1} \left(|g(x - na)| \sum_{k \neq 0} |g(x - na - 2kN_1 a)|\right) \end{aligned}$$

$$
\begin{aligned}
&+ \sum_{|n|<N_1} \left(|g(x-na)| \sum_{k\neq 0} |g(x-na-k/b_0)| \right) \\
&\le \sum_{|n|\ge N_1} |g(x-na)|B_1 + \sum_{|n|<N_1} |g(x-na)|\varepsilon_0 \\
&\le B_1\varepsilon_0 + B_1\varepsilon_0 = 2B_1\varepsilon_0 = \tilde{A}.
\end{aligned}
$$

This completes the proof.

Daubechies showed in [9] that condition (1) of Theorem 4.2.1 is a necessary condition for (g, a, b) to generate a frame for $L^2(\mathbf{R})$. In particular, $|g|$ must be bounded above. Since

$$\sum_n |g(x-na)|^2 \le \left(\sum_n |g(x-na)| \right)^2,$$

the inequality

$$0 < A \le \sum_n |g(x-na)| \quad a.e. \tag{4.16}$$

is also a necessary condition for (g, a, b) to generate a frame for $L^2(\mathbf{R})$. Certainly, if there exists a constant $A > 0$ such that $|g(x)| > A$ a.e. on $[na, (n+1)a]$ for some integer n, then the inequality (4.16) holds. On the other hand, if the series $\sum_n |g(x-na)|$ converges uniformly for almost all $x \in [0, a]$ and $|g|$ is bounded above, then we have the upper bound for $\sum_n |g(x-na)|$. Indeed, for any $M_1 > 0$ there exists a $N > 0$ such that for almost all $x \in [0, a]$,

$$\sum_{|n|>N} |g(x-na)| \le M_1.$$

Since $|g|$ is bounded above and there are only finite terms in the summation $\sum_{|n|\le N} |g(x-na)|$, there exists $M_2 > 0$ such that $esssup_{[0,a]} \sum_{|n|\le N} |g(x-na)| \le M_2$. Note that $\sum_n |g(x-na)|$ is an a-periodic function. Hence,

$$\sum_n |g(x-na)| \le B = M_1 + M_2 \quad a.e.$$

Corollary 4.2.1 *Let $g \in L^2(\mathbf{R})$ and $a > 0$ be such that*

(1) there exists a constant $A > 0$ such that $A \leq \sum_n |g(x-na)|$ a.e. and $|g|$ is bounded above,

(2) the series $\sum_n |g(x-na)|$ converges uniformly for almost all $x \in [0, a]$.

Then there exists $0 < b_0 \leq 1/a$ such that (g, a, b_0) generates a W-H frame for $L^2(\mathbf{R})$.

From the proof of Theorem 4.2.4, we see that there exists an integer $N_1 > 0$ such that (g, a, b_0) with $b_0 = 1/(2N_1 a)$ generates a W-H frame for $L^2(\mathbf{R})$. Following the lines of the proof, we have that for any integer $N > 2N_1$, (g, a, b) with $b = 1/(Na)$ also generates a W-H frame for $L^2(\mathbf{R})$. The technique used for the proof of Theorem 4.2.4 leads to the following corollary.

Corollary 4.2.2 *Let $g \in L^2(\mathbf{R})$ satisfy the conditions of Corollary 4.2.1 for some $a > 0$. Then for any fixed integer $q > 0$ there exists $0 < b_0 \leq 1/a$ such that (g, a, b) generates a W-H frame for $L^2(\mathbf{R})$ for all $0 < b = q/(pa) \leq b_0$, where $p > 0$ is an integer.*

Proof. We only need to show that for any $\tilde{A} < A$ there exists a constant $0 < b_0 \leq 1/a$ such that for all $0 < b = q/(pa) \leq b_0$,

$$\sum_{k \neq 0} \sum_n |g(x-na)||g(x-na-k/b)| \leq \tilde{A} \quad a.e., \tag{4.17}$$

where A is the lower bound in (4.12). Let $\varepsilon_0 = \tilde{A}/(2qB)$ and let the integer $N_1 > 0$ be so large that

$$\sum_{|n| \geq N_1} |g(x-na)| \leq \varepsilon_0$$

for all $x \in [0, a] \setminus V$. Set $b_0 = 1/((2N_1+1)a)$. If $q/(pa) \leq b_0$ then $p \geq (2N_1+1)q$. For each integer $k \neq 0$, we can write $kp = n_k q + i$ for some integers n_k and $0 \leq i < q$. When $k > 0$, $kp = n_k q + i \geq k(2N_1+1)q \geq (2N_1+1)q$, which implies that $n_k \geq 2N_1 + 1$. Similarly, when $k < 0$, we have $kp = n_k q + i \leq -(2N_1+1)q$, which implies that $n_k \leq -(2N_1+2)$. Hence if $|n| < N_1$, then we have both $|n+n_k| > N_1$ and $|n+n_k-1| \geq N_1$ for all integers $k \neq 0$ and $p \geq (2N_1+1)q$. Now for each $x \in [0,a] \setminus V$, $0 \leq i < q$, we either have $0 \leq x_i = x - ia/q < a$ when $x \geq ia/q$, or $0 \leq x_i = x - ia/q + a \leq a$ when $x < ia/q$. Thus we either have

$$x - na - n_k a - ia/q = x_i - (n+n_k)a$$

or

$$x - na - n_k a - ia/q = x_i - (n+n_k-1)a$$

Note that for some $x \in [0,a] \setminus V$, x_i may be in the set V, but the set containing all such x has measure zero because x is either in the set $ia/q + V$ or in the set $ia/q - a + V$ for $0 \leq i < q$. Thus, for almost all $x \in [0,a]$ and $0 < b = q/(pa) \leq b_0$, we have

$$\begin{aligned}\sum_{k\neq 0} |g(x-na-k/b)| &= \sum_{k\neq 0} |g(x-na-kpa/q)| \\ &= \sum_{i=0}^{q-1} \sum_{k\neq 0} |g(x-na-n_k a - ia/q)| \\ &\leq \sum_{i=0}^{q-1} \sum_{|n|\geq N_1} |g(x_i - ia/q - na)| \leq q\varepsilon_0.\end{aligned}$$

Following the lines of (4.16), we see that the inequality (4.17) holds.

Remark 4.2.1 *It is easy to see that if $g \in W(L^\infty, \ell^1)$ defined in Theorem 4.2.2, then $\sum_n |g(x-na)|$ converges uniformly for almost all $x \in [0,a]$. To compare with Theorem 4.2.2, we construct a function $g \in L^2(\mathbf{R})$ such that $g \notin W(L^\infty, \ell^1)$ but g does satisfies all the conditions of Theorem 4.2.4 for $a = 1$.*

Example 4.2.1 Let

$$g(x) = \begin{cases} 1/n & : \quad |x| \in [n - 1/2^{n-1}, n - 1/2^n] = U_n, n \geq 1 \\ 1/2^n & : \quad |x| \in [n-1, n] - U_n, n \geq 1 \end{cases}$$

It is not hard to prove that $g \in L^2(\mathbf{R})$ and that

$$1 \leq \sum_n |g(x-n)| < 3, \qquad x \in [0,1]. \tag{4.18}$$

For any $\varepsilon > 0$, if we take $N_2 > 0$ such that $1/N_2 + 2\sum_{n \geq N_2} 1/2^n < \varepsilon$, then

$$\sum_{n \geq N_2} |g(x-n)| \leq 1/N_2 + 2 \sum_{n \geq N_2} 1/2^n < \varepsilon, \qquad x \in [0,1]. \tag{4.19}$$

This means that the series $\sum_n |g(x-n)|$ converges uniformly on $[0,1]$. So, g satisfies all the conditions of Theorem 4.2.4 for $a = 1$.

On the other hand, it can be verified that

$$\|g\|_{W(L^\infty, \ell^1)} = 2 \sum_{n=1}^{\infty} \frac{1}{n} = \infty. \tag{4.20}$$

This means that $g \notin W(L^\infty, \ell^1)$.

4.3 The Affine Case

4.3.1 Definitions and Some Known Results

For $d > 0$ let $\mathbf{D}_d$ denote the dilation operator

$$\mathbf{D}_d f(x) = d^{-1/2} f(x/d).$$

The Fourier transform of g is $\hat{g}(\gamma) = \int g(x) e^{-2\pi i \gamma x}\, dx$. We now have the following definitions related to dilations and translations of a function g:

Definition 4.3.1 *Given $g \in L^2(\mathbf{R}), a > 1$ and $b > 0$, we say that (g, a, b) generates a affine frame for $L^2(\mathbf{R})$ if $\{\mathbf{D}_{a^n}\mathbf{T}_{mb}g\}_{m,n \in \mathbf{Z}}$ is a frame for $L^2(\mathbf{R})$. The function g is called the affine mother wavelet. The numbers a and b are the wavelet parameters.*

Definition 4.3.2 *Given $g \in L^2(\mathbf{R}), a > 1$ and $b > 0$, we say that (g, a, b) generates a dual affine frame for $L^2(\mathbf{R})$ if $\{\mathbf{E}_{mba^n}\mathbf{D}_{a^{-n}}g\}_{m,n \in \mathbf{Z}}$ is a frame for $L^2(\mathbf{R})$. The function g is called the dual affine mother wavelet.*

Since

$$(\mathbf{D}_{a^n}\mathbf{T}_{mb}g)\hat{} = \mathbf{E}_{-mba}\mathbf{D}_{a^{-n}}\hat{g},$$

we know that (g, a, b) generates an affine frame for $L^2(\mathbf{R})$ if and only if $(\hat{g}, a, b)$ generates a dual affine frame for $L^2(\mathbf{R})$. Because of this fact, we only need to consider the affine mother wavelet case.

The following conclusion was given by Daubechies [9].

Theorem 4.3.1 *Let $g \in L^2(\mathbf{R}), a > 1$ be such that*

(1) there are constants A and B such that

$$0 < A \leq \sum_n |\hat{g}(a^n\gamma)|^2 \leq B < \infty, \qquad for\, a.e.\ \gamma \in R, \tag{4.21}$$

(2) $lim_{b \to 0} \sum_{k \neq 0} \beta(k/b)^{1/2}\beta(-k/b)^{1/2} = 0$, where

$$\beta(s) = esssup_{|\gamma| \in [1,a]} \sum_n |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - s)|. \tag{4.22}$$

Then there exists $b_0 > 0$ such that (g, a, b) generates an affine frame for $L^2(\mathbf{R})$ for each $0 < b \leq b_0$.

To prove Theorem 4.3.1, one can use the following inequality (see, for example, the proof of Theorem 5.1.6 in [27])

$$\begin{aligned}
&\sum_n \sum_m |\langle f, D_{a^n}T_{mb}g\rangle|^2 \\
&\leq b^{-1}B\|\hat{f}\|_2^2 + b^{-1}\left(\sum_{k\neq 0}\sum_n \int_{\mathbf{R}} |\hat{f}|^2\, |\hat{g}(a^n\gamma)|\, |\hat{g}(a^n\gamma - k/b)|\right)^{1/2} \\
&\qquad \cdot \left(\sum_{k\neq 0}\sum_n \int_{\mathbf{R}} |\hat{f}|^2\, |\hat{g}(a^n\gamma)|\, |\hat{g}(a^n\gamma + k/b)|\right)^{1/2} \\
&= b^{-1}B\|\hat{f}\|_2^2 + b^{-1}\sum_{k\neq 0}\sum_n \int_{\mathbf{R}} |\hat{f}|^2\, |\hat{g}(a^n\gamma)|\, |\hat{g}(a^n\gamma - k/b)|
\end{aligned} \tag{4.23}$$

and the fact that $esssup_{\gamma\in\mathbf{R}} \sum_n |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - s)| = \beta(s)$ because for $\gamma \neq 0$ there exists an integer m such that $\gamma = a^m\xi$ with $|\xi| \in [1, a]$. In analogy to Theorem 4.2.3, a slight generalization of Theorem 4.3.1 can be obtained:

Proposition 4.3.1 *Let $g \in L^2(\mathbf{R}), a > 1$ and $b > 0$ be such that the condition (1) of Theorem 4.3.1 is satisfied. Assume there exists a constant $0 < \tilde{A} < A$ such that*

$$\sum_{k\neq 0}\sum_n |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - k/b)| \leq \tilde{A} \qquad a.e. \tag{4.24}$$

Then (g, a, b) generates an affine frame for $L^2(\mathbf{R})$.

4.3.2 Conditions on Affine Mother Wavelets

Theorem 4.3.2 *Let $g \in L^2(\mathbf{R}, a > 1$ be such that*

1. *there are constants A_1 and B_1 such that*

$$0 < A_1 \leq \sum_n |\hat{g}(a^n\gamma)| \leq B_1 < \infty \qquad a.e.\ \gamma \in R, \tag{4.25}$$

and the series $\sum_n |\hat{g}(a^n\gamma)|$ converges uniformly for almost all $|\gamma| \in [1, a]$,

2. *there is a constant $c > 0$ such that*

$$\sum_n |\hat{g}(x_n - nc)| < \infty$$

for any sequence $\{x_n\} \subset [0, c] \setminus V$ with $|V| = 0$.

Then there exists $b_1 > 0$ such that (g, a, b_1) generates an affine frame for $L^2(\mathbf{R})$.

Proof. Following the proof of Theorem 4.2.4, we see that inequality (4.25) implies that there exist constants $A, B > 0$ such that condition (1) of Theorem 4.3.1 is satisfied. By Proposition 4.3.1, we must show that there exists a constant $b_1 > 0$ such that

$$\sum_{k\neq 0}\sum_n |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - k/b_1)| \leq \tilde{A} \qquad a.e. \tag{4.26}$$

for any constant $0 < \tilde{A} < A$.

It is easy to see from condition 2 of Theorem 4.3.2 that $\hat{g} \in W(L^{\infty}, \ell^1)$. Indeed, for each integer n there exists $x_n \in [0, c] \setminus V$ such that $\|\hat{g}\chi_{[0,c)}\|_{\infty} < |\hat{g}(x_0)| + 1$ for $n = 0$ and such that

$$\|\hat{g}\chi_{[cn,c(n+1))}\|_{\infty} < |\hat{g}(x_n + nc)| + 1/2^{|n|}$$

for $n \neq 0$. Then

$$\sum_n \|\hat{g}\chi_{[cn,c(n+1))}\|_{\infty} < \sum_n |\hat{g}(x_{-n} - nc)| + 3 < \infty.$$

Now we set

$$B_2 = \sum_n \|\hat{g}\chi_{[cn,c(n+1))}\|_{\infty}. \tag{4.27}$$

For any $n \in \mathbf{Z}, |\gamma| \in [1, a]$, we can find uniquely $z(n, \gamma) \in \mathbf{Z}$ and $\beta(n, \gamma) \in [0, c)$ such that

$$a^n\gamma = z(n, \gamma)c + \beta(n, \gamma).$$

Define $\Delta = [1, a] \cup [-a, -1]$. Let W be the subset of Δ with measure $|W| = 0$ such that the series $\sum_n |\hat{g}(a^n\gamma)|$ converges uniformly on $\Delta \setminus W$. Define W_n as the set of all elements $\gamma \in \Delta$ such that $\beta(n, \gamma) \in V$, then $|W_n| = 0$ for all $n \in \mathbf{Z}$. If we set $U = W \cup (\cup_n W_n)$, then $U \subset \Delta$ and $|U| = 0$. It follows that for any $\gamma \in \Delta \setminus U$, we have $\beta(n, \gamma) \in [0, c] \setminus V$ for all $n \in \mathbf{Z}$.

For $\varepsilon_1 = \tilde{A}/(B_1 + B_2)$ with $0 < \tilde{A} < A$, it follows from condition 1 of Theorem 4.3.2 that there is an integer $N_1 > 0$ such that $\sum_{|n| \geq N_1} |\hat{g}(a^n\gamma)| < \varepsilon_1$ for all $\gamma \in \Delta \setminus U$. By the definition of $z(n, \gamma)$ and (4.27), we know there exists a sufficiently large integer $N_2 > 0$ such that $\sum_{|n| > N_2} |\hat{g}(x_n - nc)| < \varepsilon_1$ for any sequence $\{x_n\} \subset [0, c] \setminus V$ and such that $|z(n, \gamma)| < N_2$ for any $|n| < N_1$, $\gamma \in [1, a]$.

If we take $b_1 = 1/2N_2c$, then for any $|n| < N_1$ and $\gamma \in \Delta \setminus U$ we have

$$\sum_{k \neq 0} |\hat{g}(a^n\gamma - k/b_1)|$$

$$\begin{aligned} &= \sum_{k\neq 0} |\hat{g}(\beta(n,\gamma) + z(n,\gamma)c - 2kN_2c)| \\ &\leq \sum_{|m|>N_2} |\hat{g}(\beta(n,\gamma) - mc)| \\ &< \varepsilon_1. \end{aligned} \tag{4.28}$$

Thus, for any $\gamma \in \Delta \setminus U$, it follows from (4.25), (4.27), and (4.28) that

$$\begin{aligned} &\sum_{k\neq 0}\sum_{n} |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - k/b_1)| \\ &\quad = \sum_{k\neq 0}\sum_{|n|\geq N_1} |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - k/b_1)| \\ &\qquad\quad + \sum_{k\neq 0}\sum_{|n|<N_1} |\hat{g}(a^n\gamma)||\hat{g}(a^n\gamma - k/b_1)| \\ &\quad \leq \sum_{|n|\geq N_1} |\hat{g}(a^n\gamma)| \left(\sum_{k\neq 0} |\hat{g}(a^n\gamma - k/b_1)|\right) \\ &\qquad\quad + \sum_{|n|<N_1} |\hat{g}(a^n\gamma)| \left(\sum_{k\neq 0} |\hat{g}(a^n\gamma - k/b_1)|\right) \\ &\quad < B_2\varepsilon_1 + B_1\varepsilon_1 = \bar{A}. \end{aligned}$$

The theorem is proved by Proposition 4.3.1.

With condition 2 of Theorem 4.3.2 and $|g(\gamma)| \leq C|\gamma|^\alpha$ for $|\gamma| \leq \delta < 1$ and some constants $C, \alpha > 0$, we see that the series $\sum_n |\hat{g}(a^n\gamma)|$ converges uniformly for almost all $|\gamma| \in [1, a]$. First, $z(n,\gamma) \geq z(n,1)$ for $\gamma \in [1,a]$ because

$$\begin{aligned} a^n\gamma &= (z(n,1)c + \beta(n.1))\gamma \\ &= z(n,1)c + (\gamma - 1)z(n,1)c + \beta(n,1)\gamma \\ &\geq z(n,1)c. \end{aligned}$$

Second, there exist positive integers M_0 and n_0 such that $a > 1 + 1/M_0$ and $z(n_0, 1) \geq M_0$. So, we have $z(n_0, \gamma) \geq M_0$, which gives the inequality $z(n+1,\gamma) \geq z(n,\gamma) + 1$ for $n \geq n_0$ because

$$a^{n+1}\gamma = a\,(z(n,\gamma)c + \beta(n,\gamma))$$

$$
\begin{aligned}
&\geq az(n,\gamma)c > (1+1/M_0)z(n,\gamma)c \\
&= (z(n,\gamma)+z(n,\gamma)/M_0)c \\
&\geq (z(n,\gamma)+1)c.
\end{aligned}
$$

The similar arguments yield $z(n+1,\gamma) \leq z(n,\gamma)-1$ for $\gamma \in [-a,-1]$ and $n \geq n_0$. Thus, we conclude that for $|\gamma| \in [1,a]$ and all integers $n \geq n_0$, the intervals $[z(n,\gamma)c,(z(n,\gamma)+1)c)$ are mutually disjoint and that

$$
\begin{aligned}
\sum_{n\geq n_0} |\hat{g}(a^n\gamma)| &= \sum_{n\geq n_0} |\hat{g}(z(n,\gamma)c+\beta(n,\gamma))| \\
&\leq \sum_{n\geq M_0} \left\|\hat{g}\chi_{[nc,(n+1)c)}\right\|_\infty \quad a.e.,
\end{aligned}
$$

which implies that the series $\sum_{n\geq 0} |\hat{g}(a^n\gamma)|$ converges uniformly for almost all $|\gamma| \in [1,a]$.

For the serie: $\sum_{n<0} |\hat{g}(a^n\gamma)|$ we let n_1 be a negative integer such that $|a^n\gamma| \leq |a^{n+1}| \leq \delta$ for $n \leq n_1$ and $|\gamma| \in [1,a]$. The condition that $|g(\gamma)| \leq C|\gamma|^\alpha$ for $|\gamma| \leq \delta < 1$ and some $\alpha > 0$ implies the uniform convergence of $\sum_{n<0} |\hat{g}(a^n\gamma)|$ for almost all $|\gamma| \in [1,a]$:

$$
\sum_{n<n_1} |\hat{g}(a^n\gamma)| \leq C \sum_{n<n_1} |a^n\gamma|^\alpha \leq C \sum_{n<n_1} a^{(n+1)\alpha}.
$$

Therefore, the series $\sum_n |\hat{g}(a^n\gamma)|$ converges uniformly for almost all $\gamma \in [1,a]$.

4.4 The Semi-Irregular W-H Frames

The main condition we have provided in this dissertation is the uniform convergence of the series $\sum_n |g(x-na)|$ that is easy to verify. In this section, we will apply the techniques similar to the ones used in previous sections to derive some conditions on $g \in L^2(\mathbf{R})$ under which the set $\{E_{mb}T_{a_n}g\}_{m,n\in Z}$ or $\{E_{a_n}T_{mb}g\}_{m,n\in \mathbf{Z}}$ with an arbitrary real sequence $\{a_n\}_{n\in\mathbf{Z}}$ forms a W-H Frame and will make few remarks on the necessary conditions on W-H Frames. Since $(E_bT_ag) = T_bE_{-a}\hat{g}$ and the Fourier transform is a unitary map of $L^2(R)$ onto $L^2(R)$, it follows that $\{E_{a_n}T_{mb}g\}_{m,n\in\mathbf{Z}}$ is a W-H frame for $L^2(R)$ if and only if $\{T_{a_n}E_{-mb}\hat{g}\}_{m,n\in\mathbf{Z}}$ is a frame for $L^2(R)$. Thus, we consider only the set $\{E_{mb}T_{a_n}g\}_{m,n\in Z}$. Here, we consider irregular translates of g and the set $\{E_{mb}T_{a_n}g\}_{m,n\in Z}$ is a "semi-irregular" W-H Frame. The irregular W-H frames are useful for irregular sampling and time-frequency analysis.

Let $b > 0$ and let f be continuous and compactly supported. For fixed n, consider the $1/b-$periodic function given by

$$F_n(t) = \sum_k f(t - k/b)\overline{g(t - a_n - k/b)}.$$

Following the lines of the proof of Theorem 4.1.5 in [27], we have

$$\begin{aligned}&\sum_n\sum_m |\langle f, E_{mb}T_{a_n}g\rangle|^2\\ &= b^{-1}\int_R |f(t)|^2 \cdot \sum_n |g(t-a_n)|\ ^2dt\\ &\quad +b^{-1}\sum_{k\neq 0}\int_R \overline{f(t)}f(t-k/b)\cdot\sum_n g(t-a_n)\overline{g(t-a_n-k/b)}\,dt.\end{aligned} \tag{4.29}$$

In analogy to Theorem 4.2.3 , we have

Proposition 4.4.1 *Let $g \in L^2(\mathbf{R})$. $b > 0$ be such that*

1. *there are constants $A, B > 0$ such that*

$$A \leq \sum_n |g(x - a_n)|\ ^2 \leq B \tag{4.30}$$

for a real sequence $\{a_n\}_{n\in\mathbf{Z}}$.

2.

$$\left|\sum_{k\neq 0}\sum_n g(x-a_n)\overline{g(x-a_n-k/b)}\right| \leq \bar{A} < A \qquad a.e. \tag{4.31}$$

Then $\{E_{mb}T_{a_n}g\}_{m,n\in Z}$ forms a frame for $L^2(R)$.

Definition 4.4.1 *A sequence of real numbers $\{a_n\}_{n\in\mathbf{Z}}$ is called uniformly discrete if $\delta = \min_{n\neq m}|a_n - a_m| > 0$. The number δ is the separation constant.*

Theorem 4.4.1 *Let $\{a_n\}_{n\in\mathbf{Z}}$ be a uniformly discrete real sequence with a separation constant δ. If $g \in W(L^\infty, \ell^1)$ satisfies $A \le \sum_n |g(t-a_n)|^2$ for some constant $A > 0$, then there exists $b_0 > 0$ such that $\{E_{mb}T_{a_n}g\}_{m,n\in Z}$ forms a frame for $L^2(\mathbf{R})$ for all $0 < b \le b_0$.*

Proof. Since $g \in W(L^\infty, \ell^1)$, $\sum_n \left\|g \cdot \chi_{[cn,c(n+1))}\right\|_\infty$ is finite for all $c > 0$. We refer to Proposition 4.1.7 in [27] for the proof. We choose $c \le \delta/2$ and show that $\sum_n |g(x-a_n)| \le \sum_n \left\|g \cdot \chi_{[cn,c(n+1))}\right\|_\infty$ for almost all $x \in \mathbf{R}$. Without loss of generality we assume $a_n < a_{n+1}$.

For any $x \in \mathbf{R}$, we can find uniquely $m_x, z(x,n) \in \mathbf{Z}$ and $\beta_x, \alpha(x,n) \in [0,c)$ such that

$$x = m_x c + \beta_x, \quad \beta_x - a_n = z(x,n)c + \alpha(x,n).$$

Then

$$\begin{aligned} |z(x,n)c - z(x,n+1)c| &= |a_{n+1} - a_n + \alpha(x,n+1) - \alpha(x,n)| \\ &\ge |a_{n+1} - a_n| - |\alpha(x,n+1) - \alpha(x,n)| \\ &\ge \delta - c \ge c. \end{aligned}$$

Hence $x - a_n = (m_x + z(x,n))c + \alpha(x,n)$ lie in different intervals $[(m_x+z(x,n))c, (m_x + z(x,n)+1)c)$ for all n and

$$\begin{aligned} \sum_n |g(x-a_n)| &= \sum_n |g((m_x + z(x,n))c + \alpha(x,n))| \\ &\le \sum_n \left\|g \cdot \chi_{[cn,c(n+1))}\right\|_\infty \qquad a.e. \end{aligned}$$

Now let $\sum_n \left\|g \cdot \chi_{[cn,c(n+1))}\right\|_\infty = B_1$. It is obvious that $\sum_n |g(t-a_n)|^2 \le B$ with $B = B_1^2$, which means that condition 1 of Proposition 4.4.1 is satisfied. Next we show that condition 2 of Proposition 4.4.1 also holds. For $0 < \tilde{A} < A$, we let $\varepsilon_0 = \tilde{A}/(2B_1)$ as before. Choose the integer $N_1 > 0$ so large that

$$\sum_{|n| \geq N_1} \left\| g \cdot \chi_{[cn, c(n+1))} \right\|_\infty \leq \varepsilon_0 \tag{4.32}$$

and define $b_0 = 1/((2N_1 + 1)c)$. For $0 < b \leq b_0$ and an integer $k \neq 0$, we can set $b = 1/((2N_1 + 1)c + \theta)$ with $\theta \geq 0$. It is easy to see that for fixed $x \in \mathbf{R}$,

$$\begin{aligned} x - a_n - k/b &= (m_x + z(x,n))c + \alpha(x,n) - 2kN_1c - k(2c + \theta) \\ &= (m_x + z(x,n)) - 2kN_1c - j(n,k)c + \lambda(n,k), \end{aligned}$$

where $j(n,k) \in \mathbf{Z}$ has the same sign as k and $\lambda(n,k) \in [0,c)$ and both $j(n,k)$ and $\lambda(n,k)$ depend on $\alpha(x,n), \theta, c,$ and k. Thus, for any $|m_x + z(x,n)| < N_1, k \neq 0$, $|m_x + z(x,n) - 2kN_1 - j(n,k)| \geq N_1$ holds. It follows that for almost all $x \in \mathbf{R}$ and for a_n with $|m_x + z(x,n)| < N_1$, we have

$$\begin{aligned} \sum_{k \neq 0} |g(x - a_n - k/b)| &= \sum_{k \neq 0} |g((m_x + z(x,n))c - 2kN_1c - j(n,k)c + \lambda(n,k))| \\ &\leq \sum_{|n| \geq N_1} \left\| g \cdot \chi_{[cn, c(n+1))} \right\|_\infty \leq \varepsilon_0. \end{aligned}$$

Therefore,

$$\begin{aligned} &\sum_{k \neq 0} \sum_n |g(x - a_n)||g(x - a_n - k/b)| \\ &= \sum_{k \neq 0} \sum_{|m_x + z(x,n)| \geq N_1} |g(x - a_n)||g(x - a_n - k/b)| \\ &\qquad + \sum_{k \neq 0} \sum_{|m_x + z(x,n)| < N_1} |g(x - a_n)||g(x - a_n - k/b)| \\ &\leq B_1\varepsilon_0 + B_1\varepsilon_0 = 2B_1\varepsilon_0 = \tilde{A} \qquad a.e. \end{aligned}$$

The proof is complete.

CHAPTER 5

DEVELOPMENT AND IMPLEMENTATION OF A PARALLEL 2-D FORWARD FWT ALGORITHM

5.1 Wavelets Basics

Although wavelet theory has its roots in pure mathematics, we are not going to start with detailed mathematical description. Interested readers may want to check other references [7, 10, 40]. This section briefly introduces basic wavelet concepts and explains how wavelets work.

5.1.1 Continuous Wavelet Transform

As mentioned above, a wavelet expansion makes use of translations and dilations of a mother wavelet $\psi \in L^2(\mathbf{R})$. If the translations and dilations parameters vary continuously, the wavelet transform is said to be continuous wavelet transform. In other words, the continuous wavelet transform uses the basis functions

$$\psi_{a,b}(x) = a^{-1/2}\psi(\frac{x-b}{a}) \quad a > 0, b \in \mathbf{R}. \tag{5.1}$$

The *continuous wavelet transform (CWT)* of a function $f \in L^2(\mathbf{R})$ is then defined by

$$\begin{aligned} CWT_f(a,b) &= \int_{\mathbf{R}} f(x)\psi_{a,b}(x)dx \\ &= a^{-1/2}\int_{\mathbf{R}} f(x)\psi(\frac{x-b}{a})dx. \end{aligned} \tag{5.2}$$

where $\{\psi_{a,b}(x), a > 0, b \in \mathbf{R}\}$ are the family of "little waves" which are generally time-localized, i.e., each of them is zero outside the supporting interval. The continuous wavelet transform $CWT_f(a,b)$ measures the content of the signal f at time locale b and scale a. In other words, the continuous wavelet transform $CWT_f(a,b)$ reveals details of where and at what scales lives the information of the signal f.

Most importantly, under the *admissibility condition*

$$C_\psi = \int_{\mathbf{R}} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < \infty, \tag{5.3}$$

the original signal f can be reconstructed from its continuous wavelet transform $CWT_f(a.b)$ by the *inverse continuous wavelet transform*:

$$f(x) = \frac{1}{C_\psi} \int_{\mathbf{R}} \int_{\mathbf{R}} CWT_f(a,b)\psi_{a,b}(x) \frac{dadb}{a^2}. \tag{5.4}$$

5.1.2 Discretized Wavelet Transform

In the case of the continuous wavelet transform, the basis of wavelets is an uncountable set of functions because the parameters a and b vary continuously in the range of $\mathbf{R}^+$ and $\mathbf{R}$, respectively. Therefore, continuous wavelet transforms are not suitable for applications because it is hard to use computers to handle an uncountable set of functions and the basis of uncountable many functions is highly redundant. In practice, it is preferable to follow the basic idea of wavelet analysis and to write the signal f as a discrete superposition. To accomplish this goal, the parameters a and b can be discretized,

$$a = a_0^m. b = nb_0a_0^m. m, n \in \mathbf{Z}, a_0 > 0, b_0 > 0. \tag{5.5}$$

With this discretization, the corresponding wavelet decomposition (5.4) becames

$$f(x) = \sum_{m,n} c_{m,n}(f)\psi_{m,n}(x) \tag{5.6}$$

where

$$\psi_{m,n}(x) = \psi_{a_0^m, nb_0 a_0^m}(x) = a_0^{-m/2}\psi(a_0^{-m}t - nb_0). \tag{5.7}$$

and

$$c_{m,n} = \int_{\mathbf{R}} f(x)\psi_{m,n}(x)dx. \tag{5.8}$$

In this case, the recovery of $f(x)$ from the wavelet coefficients $\{c_{m,n}\}_{m,n\in\mathbf{Z}}$ is possible when $\{\psi_{m,n}(x)\}_{m,n\in\mathbf{Z}}$ form a frame for $L^2(\mathbf{R})$, i.e., there exist constants $A > 0, B < \infty$ so that, for all $f \in L^2(\mathbf{R})$,

$$A\|f\|^2 \leq \sum_{m,n} |c_{m,n}|^2 \leq B\|f\|^2. \tag{5.9}$$

This issue has been discussed in detail in [6, 7, 13, 23].

5.1.3 Multiresolution

Multiresolution representation of functions (signals) is well suited to many applications because it is often practical to process information in successive approximations. In image processing, for instance, an image can be compressed in such a way that different resolution information is grouped into different layers. According to applications and purposes, various amounts of detailed layers are added to the coarse layer. Multiresolution analysis is also a useful tool for a constructive description of wavelets.

Starting with the space $L^2(\mathbf{R})$, the *multiresolution analysis (MRA)* is an increasing sequence of closed subspaces $\{V_j\}_{j\in\mathbf{Z}}$ which successively approximate $L^2(\mathbf{R})$:

1. $V_j \subset V_{j+1}$.

2. $v(x) \in V_j \Leftrightarrow v(2x) \in V_{j+1}$.

3. $v(x) \in V_0 \Leftrightarrow v(x - n) \in V_0$ for all $n \in \mathbf{Z}$.

4. $\bigcup_{j\in\mathbf{Z}} V_j = L^2(\mathbf{R})$ and $\bigcap_{j\in\mathbf{Z}} V_j = \{0\}$.

5. $\phi \in V_0$ and the collection of functions $\{\phi_{0,n} = \phi(x-n) : n \in \mathbf{Z}\}$ forms an orthonormal basis of V_0, where ϕ is the scaling function.

It is immediate that the collection of functions $\{\phi_{m,n} = 2^{m/2}\phi(2^m x - n) : n \in \mathbf{Z}\}$ constitutes a orthonormal basis of V_m. For any $f \in L^2(\mathbf{R})$, if we use f_m to denote the orthonormal projection of f in V_m, i.e.,

$$f_m(x) = \sum_{n \in \mathbf{Z}} < \phi_{m,n}, f > \phi_{m,n}(x), \tag{5.10}$$

then the f can be successively approximated by f_m:

$$f(x) = lim_{m \to \infty} f_m(x) \tag{5.11}$$

Now we explain how the wavelets relate the multiresolution analysis. Because $V_0 \subset V_1$, any function in V_0 can be expressed as a linear combination of the set of functions $\{\phi_{1,n} = 2^{1/2}\phi(2x-n) : n \in \mathbf{Z}\}$, the basis of V_1:

$$\phi(x) = \sqrt{2}\sum_k h_k \phi(2x-n). \tag{5.12}$$

If we let W_j be the orthonormal complement of V_j in V_{j+1}, i.e., $V_{j+1} = V_j \bigoplus W_j$, and define

$$\psi(x) = \sqrt{2}\sum_k (-1)^k h(-k+1)\phi(2x-k), \tag{5.13}$$

then the set of functions $\{\psi_{1,k} = \sqrt{2}\psi(2x-k) : k \in \mathbf{Z}\}$ is an orthonormal basis of W_1.

Repeatly, the similarity property of MRA gives that the set of functions $\{\psi_{j,k} = 2^{j/2}\psi(2^j x - k) : k \in \mathbf{Z}\}$ forms a orthonormal basis of W_j. Since

$$\bigoplus_{j \in \mathbf{Z}} W_j = \bigcup_{j \in \mathbf{Z}} V_j = L^2(\mathbf{R}), \tag{5.14}$$

the section of functions $\{\psi_{j,k} = 2^{j/2}\psi(2^j x - k) : j, k \in \mathbf{Z}\}$ forms a orthonormal basis of $L^2(\mathbf{R})$. The function ψ is the mother wavelet, and the set $\{\psi_{j,k} = 2^{j/2}\psi(2^j x - k) :$

$j, k \in \mathbf{Z}\}$ is the wavelet (little wave) family. Having these wavelets, we can decompose any function (signal) $f \in L^2(\mathbf{R})$ as a linear combination of the orthonormal projection of f on $W_j, j \in \mathbf{Z}$, i.e.,

$$f(x) = \sum_j g_j(x), \tag{5.15}$$

with

$$g_j(x) = \sum_{k \in \mathbf{Z}} < \phi_{j,k}, f > \phi_{j,k}(x). \tag{5.16}$$

If we stop at a certain scale m, then the signal $f(x)$ can be decomposed into a coarse part $f_m(x)$ and some detailed parts $g_j(x) \in W_j, j > m$, i.e.,

$$\begin{aligned} f(x) &= f_m(x) + \sum_{j=m}^{\infty} g_j(x) && (5.17) \\ &= \sum_n c_{m,n}\phi_{m,n}(x) + \sum_{j=m}^{\infty} d_{j,k}\psi_{j,k}(x), && (5.18) \end{aligned}$$

where

$$c_{m,n} = \sum_n < \phi_{m,n}, f >= \int_{\mathbf{R}} f(t)\phi_{m,n}(x)dt, \tag{5.19}$$

$$d_{j,k} = \sum_n < \psi_{j,k}, f >= \int_{\mathbf{R}} f(t)\psi_{j,k}(x)dt. \tag{5.20}$$

5.1.4 Discrete Wavelet Transforms

In [9], Daubechies made a significant contribution to the wavelet theory. She presented a scheme to construct wavelets of compact support from certain mother scaling function ϕ which, in turn, is determined by a finite set of scaling constants $\{h_k : k = 0, 1, ..., M-1$, where M is even $\}$ by the dilation equation

$$\phi(x) = \sum_{k=0}^{M-1} h_k\phi(2x-k). \tag{5.21}$$

and the constant coefficients $\{h_k : k = 0, 1, ..., M-1\}$ satisfy the conditions

$$\sum_k h_{2k} = \sum_k h_{2k-1} = 1 \tag{5.22}$$

$$\sum_k h_k h_{k+2q} = 2\delta_{q,0}, \tag{5.23}$$

where

$$\delta_{q,0} = \begin{cases} 1 & : \quad q = 0 \\ 0 & : \quad q \neq 0 \end{cases}$$

In this case, any $f \in L^2(\mathbf{R})$ can be recovered via the inverse transform

$$f(x) = \sum_{k \in \mathbf{Z}} c_k \phi(x-k) + \sum_{m \geq 0} \sum_{k \in \mathbf{Z}} d_{m,n} \psi(2^m x - n), \tag{5.24}$$

where

$$c_k = \int_{\mathbf{R}} f(t)\phi(t-k)dt, \tag{5.25}$$

$$d_{m,n} = 2^m \int_{\mathbf{R}} f(t)\phi(2^m t - n)dt. \tag{5.26}$$

Daubechies' construction of wavelets from a set of scaling constants plays an important role in discrete wavelet transforms. A *discrete wavelet transform (DWT)* is to apply wavelet theory to discrete data (discrete signals). Given a discrete signal

$$S = \{c_0, c_1, \ldots, c_{N-1}\}, \tag{5.27}$$

we construct a continuous signal

$$f(t) = \sum_{j=0}^{N-1} c_j \phi(t-j). \tag{5.28}$$

As contrasted with (5.18), the signal f is composed of only the s-terms, and the d-coefficients are all 0. Multiresolution analysis can be employed to cast the signal f as successively scaled superpositions, and more d terms will be involved in the wavelet expression of f.

For $0 \leq k < M$, $\phi(t-k) \in V_0$, so it can be written as

$$\phi(t-k) = \sum_{k \in \mathbf{Z}} c_{i,k}\phi(t/2 - i) + \sum_m d_{m,k}\psi(t/2 - m), \tag{5.29}$$

where

$$c_{i,k} = \frac{1}{2}h_{k-2i}; \; d_{m,k} = \frac{1}{2}(-1)^k h_{M-1-k+2m}. \tag{5.30}$$

If we insert (5.29) into (5.28), then we get a wavelet representation of f that has some d terms and fewer s terms. This procedure can be applied repeatedly to produce a practical wavelet representation of the form

$$F = c_{0,0}\phi + \sum_{j,k} d_{j,k}\psi_{j,k}. \tag{5.31}$$

And, the result of the discrete wavelet transform is the following vector of wavelet coefficients:

$$DWTS = \{c_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}, d_{2,0} ..., d_{2,3}, ..., d_{N-1,0}, ..., d_{N-1,2^{N-1}-1}\}. \tag{5.32}$$

5.1.5 How Do Wavelets Work?

In this section, we use the oldest and simplest Haar wavelet family to illustrate how wavelets work. The Haar (mother) wavelet $h(x)$ is a step function taking values 1 and -1 on the intervals $[0, 1/2)$ and $[1/2, 1)$, respectively:

$$h_0(x) = \begin{cases} 1 & : \quad 0 \leq x < 1/2 \\ -1 & : \quad 1/2 \leq x < 1 \\ 0 & : \quad otherwise \end{cases}$$

The set of dilations and translations of $h(x)$, $\{h_{j,k}(x) = 2^{j/2}h(2^j x - k), j, k \in \mathbf{Z}\}$, forms an orthonormal basis of $L^2(\mathbf{R})$. The scaling function for the Haar wavelet basis is simply a constant function with value 1 on the interval $[0, 1)$, i.e.,

$$\phi(x) = \chi(0 \leq x < 1), \tag{5.33}$$

where $\chi(I)$ is the character function of the interval I, i.e., $\chi(x)$ is 1 for all $0 \leq x < 1$, and 0 for all other x.

Let $\check{Y} = (y_0, y_1, ..., y_{2^n-1})$ be the data vector of size 2^n. The data vector can be associated with a piece-wise constant function F on $[0, 1]$:

$$F(x) = \sum_{k=0}^{2^n-1} y_k \cdot \chi(k2^{-n} \leq x < (k+1)2^{-n}). \tag{5.34}$$

The (data) function F is in $L^2([0,1]$, and it has the wavelet decomposition

$$F(x) = c_{0,0}\phi(x) + \sum_{j=0}^{n-1}\sum_{k=0}^{2^n-1} d_{j,k}h(x). \tag{5.35}$$

The sum with respect to j is finite because F is finite step function. For each level, the sum with respect to k is finite because the domain of f is finite.

Let us assume we have data vector $\check{Y} = (1, 0, -3, 2, 1, 0, 1, 2)$. Figure 5.1 is the graph of the corresponding function F.

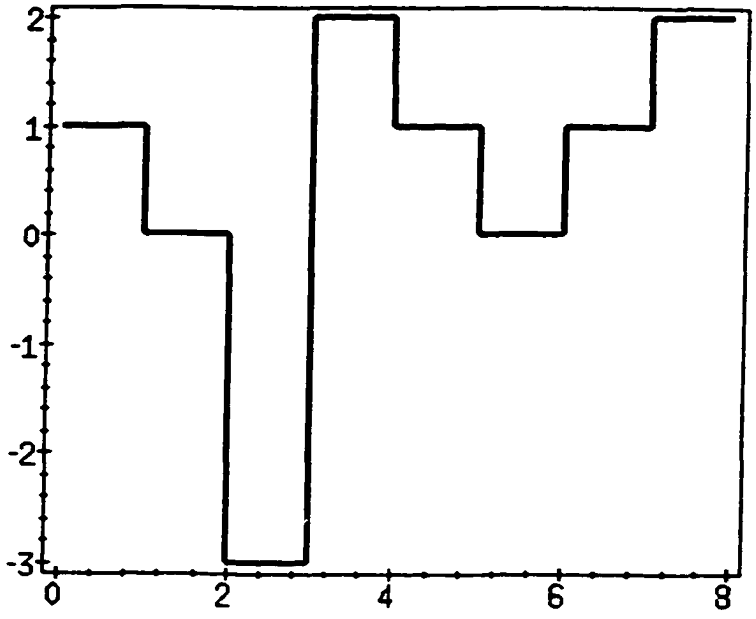

Figure 5.1 : the "data function" F on [0,1)

To approximate F using of the Haar wavelets, eight wavelets from the Haar wavelet family are needed. They are functions $\phi, h_{0,0}, h_{1,0}, h_{1,1}, h_{2,0}, h_{2,1}, h_{2,2}, h_{2,3}$ which are depicted in Figure 5.2. Among the eight Haar wavelets,

In Figure 5.2, there are four functions that code the highest resolution detail, two functions that code the coarser resolution detail, one function that codes the even coarser detail, and another function that codes the "average function level."

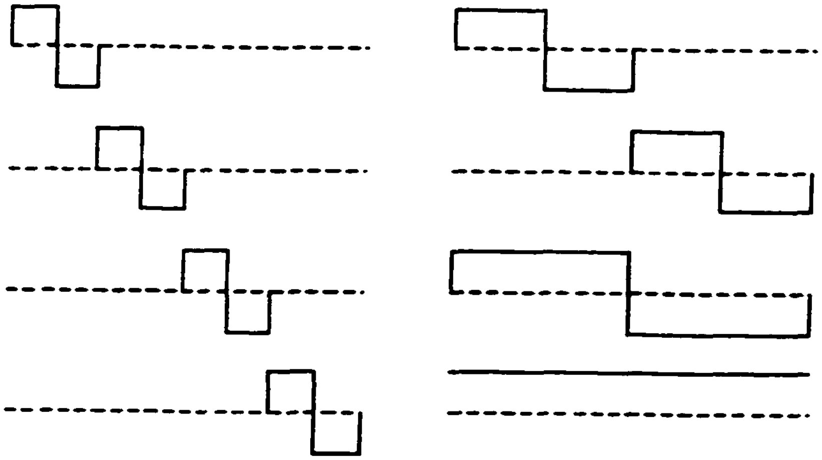

Figure 5.2 : Functions from the Haar wavelet family

When calculating the wavelet transform, the above function groups are weighted by 2, $\sqrt{2}$, 1, and 1.

To express initial function F by use of the Haar wavelets, we have to solve the following system of linear equations:

$$\begin{bmatrix} 1 \\ 0 \\ -3 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{2} & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & \sqrt{2} & 0 & -2 & 0 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & -\sqrt{2} & 0 & 0 & -2 & 0 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & 2 & 0 \\ 1 & -1 & 0 & \sqrt{2} & 0 & 0 & -2 & 0 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & 2 \\ 1 & -1 & 0 & -\sqrt{2} & 0 & 0 & 0 & -2 \end{bmatrix} \cdot \begin{bmatrix} c_{0,0} \\ d_{0,0} \\ d_{1,0} \\ d_{1,1} \\ d_{2,0} \\ d_{2,1} \\ d_{2,2} \\ d_{2,3} \end{bmatrix}$$

The vector on the left side is the initial function "F"; the matrix displayed in the middle contains the weights, and on the right is the resulting wavelet vector. Thus, the equation to be solved is

$$\text{Function} = \text{Weight\_Matrix} * \text{Wavelet\_Vector} .$$

The solution is

$$\begin{bmatrix} c_{0,0} \\ d_{0,0} \\ d_{1,0} \\ d_{1,1} \\ d_{2,0} \\ d_{2,1} \\ d_{2,2} \\ d_{2,3} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2\sqrt{2}} \\ -\frac{1}{2\sqrt{2}} \\ \frac{1}{4} \\ -\frac{5}{4} \\ \frac{1}{4} \\ -\frac{1}{4} \end{bmatrix}$$

Thus, F has the decomposition

$$F = \frac{1}{2}\phi - \frac{1}{2}h_{0,0} + \frac{1}{2\sqrt{2}}h_{1,0} - \frac{1}{2}h_{1,1} + \frac{1}{4}h_{2,0} - \frac{5}{4}h_{2,1} + \frac{1}{4}h_{2,2} - \frac{1}{4}h_{2,3} \qquad (5.36)$$

The first value, $c_{0,0}$, represents the wavelet factor for the basic "grey level" of the image. The second value, $d_{0,0}$, represents the coarsest detail. The third and fourth values, $d_{1,0}$ and $d_{1,1}$, stand for the "middle resolution detail." The last four values contain the wavelet factors for the four "highest resolution functions."

5.2 Design and Analysis of Parallel 2-D Forward FWT Algorithms

In this section, the design of a two-dimensional (2-D) parallel forward FWT algorithm is presented, the communication complexity and computation time are analyzed, and a formula is given to indicate the total execution complexity in terms of the problem size n and the number of processors p.

In Section 2.1, the 1-D FWTs (forward 1-D FWT and inverse 1-D FWT) are introduced, and the 2-D FWT algorithm is studied in Section 2.2.

5.2.1 1-D FWT

1-D Forward FWT

A 2-D forward FWT is actually a combination of one-dimensional (1-D) forward FWTs on both rows and columns of the 2-D input data array. Therefore, it is necessary to introduce the 1-D forward FWT first.

As described in Section 4.1, an n-point discrete wavelet transform can be modeled mathematically as a linear system. It can be defined as a linear transform W_n from S to $DWTS$, the discrete wavelet transform of S:

$$DWTS = W_n \cdot S, \tag{5.37}$$

where W_n is an $n \times n$ matrix.

Because the transform matrix W_n is not sparse in general, the complexity of solving the linear system is $O(n^2)$. However, the transform matrix W_n can be decomposed into the following form:

$$W_n = T_2 T_4 ... T_{n/2} T_n, \tag{5.38}$$

where the $T_i (i = 2, 4, ..., n)$ are sparse matrices of some special properties determined by the original discrete wavelet bases which, in turn, are determined by a finite set of constants $h_k (0 \leq k < M)$ with a fixed M [4, 10].

If we take $n = 8$, then the forward DWT can be implemented as follows:

$$S = \quad S_1 \xrightarrow[T_8]{} S_2 \xrightarrow[T_4]{} S_3 \xrightarrow[T_2]{} S_4 \quad .$$

More precisely,

$$S = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} \xrightarrow[T_8]{} \begin{bmatrix} c \\ c \\ c \\ c \\ d \\ d \\ d \\ d \end{bmatrix} \xrightarrow[T_4]{} \begin{bmatrix} c \\ c \\ d \\ d \\ d \\ d \\ d \\ d \end{bmatrix} \xrightarrow[T_2]{} \begin{bmatrix} c \\ d \\ d \\ d \\ d \\ d \\ d \\ d \end{bmatrix}$$

where c's are used to denote the coarse coefficients and d's are used to denote the detail coefficients.

If we choose the famous Daubechies order-4 ($M = 4$, see Figure 1.1) wavelet for which the scaling constants are

$$\{h_k\} = \{\frac{1+\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, \frac{3-\sqrt{3}}{4}, \frac{1-\sqrt{3}}{4}\}, \tag{5.39}$$

then the length-8 forward DWT is associated with the following T-matrices:

$$T_8 = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & -h_2 & h_1 & -h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & 0 & 0 & 0 & 0 & h_3 & -h_2 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ h_2 & h_3 & h_0 & h_1 & 0 & 0 & 0 & 0 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ h_1 & -h_0 & h_3 & -h_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} h_0 + h_2 & h_1 + h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_3 + h_1 & -h_0 - h_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The properties of T-matrices make the fact that once a d term is created, it is never again affected (we study only the normal wavelet transform here; for some applications the detailed terms need to be decomposed further using wavelet packet transforms. We will discuss wavelet packet transforms briefly in Section 4.4). This leads to a $O(n)$ fast algorithm.

We now implement the 1-D forward FWT algorithm using a pseudo language.

```
ONE_D_FFWT (x : the 1-D input array;
                size : the size of the input data;
                d : the number of iterations)
BEGIN
  len = size;
  for i := 0 to d-1 do
  begin
    hlen := len/2;
    for j := 0 to hlen-1 do /*first half */
    begin
      $x[j] := h_0 * x[2j] + h_1 * x[2j+1] + h_2 * x[(2j+2)\%len] + h_3 * x[(2j+3)\%len]$;
    end
    for j := 0 to hlen-1 do /*second half */
    begin
      $x[j+hlen] := h_3 * x[2j] - h_2 * x[2j+1] + h_1 * x[(2j+2)\%len] - h_0 * x[(2j+3)\%len]$;
    end
    len = len/2;
  end
END
```

To compute the wavelet transform for the k-th element, we need information of the elements at positions $2k, 2k+1, 2k+2$, and $2k+3$ (the last two terms may be wrapped around to 0 and 1 if $2k+2$ and $2k+3$ are greater than the size).

1-D Inverse FWT

For completeness, we discuss 1-D inverse FWT briefly. Similar to the 1-D forward DWT, the 1-D inverse DWT can also be defined as a linear transform W_n^{-1} from S to

IDWTS, and the transform matrix W_n^{-1} can be decomposed into the following form

:

$$W_n^{-1} = T_n^{-1} T_{n/2}^{-1} \ldots T_4^{-1} T_2^{-1}, \tag{5.40}$$

If we take $n = 8$, then the inverse FWT (IFWT) can be implemented as follows:

$$Y = \begin{bmatrix} c \\ d \\ d \\ d \\ d \\ d \\ d \\ d \end{bmatrix} \xrightarrow[T_2^{-1}]{} \begin{bmatrix} c \\ c \\ d \\ d \\ d \\ d \\ d \\ d \end{bmatrix} \xrightarrow[T_4^{-1}]{} \begin{bmatrix} c \\ c \\ c \\ c \\ d \\ d \\ d \\ d \end{bmatrix} \xrightarrow[T_8^{-1}]{} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix},$$

where

$$T_2^{-1} == \frac{1}{h_0^2 + h_1^2 + h_2^2 + h_3^2} \begin{bmatrix} h_0 + h_2 & h_1 + h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ h_1 + h_3 & -h_0 - h_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^{-1} == \frac{1}{h_0^2 + h_1^2 + h_2^2 + h_3^2} \begin{bmatrix} h_0 & h_2 & h_3 & h_1 & 0 & 0 & 0 & 0 \\ h_1 & h_3 & -h_2 & -h_0 & 0 & 0 & 0 & 0 \\ h_2 & h_0 & h_1 & h_3 & 0 & 0 & 0 & 0 \\ h_3 & h_1 & -h_0 & -h_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_8^{-1} == \frac{1}{h_0^2 + h_1^2 + h_2^2 + h_3^2} \begin{bmatrix} h_0 & 0 & 0 & h_2 & h_3 & 0 & 0 & h_1 \\ h_1 & 0 & 0 & h_3 & -h_2 & 0 & 0 & -h_0 \\ h_2 & h_0 & 0 & 0 & h_1 & h_3 & 0 & 0 \\ h_3 & h_1 & 0 & 0 & -h_0 & -h_2 & 0 & 0 \\ 0 & h_2 & h_0 & 0 & 0 & h_1 & h_3 & 0 \\ 0 & h_3 & h_1 & 0 & 0 & -h_0 & -h_2 & 0 \\ 0 & 0 & h_2 & h_0 & 0 & 0 & h_1 & h_3 \\ 0 & 0 & h_3 & h_1 & 0 & 0 & -h_0 & -h_2 \end{bmatrix}$$

The following is the 1-D IFWT algorithm written in a pseudo language.

```
ONE_D_IFWT (x : the 1-D input array ;
            size : the size of the input data;
            d : the number of iterations)
BEGIN
  len = size ≫ (d - 1); /* start from smallest length */
  for r := d-1 downto 0 do
```

```
    begin
      hlen := len/2;
      hsum := $h_0^2 + h_1^2 + h_2^2 + h_3^2$
      begin /* first two are special */
        $x[0] := \frac{1}{hsum}(h_0 * x[0] + h_2 * x[hlen-1] + h_3 * x[hlen-1] + h_1 * x[len]);$
        $x[1] := \frac{1}{hsum}(h_1 * x[0] + h_3 * x[hlen-1] - h_2 * x[hlen-1] - h_0 * x[len]);$
      end
      for i := len-1 downto 3, step 2 do
      begin
        $x[i] = \frac{1}{hsum}(h_3 * x[hlen-2] + h_1 * x[hlen-1] - h_0 * x[i-1] - h_2 * x[i]);$
        $x[i-1] = \frac{1}{hsum}(h_2 * x[hlen-2] + h_0 * x[hlen-1] + h_1 * x[i-1] + h_3 * x[i]);$
      end
      len = 2* len;
    end
END
```

5.2.2 A 2-D Parallel FWT Algorithm and Analysis

We assume the input data are organized as an $\sqrt{n} \times \sqrt{n}$ square matrix. The algorithm then checkerboard partitions the 2-D data into $\sqrt{p} \times \sqrt{p}$ processors :

$P_{0,0}$	$P_{0,1}$	...	...	$P_{0,\sqrt{p}}$
$P_{1,0}$	$P_{1,1}$	...	...	$P_{1,\sqrt{p}}$
...	...	...	...	...
$P_{\sqrt{p},0}$	$P_{\sqrt{p},1}$	...	...	$P_{\sqrt{p},\sqrt{p}}$

Figure 5.3 : Partition of 2-D data

Each processor computes $\sqrt{m} \times \sqrt{m} = m$ elements, where

$$m = \frac{n}{p}. \tag{5.41}$$

We make the assumption that $\sqrt{m} \geq 4$ because a very small task for each processor will make the communicational overhead too expensive with respect to the computational time.

5.2.3 The Algorithm

Assume the input data are stored in an array $x[0 : \sqrt{n} - 1][0 : \sqrt{n} - 1]$, and assume $P_{i,j}$ be the processor at the i-th row and j-th column. For $0 \leq i < p$ and $0 \leq j < p$, the processor $P_{i,j}$ keeps two local arrays **xrow**$[0 : \sqrt{n} - 1][0 : \sqrt{m} - 1]$ and xcol$[0 : \sqrt{n} - 1][0 : \sqrt{m} - 1]$ that are used to store temporarily the corresponding rows of the data x from $i * m$-th row to $((i + 1) * m - 1)$-th row and columns of the the data x from $j * m$-th column to $((j + 1) * m - 1)$-th column and keep track of the changes with the process of execution.

Let t_w denote the time required to transfer a unit data for one hop, and let t_c be the time required to perform a multiplication or an addition.

First, each processor sends and receives data row-wise to get the data required to compute the row-wise 1-D FWT. For each i, j, the data in processor $P_{i,j}$ are shown below:

...	...	...	...	...
...	$x[i\sqrt{m}.j\sqrt{m}]$	...	$x[i\sqrt{m},(j+1)\sqrt{m}-1]$	...
...	...	...	...	...
...	$x[(i+1)\sqrt{m}-1,j\sqrt{m}]$	...	$x[(i+1)\sqrt{m}-1,(j+1)\sqrt{m}-1]$	...
...	...	...	...	...

Figure 5.4 : The data in $P_{i,j}$

When $i < \sqrt{p}/2$, the data from position $(i\sqrt{m}, 2j\sqrt{m})$, $(i\sqrt{m}, 2j\sqrt{m}+1)$, $(i\sqrt{m}, 2j\sqrt{m}+2)$, and $(i\sqrt{m}, 2j\sqrt{m}+3)$ which are located in processor $P_{i,2j}$ are needed to compute the 1-D FWT at position $(i\sqrt{m}, j\sqrt{m})$, and the data from position $(i\sqrt{m}, 2(j+1)\sqrt{m}-2)$, $(i\sqrt{m}, 2(j+1)\sqrt{m}-1)$, $(i\sqrt{m}, 2(j+1)\sqrt{m})$, and $(i\sqrt{m}, 2(j+1)\sqrt{m}+1)$ which are located in processor $P_{i,2j+1}$ and $P_{i,2j+2}$ are needed to compute the FWT at position $(i\sqrt{m}, (j+1)\sqrt{m}-1)$. Thus, the data in the top rows of the three processors $P_{i,2j}$, $P_{i,2j+1}$, and $P_{i,2j+2}$ are required to compute the 1-D row-wise 1-D FWT for the top row of elements in processor $P_{i,j}$. Similarly, the data in the second rows of the three processors $P_{i,2j}$, $P_{i,2j+1}$, and $P_{i,2j+2}$ are required to compute the 1-D row-wise FWT for the second row of elements in processor $P_{i,j}$. And so on. Therefore, the data in processors $P_{i,2j}$, $P_{i,2j+1}$, and $P_{i,2j+2}$ need to be sent to processor $P_{i,j}$ to compute the first step 1-D row-wise FWT. In fact, only the first two columns of $P_{i,2j+2}$ are really needed in this step. We will have two implementations; in one of them the exact data needed are sent and received, and in the other more data than necessary are sent and received. It is done in this way because the branch statement overhead may be large if we try to send only part of the data from a PE. Section 4.3 will discuss these two implementations and compare the performance. Based on the above observation, the communication can be scheduled as follows:

```
for k := 2 to $\sqrt{p}-2$, step 2 do in parallel
begin
    $P_{i,k}$ sends to $P_{i,k/2}$
end
for k := 1 to $\sqrt{p}-1$, step 2 do in parallel
begin
    $P_{i,k}$ sends to $P_{i,(k-1)/2}$
```

```
end
for k := 4 to $\sqrt{p} - 2$, step 2 do in parallel
begin
  $P_{i,k}$ sends to $P_{i,(k-2)/2}$
end
```

In the first "for" loop, for each $k \in \{2, 4, ..., \sqrt{p} - 2\}$, m units of data are moved $k/2$ hops, so $kmt_w/2$ time is taken. The maximum of them is $(\sqrt{p} - 2)mt_w/2$. In the second "for" loop, for each $k \in \{1, 3, ..., \sqrt{p} - 1\}$, m units of data are moved $(k+1)/2$ hops, so $(k+1)mt_w/2$ time is taken. The maximum of them is $\sqrt{p}mt_w/2$. In the third "for" loop, for each $k \in \{4, 6, ..., \sqrt{p} - 2\}$, m units of data are moved $(k+2)/2$ hops, so $(k+2)mt_w/2$ time is taken. The maximum of them is $\sqrt{p}mt_w/2$. Together, the time taken for this communication is $(3m\sqrt{p}/2 - m/2)t_w$. Similar analysis can be applied to cases of $i \leq \sqrt{p}$; it follows that the communication time between PEs in each of the rows of the mesh is $3m\sqrt{p}t_w/2 - mt_w$. Because each row of PEs is independent of the others, the communication can be done in parallel. Therefore, the total communication time for step 1 row-wise 1-D FWT is $3m\sqrt{p}t_w/2 - mt_w$.

The computational time for the step 1 row-wise 1-D FWT is $7mt_c$ because each PE calculates $\sqrt{m} \times \sqrt{m} = m$ 1-D FWT coefficients and the calculation of each 1-D FWT coefficient needs four multiplications and 3 additions, and all the PEs do their calculations in parallel.

Thus, the execution time taken for step 1 row-wise 1-D FWT is

$$3m\sqrt{p}t_w/2 - mt_w + 7mt_c. \tag{5.42}$$

Similarly, the execution time taken for step 1 column-wise 1-D FWT is also

$$3m\sqrt{p}t_w/2 - mt_w + 7mt_c. \tag{5.43}$$

Thus, the execution time taken for the first iteration of the 2-D FWT is

$$3m\sqrt{p}t_w - 2mt_w + 14mt_c. \tag{5.44}$$

For the following iterations, the analysis is the same as above except that the size is halved for each next iteration. So, the execution time taken for the second iteration of the 2-D FWT is

$$3m\sqrt{p}t_w/2 - 2mt_w + 14mt_c, \tag{5.45}$$

and so on. The total execution time is derived by adding the terms from each iteration:

$$\begin{aligned}
&T_{para} \\
&= (3m\sqrt{p}t_w - 2mt_w + 14mt_c) + (3m\sqrt{p}t_w/2 - 2mt_w + 14mt_c) \\
&\quad +\ldots + (3mt_w - 2mt_w + 14mt_c) \\
&= 3m\sqrt{p}t_w(1 + 1/2 + 1/4 + \ldots + 1/\sqrt{p}) - 2mlog(p)t_w + 14mlog(p)t_c \\
&< 6m\sqrt{p}t_w - 2mlog(p)t_w + 14mlog(p)t_c \\
&= 6t_w\frac{n}{\sqrt{p}} - 2t_w\frac{nlog(p)}{p} + 14t_c\frac{nlog(p)}{p} \\
&= T_{communication} + T_{computation},
\end{aligned}$$

where

$$T_{communication} = 6\frac{nt_w}{\sqrt{p}} - 2t_w\frac{nlog(p)}{p}, \tag{5.46}$$

and

$$T_{computation} = 14\frac{nlog(p)t_c}{p}. \tag{5.47}$$

Because the best sequential FWT algorithm runs in $O(n)$ time, the speedup is $O(\sqrt{p})$ based on the formula

$$Speedup = \frac{T_{seq}}{T_{para}}. \tag{5.48}$$

To be more precise, $T_{seq} \approx 14n$, so

$$Speedup \approx \frac{14t_c}{6t_w}\sqrt{p} = C\sqrt{p}, \tag{5.49}$$

where C is a constant.

5.3 Experiment

The algorithm developed in Section 5.2 is implemented on the Fujitsu AP1000 machine. The C code is written and tested in CASIM simulation environment; then it is sent to run in the real parallel machine.

In Section 5.3.1, we give a brief description of the AP1000 machine and the CASIM simulator. The experimental results are presented and analyzed in Section 5.3.2.

5.3.1 A Brief Description of AP1000 and CASIM

The AP1000 is an experimental large-scale distributed multiprocessor parallel computing system developed by Fujitsu Laboratories, Japan. Individual processors or "cells" are connected in the form of 2-D mesh, and they communicate with each other by sending and receiving messages via three separate high-bandwidth communications networks: the B-net, T-net, and S-net. AP1000 is a MIMD parallel computer, and the cells do not share memory. The AP1000 is connected to and controlled by a host computer which is typically a Sun SPARCServer.

Programs for the AP1000 are written in either C or FORTRAN. Library calls are used for communication over the networks mentioned above. An application program usually consists of a host program running on the host machine and a cell program running on the cells. The host program initiates the configuration of the system, assigns tasks to cells, passes input data to cells, and receives messages from cells. The cell program communicates with the host and other cells and performs the computation. For more detail about AP1000 programming refer to [22, 23, 24, 30].

CASIM [21] is a simulator that simulates AP1000. Programs written for AP1000 can be tested in a CASIM environment. A host and a number of cells (depending on the size of available memory) can be simulated in CASIM. When the CASIM runs, we have one window to represent the host and one for each cell. So, it is easy to trace the control of the program.

5.3.2 Experimental Results and Conclusions

The input data we use for our experiment are matrices of floating point numbers. The sizes of the matrices are 64×64, 128×128, 256×256, 512×512 and 1024×1024. All the input data are randomly generated by a C program (the raw data of 512×512 lena image is used. and the result is the same as using the 512×512 random data). Experiments on 2×2 processors, 4×4 processors, and 8×8 processors with different sizes of input data are performed.

Two Different Implementations

Results from two different implementations are compared. In the first implementation, each processor gets three whole blocks of data from three other processors, i.e., processor $P_{i,j}$ receives data from processors $P_{i,2j}$, $P_{i,2j+1}$, and $P_{i,2j+2}$ for row-wise 1-D FWT computation, and from processors $P_{2i,j}$, $P_{2i+1,j}$, and $P_{2i+2,j}$ for column-wise 1-D FWT computation. Thus, more information is transferred than necessary.

In the second implementation, each processor transfers two blocks of data and two columns or rows of data from the third block. In other words, processor $P_{i,j}$ receives the data blocks in processors $P_{i,-j}$, $P_{i,2j+1}$ and the first two columns of data from processor $P_{i,2j+2}$ for row-wise 1-D FWT computation, and receives the data blocks in processors $P_{2i,j}$, $P_{2i+1,j}$ and the first two rows of data from processor $P_{2i+2,j}$ for column-wise 1-D FWT computation. Thus, only the required data are transferred.

Tables 5.1, 5.2, and 5.3 list the experimental results from implementation 1 for different sizes of data on 2 × 2, 4 × 4, and 8 × 8 processors, respectively. Tables 5.4, 5.5, and 5.6 list the experimental results from implementation 2 for different sizes of data on 2 × 2, 4 × 4, and 8 × 8 processors, respectively. Comparing the corresponding results, the result from implementation 2 is better than from implementation 1. This coincides with the expectation because a smaller amount of data is transferred in implementation 2 than in implementation 1 and the branch overhead is less than the overhead of transferring the extra data.

Results with Different Numbers of Processors

From Tables 5.1 and 5.4, we get the constant $C \approx 0.77$ for the speedup formula (5.48) derived in Section 4.2, i.e.,

$$speedup \approx 0.77\sqrt{p} \tag{5.50}$$

From Tables 5.2 and 5.5, we get the constant $C \approx 1.0$ for the speedup formula (5.48), i.e.,

$$speedup \approx 1.0\sqrt{p} \tag{5.51}$$

From Tables 5.3 and 5.6, we get the constant $C \approx 1.5$ for the speedup formula (5.48), i.e.,

$$speedup \approx 1.5\sqrt{p} \tag{5.52}$$

The above derivations seem to conflict with each other, but careful analysis reveals the reason. When the number of processors is small, the parallelism in the communication pattern of the algorithm can not be fully explored. As the number of processors is increased, the parallelism can be more fully explored. As the processor number increases, the communication overhead decreases compared with the computation time, resulting in the overall increase in speedup. From this analysis, the constant C in the speedup formula (5.48) should be at least 1.5. Figure 5.5 shows the difference between the ideal speedup and the actual speedup.

Results with Small Number of Iterations

In many applications, it is often the case that we do not need to compute all *logn* iterations of the FWT, but only two or three iterations of the 2-D FWT may be sufficient. Tables 5.7, 5.8, and 5.9 show the results from three iterations of implementation 2 on different sizes of data on 2×2, 4×4, and 8×8 processors, respectively. These results show that more speedup is achieved when a smaller number of iterations is used. This is due to the fact that the number of processors used in the actual computation decreases as the number of iterations is increased.

Results with Different Sizes of Input

From any of the nine tables, we can conclude that the speedup increases as the size of the input data increases. In fact, when the size of input data is small, the communication overhead dominates the execution time because the amount of computation done by a processor is small. Figure 5.6 also shows the fact just mentioned.

Table 5.1 : Implementation 1 on 2 × 2 PEs

size	iteration	PE	para	seq	speedup
64 × 64	6	2 × 2	0.488925	0.728275	1.49
128 × 128	7	2 × 2	1.904083	2.986986	1.57
256 × 256	8	2 × 2	7.612872	11.631505	1.53
512 × 512	9	2 × 2	30.788989	47.312410	1.54
1024 × 1024	10	2 × 2	124.497876	188.434409	1.51

Table 5.2 : Implementation 1 on 4 × 4 PEs

size	iteration	PE	para	seq	speedup
64 × 64	6	4 × 4	0.246280	0.728275	2.96
128 × 128	7	4 × 4	0.788887	2.986986	3.79
256 × 256	8	4 × 4	2.980293	11.631505	3.90
512 × 512	9	4 × 4	11.816582	47.312410	4.00
1024 × 1024	10	4 × 4	48.121094	188.434409	3.92

Table 5.3 : Implementation 1 on 8 × 8 PEs

size	iteration	PE	para	seq	speedup
64 × 64	6	8 × 8	0.319232	0.728275	2.28
128 × 128	7	8 × 8	0.641096	2.986986	4.66
256 × 256	8	8 × 8	1.236428	11.631505	9.41
512 × 512	9	8 × 8	4.217318	47.312410	11.22
1024 × 1024	10	8 × 8	16.180682	188.434409	11.65

Table 5.4 : Implementation 2 on 2 × 2 PEs, exact data transferred

size	iteration	PE	para	seq	speedup
64 × 64	6	2 × 2	0.487405	0.728275	1.49
128 × 128	7	2 × 2	1.903066	2.986986	1.57
256 × 256	8	2 × 2	7.613414	11.631505	1.53
512 × 512	9	2 × 2	30.801409	47.312410	1.54
1024 × 1024	10	2 × 2	124.547476	188.434409	1.51

Table 5.5 : Implementation 2 on 4 × 4 PEs, exact data transferred

size	iteration	PE	para	seq	speedup
64 × 64	6	4 × 4	0.235471	0.728275	3.09
128 × 128	7	4 × 4	0.772334	2.986986	3.97
256 × 256	8	4 × 4	2.931493	11.631505	3.97
512 × 512	9	4 × 4	11.640385	47.312410	4.06
1024 × 1024	10	4 × 4	47.224833	188.434409	3.99

Table 5.6 : Implementation 2 on 8 × 8 PEs, exact data transferred

size	iteration	PE	para	seq	speedup
64 × 64	6	8 × 8	0.297807	0.728275	2.45
128 × 128	7	8 × 8	0.469157	2.986986	6.37
256 × 256	8	8 × 8	1.193738	11.631505	9.74
512 × 512	9	8 × 8	4.104928	47.312410	11.53
1024 × 1024	10	8 × 8	15.785377	188.434409	11.94

Table 5.7 : Implementation 2 on 2 × 2 PEs, only 3 iterations

size	iteration	PE	para	seq	speedup
64 × 64	3	2 × 2	0.473999	0.732229	1.55
128 × 128	3	2 × 2	1.844659	2.852886	1.55
256 × 256	3	2 × 2	7.373729	11.435084	1.55
512 × 512	3	2 × 2	29.860538	46.183462	1.55
1024 × 1024	3	2 × 2	120.682095	185.348481	1.54

Table 5.8 : Implementation 2 on 4 × 4 PEs, only 3 iterations

size	iteration	PE	para	seq	speedup
64 × 64	3	4 × 4	0.224600	0.732229	3.26
128 × 128	3	4 × 4	0.713379	2.852886	4.00
256 × 256	3	4 × 4	2.694941	11.435084	4.24
512 × 512	3	4 × 4	10.684133	46.183462	4.32
1024 × 1024	3	4 × 4	43.400305	185.348481	4.27

Table 5.9 : Implementation 2 on 8 × 8 PEs, only 3 iterations

size	iteration	PE	para	seq	speedup
64 × 64	3	8 × 8	0.294290	0.732229	2.49
128 × 128	3	8 × 8	0.406810	2.852886	7.01
256 × 256	3	8 × 8	0.961462	11.435084	11.89
512 × 512	3	8 × 8	3.150642	46.183462	14.66
1024 × 1024	3	8 × 8	11.969213	185.348481	15.49

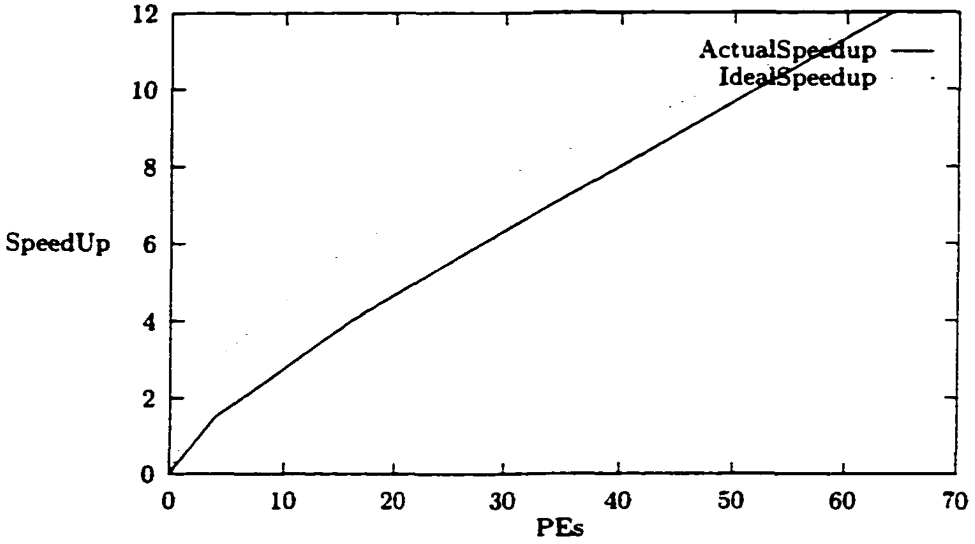

Figure 5.5 : Speedup vs. PE numbers

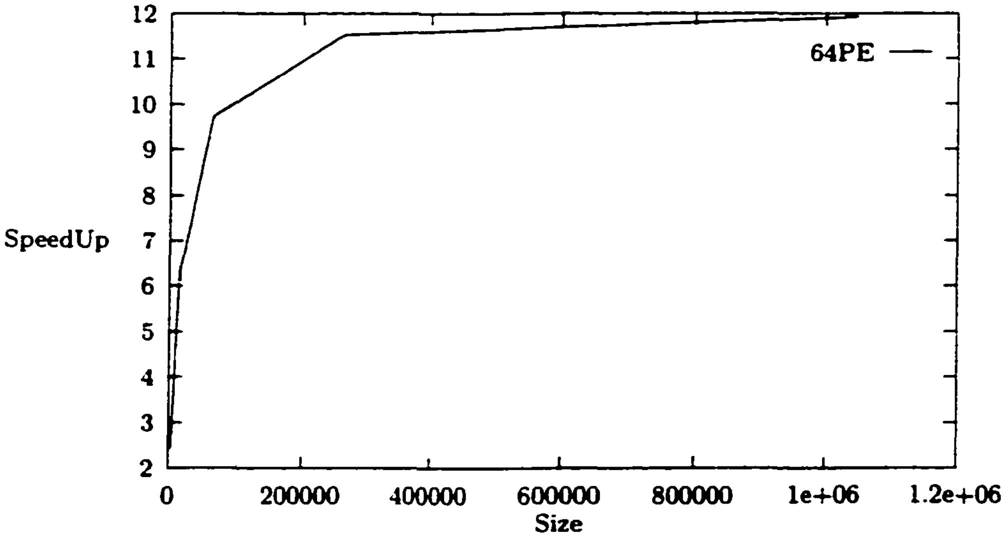

Figure 5.6 : Speedup vs. input sizes

5.4 Wavelet Packets

Wavelet packets [8] are extensions of wavelets. In each iteration, the wavelet transform only expands the coarse coefficients from the last iteration into a coarse part and a detailed part and leaves the detailed coefficients from the last iteration unchanged. Wavelet packets expand both the coarse coefficients and the detailed coefficients into a coarse part and a detailed part.

A wavelet packet transform can also be modeled mathematically as a linear transform $\grave{W}_n$ from S to WPS, the wavelet packet transform of S. If every detailed term is further decomposed, then the transform matrix $\grave{W}_n$ can be decomposed into the following form:

$$\grave{W}_n = \grave{T}_n \grave{T}_{n/2} ... \grave{T}_4 \grave{T}_2, \tag{5.53}$$

If we take $n = 8$, then the wavelet packet transform can be implemented as follows

:

$$S = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} \underset{\grave{T}_8}{\longrightarrow} \begin{bmatrix} c \\ c \\ c \\ c \\ d \\ d \\ d \\ d \end{bmatrix} \underset{\grave{T}_4}{\longrightarrow} \begin{bmatrix} c \\ c \\ d \\ d \\ c \\ c \\ d \\ d \end{bmatrix} \underset{\grave{T}_2}{\longrightarrow} \begin{bmatrix} c \\ d \\ c \\ d \\ c \\ d \\ c \\ d \end{bmatrix}$$

where c's and d's are used to denote the coarse and detailed coefficients, respectively.

If we choose the Daubechies order-4 wavelet for which the scaling constants are

$$\{h_k\} = \{\frac{1+\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, \frac{3-\sqrt{3}}{4}, \frac{1-\sqrt{3}}{4}\}, \tag{5.54}$$

then the length-8 wavelet packet transform is associated with the following T-matrices:

$$\grave{T}_8 = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & -h_2 & h_1 & -h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & 0 & 0 & 0 & 0 & h_3 & -h_2 \end{bmatrix}$$

$$\grave{T}_4 = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ h_2 & h_3 & h_0 & h_1 & 0 & 0 & 0 & 0 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ h_1 & -h_0 & h_3 & -h_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & h_2 & h_3 & h_0 & h_1 \\ 0 & 0 & 0 & 0 & h_3 & -h_2 & h_1 & -h_0 \\ 0 & 0 & 0 & 0 & h_1 & -h_0 & h_3 & -h_2 \end{bmatrix}$$

$\grave{T}_2 =$

$$\begin{bmatrix}
h_0+h_2 & h_1+h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
h_3+h_1 & -h_0-h_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & h_0+h_2 & h_1+h_3 & 0 & 0 & 0 & 0 \\
0 & 0 & h_3+h_1 & -h_0-h_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & h_0+h_2 & h_1+h_3 & 0 & 0 \\
0 & 0 & 0 & 0 & h_3+h_1 & -h_0-h_2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & h_0+h_2 & h_1+h_3 \\
0 & 0 & 0 & 0 & 0 & 0 & h_3+h_1 & -h_0-h_2
\end{bmatrix}$$

We now implement the wavelet packet transform algorithm using a pseudo language.

```
WPT (x : the 1-D input array ;
      size : the size of the input data;
      d : the number of iterations)
BEGIN
  len = size;
  for i := 0 to d-1 do
  begin
    hlen := len/2;
    for k := 0 to d-1 do
    begin
      for j := 0 to hlen-1 do
      begin
        $x[j + k * len] := h_0 * x[2j + k * len] + h_1 * x[2j + k * len + 1]+$
          $h_2 * x[(2j + k * len + 2)\%len] + h_3 * x[(2j + k * len + 3)\%len];$
```

```
            end
            for j := 0 to hlen-1 do
            begin
              $x[j + k * len + hlen] := h_3 * x[2j + k * len] - h_2 * x[2j + k * len + 1]+$
                $h_1 * x[(2j + k * len + 2)\%len] - h_0 * x[(2j + k * len + 3)\%len];$
            end
            len = len/2;
        end
    end
END
```

If $d = logn$, then the complexity of the above implementation is $O(nlog(n))$. Our parallel algorithm described in Section 4.2 can be modified to fit wavelet packet transforms; the complexity is still $O(n/\sqrt{p})$. Therefore, the speedup will be $O(\sqrt{p}log(n))$ in parallel wavelet packet transform implementation using our algorithm.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

6.1 On Signal Recovery

Two schemes are developed to reconstruct signals from their sample values. One of the algorithms is to regain a finite number of lost samples from a Nyquist-rate-sampled band-limited signal f of finite energy by replenishing new sample values of the same number. The other is to recover a band-limited function f of finite energy from its sampling values on real sequences with an accumulation point.

Although the two algorithms are both constructive and very programmable, implementations have not been provided. It could be a very interesting future research project to put the algorithms into real-world use.

Another interesting topic is worthy to be mentioned. Although the recovery schemes derived from sampling theory are closely related to complex function theory, the comparison between the classical complex function theory and the sampling schemes deserves some discussion. To our suprise, very few references of this kind can be found in the literature although many authors pointed out that the sampling theory is rooted deeply in classical complex function theory. The following statement can be found in [9]: "Cardinal series has found favor in signal-processing applications, undoubtedly because of the neat way in which it fits into the accompanying Fourier analysis."

Although the sampling theory has the complex function theory as its solid background, it was originally introduced ([7], [8], [11]) from the application domain. The sampling recovery schemes are more application-oriented while the methods that come directly from the classical complex function theory are stricter. The fact is that the two approaches are getting closer and no clear boundary currently exists.

6.2 On Frames

The existence of two kinds of frames, Weyl-Heisenberg frames and affine frames, is studied. Precisely, we provide conditions on a function $g \in L^2(\mathbf{R})$ such that the regular Weyl-Heisenberg system $\{e^{2\pi imbx}g(x-na)\}_{m,n\in\mathbf{Z}}$ or the semi-irregular Weyl-Heisenberg system $\{e^{2\pi imbx}g(x-a_n)\}_{m,n\in\mathbf{Z}}$ with an arbitrary real sequence $\{a_n\}_{n\in\mathbf{Z}}$ forms a Weyl-Heisenberg frame for $L^2(\mathbf{R})$. Also, conditions are given on $g \in L^2(\mathbf{R})$ such that the affine system $\{a^{-n/2}g((x-mba^n)a^{-n})\}_{m,n\in\mathbf{Z}}$ forms an affine frame. The conditions provided improve the known conditions and, in addition, are easy to verify. Some remarks on W-H frames are in order.

1. Theorem 4.4.1 still holds if $\{a_n\}_{n\in\mathbf{Z}}$ is a finite disjoint union of uniformly discrete sequences of real numbers. One can write $\sum_n |g(x-a_n)|$ as a finite sum of infinite series. The proof is analogous to the one of Corollary 4.2.2.

2. If there exist constants A. $B_0 > 0$ such that

$$\forall f \in L^2(R), \qquad A\,\|f\|_2^2 \le \sum_{m,n} |\langle f, E_{mb}T_{a_n}g\rangle|^2,$$

and $|g(x)| \le B_0$ a.e., then

$$\sum_n |g(x-a_n)| \ge A/B_0 \qquad a.e.$$

To see this, we assume ess $\sup_{x\in\mathbf{R}} \sum_n |g(x-a_n)| < A/B_0$. We can find a set $E \subset I \subset R$, where I is an interval of length $1/b$, such that $|E| > 0$ and

$\sum_n |g(x-a_n)| < A/B_0$ on E. If we set $f = \chi_E$ then it follows from (4.29) that

$$\begin{aligned}\sum_n\sum_m |\langle f, E_{mb}T_{a_n}g\rangle|^2 &= \int_E |f(t)|^2 \cdot \sum_n |g(t-a_n)|\ ^2dt \\ &\le B_0 \int_E \sum_n |g(t-a_n)|\ dt \\ &< A\,\|f\|_2^2\end{aligned}$$

This is a contradiction.

3. We conclude immediately from Remark 2 that, if there exists a constant A such that

$$\forall f \in L^2(\mathbf{R}), \qquad A\,\|f\|_2^2 \le \sum_{m,n} |\langle f, E_{a_n}T_{mb}g\rangle|^2,$$

and $|\hat{g}(\gamma)| \le B_0$ a.e., then

$$\sum_n |\hat{g}(\gamma - a_n)| \ge A/B_0 \qquad a.e.$$

4. So far as we know, no “if and only if ” conditions on mother wavelets have yet been obtained. It deserves more attention to search for sufficient and necessary conditions on g so that the irregular W-H system $\{E_{b_n}T_{a_n}g\}_{m,n\in\mathbf{Z}}$ is frame for $L^2(\mathbf{R})$, where $\Lambda = \{(a_n, b_n)\}_{n\in\mathbf{Z}}$ is a discrete subset of $\mathbf{R}\times\mathbf{R}$.

6.3 On Fast Wavelet Transformation

A parallel algorithm of fast wavelet transform is developed. The algorithm is implemented on the Fujitsu AP1000 multiprocessor machine. Experiments are performed on different input data sizes using various numbers of processors. The experimental results support our analysis.

To be more precise, with p processors working on an input of size n, the overall execution time is $6t_wn/\sqrt{p}+14t_cnlog(p)/p$, where t_w denotes the time used to transfer a unit data for one hop, and t_c the time needed to perform a multiplication or addition. The speedup is $O(\sqrt{p})$. The experiments show that the constant in the speedup term is about 1.5.

A parallel algorithm is developed only for the 2-D forward FWT. A similar algorithm may be developed for a 2-D inverse FWT. The algorithm can be modified for wavelet packet transforms.

To get good performance on a distributed data architecture, an algorithm must be implemented such that communication overhead is minimized. For an efficient parallel 2-D FWT, hypercube architecture may provide a better intercommunication topology.

REFERENCES

1. Ali N. Akansu and M. J. T. Smith. *Subband and Wavelet Transforms: Design and Applications*, Kluwer Academic Publishers, Boston, Dordrecht, London, 1996.

2. J. Benedetto, C. Heil, D.F. Walnut, *"Differentiation and the Balian-Low Theorem,"* Journal of Fourier Analysis and Applications, Vol. 1, No. 4, 1995.

3. A. Beurling and P. Malliavin, *"On the Closure of Characters and the Zeros of Entire Functions,"* Acta Math. 118 (1967), pp. 79-95.

4. G. Beylkin, R. Coifman and V. Rokhlin, *"Fast Wavelet Transforms and Numerical Algorithms I,"* comm. Pure Appl. Math. 44(1991) 141-183.

5. R. P. Boas, *Entire Functions* Academic Press, New York, 1954.

6. P. L. Butzer, W. SplettstÖsser, and R. Stens, *"The Sampling Theorem and Linear Prediction in Signal Analysis,"* Jber. d. Dt. Math.-Verein 90 (1987), pp. 1-70.

7. C. K. Chui, Wavelets: *"A Tutorial in Theory and Applications,"* Academic Press, San Diego, 1992.

8. M. A. Cody, *"The Wavelet Packet Transform Extending the Wavelet Transform,"* Dr. Dobb's Journal (April 1994).

9. I. Daubechies, *"Orthonormal Bases of Compactly Supported Wavelets,"* Comm. Pure and Appl. Math., 41(1988), pp. 909–996.

10. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadephia, PA 1992

11. I. Daubechies, A. Grossmann, Y. Meyer, *"Painless Nonorthogonal Expansions,"* J. Math. Phys., 27 (1986), pp. 1271-1283.

12. Daubechies, *"The Wavelet Transform, Time-frequency Localization and Signal Analysis,"* IEEE Trans. Inform. Theory, 36, pp. 961-1005, 1990.

13. I. Daubechies, A. Grossmann, and Y. Meyer, *"Painless Nonorthogonal Expansions,"* J. Math Phys., 27 (1986), pp. 1271-1283.

14. P. J. Davis, *Interpolation and Approximation*, New York, 1963.

15. R. Duffin and A. Schaffer, *"A Class of Nonharmonic Fourier Series,"* Trans. Amer. Math. Soc. 72 (1952), pp. 341-366.

16. E. Feig and C. A. Miccelli, "*L^2- Synthesis by Ambiguity Functions,*" Inter. Ser. Num. Math., 90 (1989), pp. 143-156.

17. M. Farge, "*Wavelet Transforms and Their Applications to Turbulence,*" Annu. Rev. Fluid Mech., 24(1992),pp. 395-457.

18. H. G. Feichtinger and K. GrÖchenig, "*Theroy and Practice of Irregular Sampling,*" Wavelets, Mathematics and Applications, pp. 305-363, 1994.

19. H. G. Feichtinger and K. GrÖchenig, "*Irregular Sampling Theorems and Series Expansions of Band-limited Functions,*" J. Math. Anal. Appl. 167 (1992), pp. 530-556.

20. H. G. Feichtinger and K. GrÖchenig, "*Iterative Reconstruction of Multivariate Band-Limited Functions From Irregular Sampling Values,*" SIAM J. Math. Anal. 23(1992), pp. 244-261.

21. FUJITSU LAB. LTD., *Casim User's Guide*, Casim Release 0.3.0, 1991.

22. FUJITSU LAB. LTD., *AP1000 Program Development Guide: C-Language Interface*, 1993.

23. FUJITSU LAB. LTD., *AP1000 Library Manual [Host]: C-Language Interface*, 1993.

24. FUJITSU LAB. LTD., *AP1000 Library Manual [Cell]: C-Language Interface*, 1993.

25. A. Grossmann and J. Morlet, "*Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape,*" SIAM J. Math, Anal., 15 (1984), pp. 723-736.

26. C. Heil, "*Generalized Harmonic Analysis in Higher Dimensions; Weyl-Heisenberg Frames and the Zak Transform,*" Ph.D. thesis, University of Maryland, College Park, MD, 1990.

27. C. E. Heil and D. F. Walnut, "*Continuous and Discrete Wavelet Transforms,*" SIAM Review, 31(1989), pp. 1-40.

28. J. R. Higgins. "*Five Short Stories about the Cardinal Series,*" Bull. Amer. Math. Soc., Vol. 12 (1985), pp. 45–89.

29. Y. Katznelson, *An Introduction to Harmonic Analysis*, John Wiley, New York, 1968.

30. C. Kim, *A Brief Description of the Communication Functions in the Fujitsu AP1000 Multiprocessor System*, preprint.

31. V. A. Kotel'nikov, *On the Transmission Capacity of "ether" and Wire in Electrocommunications*, Izd. Red. Upr. Svyazi RKKA(Moscow), 1933.

32. V. Kumar, A. Grama, A. upta and G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, The Benjamin/Cummings Publishing Company, Inc., New York, 1994.

33. S. Mallat, *"A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,"* IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.11, 7(1989), pp. 674-693.

34. S. Mallat, *"Multiressolution Approximations and Wavelet Orthonomal Bases of $L_2(\mathbf{R})$,"* Trans. of Amer. Math. Soc., Vol. 315, 1(1989), pp. 69-87.

35. R. J. Marks II, *Advances in Shannon Sampling*, Springer-Verlag, Berlin, Vienna, New York, 1993.

36. R. J. Marks II, *"Restoring Lost Samples from an Oversampled Bandlimited Signal,"* IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-31(1983), pp. 752-755.

37. F. Marvasti, P. M. Clarkson. M. V. Dokic, U. Goenchanart and C. Liu, *"Reconstruction of Speech Signals with Lost Samples,"* IEEE Trans. Signal Processing, Vol. 40, pp. 2879-2903.

38. F. Marvasti, *Nonuniform Sampling*, Advanced Topics in Shannon Sampling and Interpolation Theory, R. Marks II, Ed., Springer-Verlag, Berlin, Vienna, New York, 1993.

39. Martin Vetterli and Jelena Kovacevic, *Wavelets and Subband Coding*, Prentice-Hall, 1995.

40. Y. Meyer, *Wavelets-Applications and Algorithms*, SIAM, Philadelphia, 1993.

41. R.L. Motard and B. Joseph. *Wavelet Applications in Chemical Engineering*, Kluwer Academic Publishers, Boston, Dordrecht, London, 1994.

42. H. Nyquist, *"Certain topic in Telegraph Transmission Theory,"* AIEE Trans., Vol. 47 (1928), pp. 617-644.

43. A. Papoulis, *"A New Algorithm in Spectral Analysis and Band-Limited Extrapolation,"* IEEE Trans. Circuits Syst. CAS-22 (1975), pp. 735-742.

44. O. Rioul and P. Duhamel, *"Fast Algorithms for Discrete and Continuous Wavelet Transform,"* IEEE Trans. Inform. Theory, Vol. 38, no.2, pp. 569-586.

45. O. Rioul and M. Vetterli, *"Wavelets and Signal Processing,"* IEEE Signal Proc. Mag., October 1991, 14–38

46. D. Walnut, *Weyl-Heisenberg Wavelet Expansions: Existence and stability in Weighted Spaces*, Ph.D. thesis. University of Maryland, College Park, MD, 1989.

47. C. E. Shannon, *"Communications in Presence of Noise,"* Proc. IRE, Vol. 37 (1949), pp. 10-21.

48. J. L. Yen, *"On Nonuniform Sampling of Bandwidth-Limited Signals,"* IRE Trans. Circuit Theory, Vol. CT-3(1956), pp. 251-257.

49. R. Young, *In Introduction to Nonharmonic Fourier Series*, Academic press, New York, 1980.

50. M. Yuito and N. Matsuo, *"A New Sample-Interpolation Method for Recovering Missing Speech Samples in Packet Voice Communications,"* Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, 1989, pp. 381-384.

VITA

Changbai Xiao got his Bachelor degree in Computational Mathematics from Zhejiang University in 1982. He obtained his master and Ph.D. in Mathematics from Nanjing University in 1985 and 1988 respectively. From 1988 to 1994, he worked in Nanjing University teaching and pursuing research in Fourier analysis, Walsh Analysis and Local fields. Fifteen of his research papers have been published in high quality journals including IEEE Transactions on Signal Processing and Journal of Approximation. In 1993, he won a Young Scientist Grant the Chinese National Natural Science Foundation. This grant award is issued only to a handful of young scientists nationwide who has made significant achievements in their fields and has demonstrated extraordinary potential in contributing to natural sciences. He was profiled in the prestigious Marquis Who's Who in the World 1995. He has been reviewer for Journal of Approximation Theory, IEEE Transaction of Signal Processing, IEEE Signal Processing Letters and Mathematical Reviews. In 1996, he obtained his (second) Master degree in Computer Science from Louisiana Tech University. There after he has been working for eN2Z Corporation as a senior research engineer. At the mean time, he has been working hard to finish this dissertation to obtain his (second) Ph.D. degree in Applied Computational Analysis and Modeling.