Wavelet Applications in Sensing Technologies

Andrew K. Chan and Cheng Peng

July 3, 2003

Contents

Chapter 1

What This Book is About

Sensing technology is a broad discipline aimed at extracting information from events that are beyond the reach of human sensory. Although an individual is equipped with only five senses, he is able to detect and classify signals emitted from optical, acoustic, chemical, heat and mechanical energy sources. His senses allow him the enjoyment of life, as well as the protection from hostile agents. However, the limited range of human senses permits him to sense events occurring only in his immediate vicinity. Remote sensing, defined as "sensing at a distance," aims at increasing ranges and capabilities of the human sensory. Sensing technologies include both sensor design and methods of processing the sensor signals.

Sensing technologies have experienced explosive growth in the past 30 years partly due to demands from the space age, the computer age and the information age, and partly due to the advancements in computer technology and digital signal processing. Due to the increasing demands of better health care, higher requirements on national energy reserves, an increased protection of the environment and usage of land and sea resources, higher safety requirements on all modes of transportation and so on, there are numerous sensors and processing algorithms continuously being developed and implemented. Today, a desktop computer has more computing power than a mainframe computer fourty years ago. The gigahertz computing speed allows sensors to be operated at near real-time speed. Digital signal processing algorithms are being constructed, validated and implemented in applications in the global

economy, space expedition, machine control and food production. Without a doubt, sensing technology will continue to grow in this century in order to meet new challenges for a better living environment and meet new threats to our national security.

For a given application, signals from a sensor must be processed to reveal the information content. Proper use of this information helps to achieve beneficial results for that application. Major uses for signal processing include signal detection and classification for recognizing certain events, compression of signals for storage and efficient information transmission, denoising signals for revealing the true information content, to name a few. Many algorithms have been designed and implemented for signal processing. They may be classified as linear or nonlinear techniques, the Fourier spectral method or time domain method, online or offline processing and so on. Offline processing using Fourier based techniques has been the dominant approach for the past several decades. They are still the best choices for processing of periodic or stationary signals.

Although the wavelet analysis (not in the present form) was developed in the early 1900s, it had been neglected until mid-1980 when several significant developments occurred about the same time in the fields of mathematics, geophysics and signal processing. The wavelet theory was fully developed in the early 1990s and its applications to science and engineering blossomed throughout the decade. Wavelet applications are continually being developed virtually in every scientific discipline.

Wavelet analysis allows the user to divide a complicated signal into several components and process them individually. This important property, along with the high efficiency of its algorithms, makes wavelet analysis a very attractive tool for the analysis of signals. Wavelets and filter banks have been used in pattern recognition, image compression, mechanical fault diagnostics, statistical analysis and geophysical signal processing.

1.1 From Fourier Analysis to Wavelet Analysis

The most important problem in signal processing is to represent the signal accurately and effectively. Since most signals are non-stationary and of finite duration, the spectrum must contain all frequencies since the Fourier analysis uses sine and cosine functions (infinite duration function) as its basis. A particular event (in time domain) is dispersed in the spectrum since time domain information is lost by the transformation. The wavelet representation uses localized basis (finite duration or fast decaying functions) in the time domain so that the occurrence of an event can be localized and observed after wavelet transformation. The wavelet algorithms effectively generate a time-frequency transformation so that a given event is localized in a region on the time-frequency plane. Compared to the Fourier analysis, the wavelet analysis gives up a certain amount of frequency resolution but gains a lot of time domain information. By using the powerful multiresolution analysis, a signal can be effectively separated into components at different resolutions. This transformation is particularly suitable for adaptively processing these components for best results. Taking advantage of the multiresolution representation, we can combine wavelet techniques with other powerful processing tools such as the Markov random field, anisotropic diffusion, Embedded wavelet zerotree, neural network, support vector machine and so on for processing of one-, two- and three-dimensional signals.

There have been many books on wavelet analysis written by mathematicians, theoretical physicists and multirate signal processing theorists. These are monographs reporting the frontier of research on wavelet analysis in the author's field of specialty. Several standard texts covering the wavelet analysis with applications have appeared in the last few years. Books have been written on specific application areas such as image coding and book chapters written on multimedia applications. However, it appears that research works on wavelet applications to sensing technology are scattered among journals and conference proceedings of different disciplines. As a result, published works in one discipline may not be known to researchers in other disciplines. This was the motivation for the authors to write this volume to generate interest in using wavelet analysis for applications in sensing technology. Our goal for this book is three-fold: (1) introduce to the readers the Wavelet Transforms

for uses in sensing technology, (2) demonstrate the usefulness of combining the WT with other signal processing tools to solve many complicated problems in sensing technology and (3) produce several algorithms and matlab codes for readers to have hands-on experience. Except for some derivations on multiwavelets and the lifting algorithm, the mathematics can be understood with basic knowledge of linear algebra and vector space. Some knowledge in digital signal processing would be very helpful.

The book is organized with emphasis on using wavelets for applications in sensing technology. After a quick review of the Fourier analysis, the wavelet analysis is introduced in a parallel fashion. Without giving rigorous proofs, different types of wavelets are introduced and a brief outline on the wavelet construction is included in the appendices of Chapter 2. We emphasize the multiwavelet theory and algorithm for signal compression. Wavelet decomposition algorithm and other signal processing algorithms are introduced to demonstrate their power in solving problems in sensing technology. Neural network and the lifting algorithm are considered for efficient computation in solving real-time recognition and classification problems. Speckle noise filtering and smoothing is a major problem in satellite remote sensing. Various wavelet-based statistical techniques are considered for removing the speckles. Directional clutter is difficult to be removed from optical images captured by aerial photography. However, the development of directional wavelet transform (the X-ray wavelet transform) greatly reduces the clutter due to waves on the ocean surface. Compression algorithms are developed for remote compressing the satellite images before down linking to earth stations. Neural networks and support vector machine algorithms are coupled with statistical recognition techniques based on wavelet transforms to detect and classify wheel bearing faults. The Markov random field and best wavelet basis help to identify malignant masses and microcalcifications in screening mammograms. Wavelet techniques are used in well-log analysis to detect subsurface layer boundaries and the discrete wavelet transform has been used in two- and three-dimensional data for reduction of reservoir data.

Chapters 2 and 3 are devoted to the basics of wavelet analysis. The Fourier analysis is briefly reviewed for use in the parallel development of wavelet analysis. Window Fourier Transform is developed for the understanding of localized spectrum. Definitions for window measures are shown and comparisons are made between the window function and the

wavelet. The Continuous Wavelet Transform is defined and the scale and translation parameters are explained. The multiresolution analysis is briefly discussed so that the readers are familiar with the concept of nested subspaces. The two-scale relations and the decomposition relations are given without proof so that they can be used in the development of the discrete wavelet algorithms. The types of wavelet: the orthonormal, semi-orthogonal and the biorthogonal are defined and discussed. For efficient processing, the multiwavelet has been used in image compression. The GHM multiwavelets and optimal design multiwavelets are shown without proof. The latter is applied to SAR image compression. For those who wish to design wavelets for specific applications, two design methods are outlined in Chapter 2. The decomposition (analysis) and reconstruction (synthesis) wavelet algorithms are derived in Chapter 3. Their implementations using discrete convolution with up- and down-sampling are shown. For higher computational efficiency, the wavelet-lifting algorithm is given in Chapter 3. For certain applications, the wavelet packet algorithms are preferred over the wavelet algorithms. The generation of the wavelet packets and the associated algorithm are given. Finally, the Markov random field and the neural network algorithms are briefly mentioned. The application of these algorithms is shown in subsequent chapters.

Noise removal from SAR images is an important topic in remote sensing applications. The SAR is an active remote sensing device whose receiver is susceptible to electromagnetic scattering from objects around the target. The speckle noise comes from this backscattering. In Chapter 4, the noise statistics are reviewed and compared with those of the wavelet coefficients. The anisotropic diffusion method combined with a wavelet algorithm produces the best results compared to the conventional wavelet thresholding, wavelet Bayesian thresholding and wavelet Bayesian Markov random field. Aerial photography is a useful tool in subsurface sensing of the ocean. The aerial images often suffer from noise caused by the wind and tide. Since the motion of the ocean wave is directional, the x-ray wavelet algorithm is constructed to smooth out the wave. A subband filtering approach is performed on the wavelet coefficients for removing the directional ocean wave to reveal the ocean bottom sedimentary map.

Image compression techniques are introduced in Chapter 5. Traditional coding algorithms such as the Huffman code, arithmetic code, scalar and vector quantization code can be

applied to wavelet coefficients. However, the embedded zero tree code and the spatial oriented tree approach are adaptive coding techniques such that the coding gains can be controlled by the user. They are also capable of real-time progressive transmitting of the code words. The EZW and SPIHT algorithms are applicable to 2-D and 3-D compression. In addition, by combining the SPIHT and wavelet thresholding, it is possible to achieve noise removal and image compression simultaneously.

The topics in Chapter 6 include detection and classification of signals in remote sensing applications. In the last two chapters of this book, the meaning of remote sensing is broadened to include sensing systems other than satellite and aerial SAR imaging of the earth surface. In these chapters, signals to be analyzed include acoustic signals from faulty bearings and mammographic x-ray images. In Chapter 6, we apply multilayer perceptrons and support vector machines to the wavelet coefficients for detection and classification of bearing signals issued from the wheels of a train. Features and feature extraction are discussed in detail. Genetic algorithm is used to select the best subbands from wavelet decomposition and also select the most prominent features to reduce the computation load. The oversampling wavelet technique and wavelet packet transform for texture classification are also included in this chapter. Various neural network algorithms have been used in conjunction with wavelet transforms for classification of textures in SAR images.

Chapter 7 concerns two approaches for detection of malignant masses and microcalcification from mammograms. Multiresolution Markov random field is the main algorithm for separating images of the masses from the rest of the tissues. As a preprocessor, the fractal dimension is computed for each subimage to rule out those that contain no mass. Image features are defined for a binary tree detector. For detection of microcalcification, an adaptive wavelet packet algorithm is used to detect small bright spots on the mammogram. Rule based classification is employed to locate possible microcalcifications. Both algorithms have been tested using a mammographic image data base from great Britian.

The material of this book comes from the teaching notes and research articles of the authors. We also used other research articles in the literature to supplement the contents of certain topics. The book is designed not only as a reference for researchers and practicing

engineers in the field of remote sensing, it is also suitable as a supplementary text for a course in remote sensing techniques at the senior or graduate level. Topics can be selectively taught and amplified to supplement the contents of remote sensing courses. Several Matlab codes are furnished for hands-on practices.

Chapter 2

Wavelet Fundamentals

In the history of signal analysis, various ways have been used to represent a signal. A continuous time signal (analog signal) represented by $x(t)$ may exist in the time interval $(-\infty < t < \infty)$. In most engineering applications, the signals to be considered usually are causal signals, which exist only for $t > 0$. They may be finite duration signals ($x(t) = 0$ outside of the interval $t_0 \leq t \leq t_1$). These signals may be represented by a linear combination of basis functions such that a component associated with the chosen basis set may be singled out for analysis and processing. A signal representation takes on the form $x(t) = \sum_k c_k \phi_k(t)$ where $\{\phi_k(t)\}, k \in \mathbf{Z}$ is the basis function set and $\mathbf{Z}$ is the set of all integers. Intuitively, it is easy to see that if the signal $x(t)$ is an infinite duration signal, it is more natural to have $\phi_k(t)$ also be infinite duration. Hence, it is awkward and cumbersome to use an infinite duration basis function set to represent a finite duration signal. For example, it is well known that it takes an infinite number of sinusoidal functions (they are infinite duration basis functions) to exactly represent a finite duration rectangles pulse. Before the development of Fourier analysis, an analog signal way represented by a power series of complex variables. The basis functions used in this series representation are $\left\{t^k\right\}, k \in \mathbf{Z}_+$, where $\mathbf{Z}_+$ is the set of all positive integers. The work of Joseph Fourier opens up a new territory for signal presentation and analysis. Infinite duration signals (particularly periodic signals) are best represented by a linear combination of sinusoidal functions of different frequencies. One

may alter the given signal by processing the Fourier coefficients (frequency filtering) in order to achieve the objectives of a specific application. Since the development of the Haar basis (1910), a continuous function may be approximated arbitrarily close by an orthonormal basis with local support (finite duration). In engineering terminology, the Haar representation of a signal is the well-known staircase approximation of a function. Representation using the Haar basis has not found great popularity because the staircase approximation is a discontinuous function. Analog signals represented by orthonormal basis with local support become extremely useful after the development of Daubechies' orthonormal approximation function and wavelet in the mid 1980s..

2.1 From Fourier Analysis to Wavelet Analysis

Since its development in the 1800s, Fourier analysis has been used virtually in all branches of engineering and applied mathematical science. It serves as the backbone of all analyses in the frequency-domain. The Fourier analysis converts a signal represented in the time-domain (or spatial-domain) to one in the frequency-domain (or spatial frequency-domain). The frequency-domain representation reveals information from the signal that is not perceivable in the time-domain representation. This characteristic is particularly useful in analysis of acoustic signals and speech analysis in which tonal frequencies and characteristic frequencies are important features for recognition and classification. We give a very brief review of the Fourier analysis here primarily for the reader to recall terminologies in linear transformations.

2.1.1 Fourier Series

Let the signal $x(t)$ be periodic such that $x(t) = x(t + kT), \quad k \in \mathbf{Z}$ the Fourier series representation is an infinite sum of exponentials

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}. \tag{2.1}$$

where $\omega_0 = 2\pi f_0 = \frac{2\pi}{T}$ is the fundamental angulus frequency of the signal and T is the period. The coefficients are obtained based on the orthogonality between the exponential basis functions. Using an expression for the inner product $\langle f, g\rangle = \int_{-\infty}^{\infty} f(t)\overline{g(t)}dt$, where the over bar denotes complex conjugation, orthogonality between two exponential functions implies

$$\left\langle e^{jk\omega_0 t}, e^{j\ell\omega_0 t}\right\rangle = \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{jk\omega_0 t}e^{-j\ell\omega_0 t}dt = 0 \quad \text{if } k \neq \ell. \tag{2.2}$$

The Fourier coefficients are expressed using the inner product notation

$$c_k = \frac{1}{T}\left\langle x(t), e^{jk\omega_0 t}\right\rangle = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)e^{-jk\omega_0 t}dt. \tag{2.3}$$

The magnitude of the Fourier coefficient $|c_k|$ measures the strength of the k^{th} harmonic in the signal. The Fourier series is an energy preserving transformation so that the average power per cycle in the time-domain is the same as that in the frequency-domain

$$\frac{1}{T}\int_T |x(t)|^2\, dt = \sum_{k=-\infty}^{\infty} |c_k|^2\,. \tag{2.4}$$

This is the Parseval's identity for power signals.

2.1.2 Fourier Transform

Since most natural or man-made signals are not periodic, it is difficult to use the series expansion to access the frequency of a non-periodic signal. We remove the periodicity constraint on the signal by extending the period to infinity. Using the series expansion as in (2.1), we replace the coefficient c_k in (2.3)

$$\begin{aligned} x(t) &= \sum_{k=-\infty}^{\infty} \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t')e^{-jk\omega_0 t'}\,dt' e^{ik\omega_0 t} \\ &= \frac{1}{2\pi}\sum_{k=-\infty}^{\infty} \omega_0 \left[\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t')e^{-jk\omega_0 t'}\,dt'\right] e^{ik\omega_0 t}. \end{aligned} \tag{2.5}$$

As we extend the period T to infinity so that ω_0 becomes $d\omega$ and $k\omega_0$ approaches ω, the summation in (2.5) becomes an integral so that

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega \left[\int_{-\infty}^{\infty} x(t^{'}) e^{-j\omega t^{'}} dt^{'} \right] e^{i\omega t}. \tag{2.6}$$

We denote the quantity in the bracket by

$$\widehat{x}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt. \tag{2.7}$$

$\widehat{x}(\omega)$ is called the "Frequency spectrum" or simply "Spectrum," and is a continuous complex function of the frequency variable ω. Equation (2.7) is usually called the "analysis transform" or "forward transform." The magnitude of $\widehat{x}(\omega)$ represents the strength of the signal at ω while the phase of $\widehat{x}(\omega)$ is the signal delay. If we use the inner product notation, the analysis transform is written as

$$\widehat{x}(\omega) = \left\langle x(t), \quad e^{j\omega t} \right\rangle = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt. \tag{2.8}$$

The original signal may be recovered from (2.6) by

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{x}(\omega) e^{j\omega t} d\omega. \tag{2.9}$$

This inverse transform in (2.9) is a superposition integral and is also called the "Synthesis transform" in the sense that the original function $x(t)$ may be put back together from its frequency components in $\widehat{x}(\omega)$.

From this section, it shows that the Fourier analysis is the best tool for analyzing signals that are periodic with infinite duration. However, for signals that are transient in nature, such as music, image, speech, acoustic noise, seismic signals, thunder and lightning, representation of these signals by Fourier series or Fourier transform is very cumbersome. If one compares two signal representations, a rectangular impulse with very narrow time-width would take an infinite number of frequency components to reconstruct the impulse. On the other hand, if the signal is a sinusoidal function of a single frequency such as a 60Hz household electrical signal, the spectrum is a Delta function. It is well understood that fi-

nite duration signals are efficiently represented by localized finite energy basis functions. In certain signal processing applications where real-time computation is necessary, the Fourier transform cannot fulfill this requirement since it requires the knowledge of the entire signal before the computation begins. Another drawback of the Fourier analysis is that it cannot provide time domain and frequency-domain information simultaneously.

2.1.3 Window Fourier Transform

In order to gain time domain information from the spectrum, engineers have used the Window Fourier Transform (WFT), also called the Short Time Fourier Transform (STFT), to investigate the spectrum at a particular time interval of the signal. The WFT is formally defined by the integral transform

$$W_\phi x(t,\omega) = \int_{-\infty}^{\infty} x(\tau)\overline{\phi(\tau - t)}e^{-j\omega\tau}d\tau. \tag{2.10}$$

The function $\phi(t)$ is called the window function chosen by the user. There are many windows that can be chosen. The popular ones are the boxcar (rectangular), the Gaussian, raised cosine, Hamming and Hanning and so on. The mathematical expression for these window functions may be found in standard signal processing texts. The time-frequency variable pair (t,ω) is the result from transforming from a single time variable τ. Hence, the WFT transforms a one-variable function into a two-variable function. The complex spectrum of $W_\phi x(t,\omega)$ gives approximate spectral properties of the signal in the vicinity of the time location t. It is only an approximation since the expression in WFT is the Fourier transform of the product of the functions $x(\tau)$ and $\overline{\phi(\tau - t)}$. In other words, WFT gives the spectrum of the product function $x(\tau)\overline{\phi(\tau - t)}$ instead of the function $x(\tau)$ alone. For example, if $\phi(t)$ is a rectangular window with unit amplitude and $x(\tau)$ is a pure sinusoidal function $\cos(\omega_0 t)$, the spectrum of $x(\tau)\overline{\phi(\tau - t)}$ is not a Delta function in the ω-axis. It actually displays the convolution of the Delta spectrum with the sinc function spectrum resulting from the rectangular function. It broadens the Delta spectrum $\delta(\omega - \omega_0)$ to a $sinc(\omega - \omega_0)$ function due to the truncation of the cosine function. This is the windowing effect on the function $x(\tau)$. Intuitively, we want to use a window with long duration (large time-width)

to capture the low frequency component of a signal and a short time duration window for high frequency components. One must remember that the time domain resolution and the frequency resolution are governed by the uncertainty principle. However, the product of the time-duration and frequency bandwidth is a constant for a given window function ϕ.
The spectral domain equivalent of (2.10) comes from the Parseval identity, which gives

$$\begin{aligned} W_\phi x(t,\omega) &= \int_{-\infty}^{\infty} x(\tau)\overline{\phi(\tau-t)}e^{-j\omega\tau}d\tau. \\ &= \frac{1}{2\pi}e^{-jt\omega}\int_{-\infty}^{\infty}\widehat{x}(\theta)\overline{\widehat{\phi}(\theta-\omega)}e^{jt\theta}d\theta. \end{aligned} \tag{2.11}$$

Equation (2.11) indicates the time domain windowing process is also a spectral domain windowing process. It provides information on the spectral energy of the signal near the center of the window.

The original signal $x(t)$ is uniquely recovered from the WFT. Hence the WFT is a linear transformation with a unique inverse. The inversion formula for the WFT is given by

$$x(t) = \frac{1}{2\pi\left\|\phi\right\|}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_\phi x(\tau,\omega)\phi(\tau-t)e^{j\omega t}d\omega d\tau, \tag{2.12}$$

where $\|\phi\|$ is the norm of the function that will be defined in the next section.

2.1.4 Window Measures

Since there are many choices of window, there exist some measures of "goodness" for making choices on the window function. In wavelet literature, the RMS (Root Mean Square) width of the window function is used as a measure on the energy concentration in the time and frequency domains. These measures are explicitly expressed by

$$\Delta w = \frac{1}{\|w\|}\left\{\int_{-\infty}^{\infty}(t-t^*)^2\left|w(t)\right|^2 dt\right\}^{\frac{1}{2}}, \tag{2.13}$$

$$t^* = \frac{1}{\|w\|_+^2} \left\{ \int_{-\infty}^{\infty} t \, |w(t)|^2 \, dt \right\}, \tag{2.14}$$

and

$$\Delta\widehat{w}^+ = \frac{1}{\|\widehat{w}\|} \left\{ \int_0^{\infty} (\omega - \omega^{*^+})^2 \, |\widehat{w}(\omega)|^2 \, d\omega \right\}^{\frac{1}{2}}, \tag{2.15}$$

$$\omega^{*^+} = \frac{1}{\|\widehat{w}\|_+^2} \left\{ \int_0^{\infty} \omega \, |\widehat{w}(\omega)|^2 \, d\omega \right\}. \tag{2.16}$$

Here, the energy norm of a function

$$\|w\| = \left\{ \int_{-\infty}^{\infty} |w(t)|^2 \, dt \right\}^{\frac{1}{2}}, \tag{2.17}$$

$$\|\widehat{w}\|_+ = \left\{ \int_0^{\infty} |\widehat{w}(\omega)|^2 \, d\omega \right\}^{\frac{1}{2}} \tag{2.18}$$

has been applied to the RMS width. Δw measures one-half of the time domain energy concentration of the window. A smaller time window width means more energy is concentrated in a short time. High frequency features in a signal are extracted by adjusting the time window width to be narrow.

For a real signal, the magnitude of its Fourier transform is symmetric with respect to the origin of the frequency axis. We only need to use the positive $\omega-$axis. Hence, the frequency center ω^* is located on the ω^+- axis rather than at the origin. In general, one must compute the window widths numerically since the window function may or may not be integrable

With some simple algebra, it can be shown that the window area of the WFT on the $t-f$ plane is given by

$$[(t + t^* - \Delta w), (t + t^* + \Delta w)] \times [(\omega + \omega^{*^+} - \Delta\widehat{w}^+), (\omega + \omega^{*^+} + \Delta\widehat{w})^+]. \tag{2.19}$$

One may choose any window function to satisfy the processing goals, however, once the window $w(t)$ is chosen, the window widths $2\Delta w$ and $2\Delta\widehat{w}$ are fixed everywhere in the $t-f$ plane since they are independent of t and ω. This characteristic causes difficulty in processing non-stationary signals with time-dependent spectra. The user must scale the window several

times in order to extract features that may be hidden in the signal.

2.1.5 Wavelet Basis

For finite energy transient signals, it is natural to represent these signals by localized finite-energy bases. Wavelet bases are finite energy bases, and it is very effective to use wavelets for signal representation so that local signal components at different scales may be analyzed for various signal processing needs

Wavelets are finite-energy functions that are very different than the sinusoidal functions with infinite energy (big waves). A "wavelet" implies a function like "a small wave" with finite energy that can be localized around the RMS center of the function. A wavelet must have zero mean, which means $\int_{-\infty}^{\infty} \psi(t)dt = 0$. In the spectral-domain, $\widehat{\psi}(0) = 0$. In engineering terms, a wavelet has no d-c offset. In filter design literature, a wavelet always behaves as a bandpass filter. Some wavelet users have imposed the requirement of a multiresolution analysis on the wavelet in order for it to be implemented in the Discrete Wavelet Transform (DWT) algorithm. We will discuss this point in greater detail later in this chapter

For a given wavelet $\psi(t)$, a scaled and translated version of the wavelet forms a two-parameter function

$$\psi_{b,a}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right). \tag{2.20}$$

The parameter "a" corresponds to a scale of size a, while "b" is a translation parameter. The wavelet $\psi_{0,1}(t) = \psi(t)$ is the basic wavelet. The Daubechies D_2 (with four filter coefficients) wavelet is shown in Fig. 2.1.

The shape of the wavelet remains the same under translation and scaling. Choosing a parameter set (b, a) with a normalization constant, the set of wavelets $\psi_{k,a}(t), k \in \mathbf{Z}$ constitutes the basis functions of a Reisz (or stable) basis in the L^2 (or finite-energy) space. Recall that the Fourier series coefficients are the component values of projection onto different basis functions, the wavelet coefficients are signal components associated with wavelets at different scales and different locations along the time axis.

2.2 Continuous and Discrete Wavelet Transform

The Continuous Wavelet Transform (CWT) of a signal $x(t)$ is a linear transform defined by the integral with a chosen wavelet as the transform kernel

$$\begin{aligned} CWT_\psi x(b,a) &= \int_{-\infty}^{\infty} x(t)\overline{\psi_{b,a}(t)}dt \\ &= \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(t)\overline{\psi\left(\frac{t-b}{a}\right)}dt \\ &= \langle x(t), \psi_{b,a}(t)\rangle . \end{aligned} \tag{2.21}$$

$CWT_\psi x(b,a)$ is the wavelet coefficient at the time location b and scale a computed via the inner product formula. It indicates the correlation between $x(t)$ and $\psi_{b,a}(t)$. A strong correlation produces a high coefficient value

From the inner product representations, one may see immediately that the WFT and CWT are of similar nature since

$$W_\phi(b,\omega) = \int_{-\infty}^{\infty} x(t)\overline{\phi(t-b)}e^{-j\omega t}dt = \left\langle x(t), \phi(t-b)e^{j\omega t}\right\rangle . \tag{2.22}$$

They are interchangeable if we switch between them

$$\phi(t-b)e^{j\omega t} \leftrightarrow \psi_{b,a}(t). \tag{2.23}$$

The basis function $\phi(t-b)e^{j\omega t}$ is a single frequency sinusoidal function modulated by the window function $\phi(t)$. The window widths have been fixed by the choice of ϕ, independent of b and ω. On the other hand, the window widths for $\psi_{b,a}(t)$ depend on the scale while the window area in the $t-f$ plane

$$[t+a(t^*-\Delta\psi), t+a(t^*+\Delta\psi)] \times [\frac{1}{a}(\omega_+^*-\Delta\widehat{\psi}), \frac{1}{a}(\omega_+^*+\Delta\widehat{\psi})] \tag{2.24}$$

remains the same, namely: $4\Delta\psi\Delta\widehat{\psi}$. A large value of a (i.e. at low frequency) makes the time window width large and the spectral window width small. By varying the scale parameter a, the fine details of a signal can be separated from its coarse background at the locations

where the events occur. For signal recognition and classification, wavelet signal processing has this big advantage over the Fourier approach

The original signal can be uniquely recovered by the double integral

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} CWT_\psi(x)(b,a)\psi_{b,a}(t) \frac{dbda}{a^2}, \tag{2.25}$$

where C_ψ is a finite constant defined as

$$C_\psi = \int_{-\infty}^{\infty} \frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|} d\omega < \infty. \tag{2.26}$$

Unlike the Discrete Fourier Transform which computes the Fourier transform of a discrete signal

$$\widehat{X}(\omega) = \sum_{k=-\infty}^{\infty} x(k) e^{-jk\omega}, \tag{2.27}$$

the Discrete Wavelet Transform (DWT) computes the CWT at a certain discrete set of scale for a discrete signal. It is a fast algorithm to compute the CWT at a very sparse set of points on the $t - f$ plane. The perfect reconstruction property of wavelet allows the recovery of the original signal. Hence, the CWT is a linear transform with redundancy while the DWT is a shift variant transformation with no redundancy. To understand the basics of DWT, the multiresolution analysis (MRA) and the associated relationships between the functional spaces are discussed briefly in the following sections.

2.2.1 Multiresolution Analysis

A thorough understanding of the "Multiresolution Analysis" is important for the construction of a wavelet basis and for understanding the DWT algorithms. This analysis gives the user a systematic view of the signal at different resolutions. It acts as a zoom lens of a camera allowing the user to see the signal details (zoom in) or the broad background (zoom out). By applying the MRA to a signal, one separates a signal into many components at different resolutions so that each component may be processed with a different algorithm. Procedures may be made adaptive to the processing goals and the characteristics of the

signal. For a specific choice of the scaling parameter $a = 2^{-j}, j \in \mathbf{Z}$, the decomposition algorithm is equivalent to putting signal components into successive frequency octaves. Different filtering algorithms may be applied to different octave bands and the processing results can be recombined to form the processed signal.

Approximation subspace

A function $\phi(t) \in L^2$, is a scaling function (or approximate function) that generates a nested sequence of subspaces (approximation subspaces) $\{V_j\}_{j \in Z}$ of L^2 such that

$$\{\mathbf{0}\} \longleftarrow ... \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset ... \longrightarrow \mathbf{L}^2 \tag{2.28}$$

and at the same time satisfies a refinement equation

$$\phi(t) = \sum_k p_k \phi(at - k) \tag{2.29}$$

for some $a > 0, a \neq 1$, and coefficient sequence $\{p_k\} \in \ell^2$. The space ℓ^2 is the finite energy space for discrete signals. We choose $a = 2$ such that the two adjacent scales are twice and half of the original scale. This choice generates a dyadic octave-scale in the spectral-domain. That is, the center frequency of the scaled wavelet will be either half the frequency (coarser scale) or twice the frequency (finer scale). The subspace V_0 is generated by the scaling function at integer shifts $\{\phi(\cdot - k) : k \in \mathbf{Z}\}$. In general, V_n is generated by $\{\phi(2^n \cdot -k) : k \in \mathbf{Z}\}$. For the parameter-pair (b, a), we use $(\frac{k}{2^j}, \frac{1}{2^j})$ so that the CWT of a signal $x(t)$ at those parameters

$$\begin{aligned} CWT_\psi x(\frac{k}{2^j}, \frac{1}{2^j}) &= \int_{-\infty}^{\infty} x(t) \overline{\psi_{(\frac{k}{2^j}, \frac{1}{2^j})}(t)} dt \\ &= 2^{j/2} \int_{-\infty}^{\infty} x(t) \overline{\psi\left(2^j t - k\right)} dt \\ &= \left\langle x(t), \psi_{(\frac{k}{2^j}, \frac{1}{2^j})}(t) \right\rangle. \end{aligned} \tag{2.30}$$

The last expression is the CWT coefficient for $x(t)$ at $(\frac{k}{2^j}, \frac{1}{2^j})$.

Wavelet subspaces

Let the subspace V_n be an orthogonal sum (for which we use the symbol $\oplus$)of mutually orthogonal wavelet subspaces $W_j \in L^2$

$$V_n = \oplus_{j=-\infty}^{n-1} W_j. \tag{2.31}$$

Then

$$\begin{aligned} V_{n+1} &= \oplus_{j=-\infty}^{n} W_j \\ &= \oplus_{j=-\infty}^{n-1} W_j \oplus W_n \\ &= V_n \oplus W_n. \end{aligned} \tag{2.32}$$

The wavelet subspace W_n orthogonally complements V_n in V_{n+1}. This is similar to vector analysis where orthogonal components combine to form the vector in two dimensions. We observe the following characteristics.

(1) Subspaces $\{V_j\}$ are nested. That is, V_j is a proper subset of V_{j+1}.

(2) Subspaces $\{W_j\}$ are mutually orthogonal

We can write these statement succinctly by

$$V_j \cap V_\ell = V_j, \quad \ell > j \tag{2.33}$$

$$W_j \cap W_\ell = \{0\}, j \neq \ell \tag{2.34}$$

$$V_j \cap W_\ell = \{0\}, j \leq \ell. \tag{2.35}$$

The bases of subspaces $\{W_n\}$ are the results of the dilations and translations of some function $\psi(t)$, called a "wavelet," similar to the $\{V_n\}$ that are generated by $\phi(t)$. For any function $f_j(t) \in V_j$, we write

$$f_j(t) = \sum_{k=-\infty}^{\infty} c_{j,k} \phi(2^j t - k) = \sum_{k=-\infty}^{\infty} c_{j,k} \phi_{j,k}, \tag{2.36}$$

and another function $g_j(t) \in W_j$ as

$$g_j(t) = \sum_{k=-\infty}^{\infty} d_{j,k}\psi(2^j t - k) = \sum_{k=-\infty}^{\infty} d_{j,k}\psi_{j,k}, \tag{2.37}$$

for some coefficient sets $\left\{c_{j,k}\right\}_{k\in\mathbf{Z}}$ and $\left\{d_{j,k}\right\}_{k\in\mathbf{Z}}$ in $\ell^{\,2}$ space. The equations above use the indices (j,k) to indicate the level of resolution and the translation location

$$\phi_{j,k} = \phi(2^j t - k),$$

$$\psi_{j,k} = \psi(2^j t - k).$$

Any real signal $f(t)$ may be separated into many wavelet components $g_j(t), 1 \le j \le M'$ such that

$$f_M(t) = \sum_{m=1}^{M'} g_{M-m}(t) + f_{M-M'}(t) \tag{2.38}$$

where the function $f_M(t)$ is the approximation of $f(t)$ in the space V_M.

Two-Scale relations

The two-scale relation formalizes the relationship between two components at adjacent dyadic scales using

$$\phi(t) \in V_0 \subset V_1; \tag{2.39}$$

$$\psi(t) \in W_0 \subset V_1. \tag{2.40}$$

These relations are established by two sequences $\{p_k\}$ and $\{q_k\} \in \ell^2$ such that

$$\phi(t) = \phi(2^0 t) = \sum_k p_k \phi(2^1 t - k) = \sum_k p_k \phi_{1,k}(t), \tag{2.41}$$

$$\psi(t) = \psi(2^0 t) = \sum_k q_k \phi(2^1 t - k) = \sum_k q_k \phi_{1,k}(t). \tag{2.42}$$

For arbitrary $j \in \mathbf{Z}$, we have the two scale relations

$$\phi(2^j t) = \sum_k p_k \phi(2^{j+1}t - k), \tag{2.43}$$

$$\psi(2^j t) = \sum_k q_k \phi(2^{j+1}t - k). \tag{2.44}$$

Decomposition Relation

The decomposition relation separates a function at high resolution to two functions at lower resolutions. The subspace relation in an MRA

$$V_{j+1} = V_j \oplus W_j$$

indicates that there exists a relationship between the basis functions in these subspaces. Writing the two distinct basis functions $\phi(2t)$ and $\phi(2t-1)$ in V_1 using two sequences $\{a_k\}$ and $\{b_k\}$ in ℓ^2 such that

$$\begin{aligned} \phi(2t) &= \sum_k \{a_{2k}\phi(t-k) + b_{2k}\psi(t-k)\}; \\ \phi(2t-1) &= \sum_k \{a_{2k-1}\phi(t-k) + b_{2k-1}\psi(t-k)\}, \end{aligned} \tag{2.45}$$

one may generalize to arbitrary resolution j and for all $\ell \in Z$

$$\begin{aligned} \phi_{\ell,j+1} &= \phi(2^{j+1}t - \ell) \\ &= \sum_k \{a_{2k-\ell}\phi(2^j t - k) + b_{2k-\ell}\psi(2^j t - k)\} \\ &= \sum_k \{a_{2k-\ell}\phi_{j,k} + b_{2k-\ell}\psi_{j,k}\}. \end{aligned} \tag{2.46}$$

2.2.2 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) algorithms to be developed are based on the relations between subspaces described in the previous section. These are the most important algorithms in wavelet signal processing. It is based on a divide-and-conquer strategy that

separates a signal into components at various resolutions so that they may be adaptively processed for specific applications. For basic understanding of these algorithms, it suffices to state these algorithms for the purpose of explaining the meaning of each operation

The decomposition algorithm consists of two discrete convolutions followed by subsampling by "2" operation. We write

$$c_{j,k} = \sum_{\ell} a_{2k-\ell} c_{j+1,\ell}; \tag{2.47}$$

$$d_{j,k} = \sum_{\ell} b_{2k-\ell} c_{j+1,\ell}, \tag{2.48}$$

where $c_{j,k}$ and $d_{j,k}$ are the coefficients of the scaling function and wavelet at resolution "j" respectively. These formulas relate the coefficients of the scaling function from a given scale to the scaling function and wavelet coefficients at twice the given scale. In other words, a signal represented by a set of scaling function coefficients $c_{j+1,\ell}$ is separated into two signals represented respectively by the scaling function coefficient $c_{j,k}$ and wavelet coefficients $d_{j,k}$. It is easy to see that these relations compute the scaling function and wavelet coefficients at a resolution j from the scaling function coefficients at a higher resolution $j+1$. This is equivalent to drawing the high frequency information out from the original signal and leaving the rest in a lower resolution approximation space. The scaling function component at the resolution "j" can be further decomposed into two components at resolution "$j-1$". This process may be repeated indefinitely, or until there is no more information left in the signal

The sequence $\{a_n\}$ is the lowpass filter corresponding to a given scaling function while the sequence $\{b_n\}$ is the highpass filter. The wavelet coefficients $d_{j,k}$ contain high frequency information to be processed to achieve the desired goals. If these wavelet coefficients are not processed, they may be recombined (reconstructed) to recover the original signal. Sometimes the reconstruction of a signal is called the inverse DWT or IDWT

The reconstruction equation is expressed by

$$c_{j+1,\ell} = \sum_{k} [p_{\ell-2k} c_{j,k} + q_{\ell-2k} d_{j,k}] \tag{2.49}$$

where the right side of the equation corresponds to an up-sampling before discrete convolution.

2.2.3 Types of wavelet

The basis functions that span the subspace may form an orthogonal, a semi-orthogonal or a biorthogonal basis. Conditions are imposed on the scaling function and the wavelet for each type of basis. We will state these conditions separately for each type of basis.

Orthonormal Wavelet Bases

Based on the inner product defined in the L^2 space shown in previous sections, two functions are orthogonal to each other if the inner product of these functions is zero

$$< f, g >= \int_{-\infty}^{\infty} f(t)\overline{g(t)}dt = 0. \tag{2.50}$$

If $g(t) = f(t)$, and $< f, f >= \int_{-\infty}^{\infty} f(t)\overline{f(t)}dt = 1$, the function $f(t)$ is normalized. If the set of basis functions $\phi_j(t)$, $j \in Z$ spans an scaling space V_j with

$$< \phi_i, \phi_j >= \int_{-\infty}^{\infty} \phi(t-i)\overline{\phi(t-j)}dt = \delta_{i,j}, \tag{2.51}$$

it is an orthonormal basis set. The same condition applies to the wavelet subspace. A wavelet is said to be orthonormal if $\psi_{j,k} = 2^{j/2}\psi(2^j t - k)$ satisfies

$$< \psi_{i,\ell}, \psi_{j,k} >= \int_{-\infty}^{\infty} \psi_{i,\ell}(t)\overline{\psi_{j,k}(t)}dt = \delta_{i,j}\delta_{k,\ell}. \tag{2.52}$$

Since the wavelet subspace and the approximation subspace are orthogonal, we have

$$< \phi_{0,i}, \psi_{0,j} >= \int_{-\infty}^{\infty} \phi_{0,i}(t)\overline{\psi_{0,j}(t)}dt = 0, \ \forall\ i, j \in \mathbf{Z}. \tag{2.53}$$

Notice that the relationships in (2.51) to (2.53) are all referred to the same level of resolution. The wavelets at different resolution are also orthogonal to one another. That is

$$< \psi_{k,i}, \psi_{\ell,j} >= \delta_{i,j}\delta_{k,l}. \tag{2.54}$$

Orthonormal wavelet bases include the Haar system, the Daubechies orthonormal wavelet bases of all orders, the Meyer wavelets of all orders and the Battle-Lemarie orthonormal (non-compactly supported) spline wavelets. As a remark, both Meyer and Battle-Lamarie wavelets are not compactly supported (finite time duration), but the processing coefficients decay quickly so that the length of the processing filters can be truncated for real-time implementation.

Semi-orthogonal Wavelet Bases

It is known that the cardinal B-spline functions of orders higher than one, and their associated compactly supported spline wavelets, the integer translates of the scaling functions, i.e. the B-splines, as well as the spline wavelets are not orthogonal within their own subspaces. Mathematically, these conditions are

$$\int_{-\infty}^{\infty} N_m(t-i)\overline{N_m(t-j)}dt \neq \delta_{i,j}, \tag{2.55}$$

$$\int_{-\infty}^{\infty} \psi_{N_m}(t-i)\overline{\psi_{N_m}(t-j)}dt \neq \delta_{i,j}. \tag{2.56}$$

However, orthogonality still holds for the spline function N_m and the spline wavelet ψ_{N_m}, as well as translates of wavelets at different resolutions. They are called a semi-orthogonal wavelet. For efficient computations of the expansion coefficients, a dual scaling function $\widetilde{\phi}$ and a dual spline wavelet $\widetilde{\psi}$ can be found to satisfy the biorthogonality relation

$$< \phi_{k,i}, \widetilde{\phi}_{\ell,j} >= \delta_{i,j}\delta_{k,l} \tag{2.57}$$

$$< \psi_{k,i}, \widetilde{\psi}_{\ell,j} >= \delta_{i,j}\delta_{k,l}. \tag{2.58}$$

We remark here that both the dual approximation function (the dual B-spline) and the dual spline wavelet are both functions in the spline space of the same order. In other words, the B-spline and the dual B-spline are in the same spline space. These duals can be expanded into B-spline series of the same spline order.

Biorthogonal Wavelet Bases

Finite impulse response filters (FIR) with symmetric or antisymmetric coefficients are known to have linear phase or psuedo-linear phase characteristics. This property is important for minimizing signal distortion in the filtering process. The FIR filters $\{p_k\}$ and $\{q_k\}$ of finitely supported orthonormal wavelets are not symmetric. In designing wavelets, orthogonality may be given up to gain the symmetric filter property. If there exists a basis function set $\{\phi_k\}$ in which the members themselves are not orthogonal to one another, this is not an orthogonal basis. However if it further exists another basis set $\left\{\widetilde{\phi}_k\right\}$ such that the basis functions in this basis set is orthogonal to all other basis functions in $\{\phi_k\}$ except $\left\langle \phi_k, \widetilde{\phi}_k \right\rangle \neq 0$. We say that $\{\phi_k\}$ and $\left\{\widetilde{\phi}_k\right\}$ are biorthogonal basis for each other. The set $\left\{\widetilde{\phi}_k\right\}$ is called the dual basis of $\{\phi_k\}$ and vice versa. Filters designed in a two-channel filter bank are examples of a biorthogonal basis. Orthogonality between filters is not required. The only requirement for filter bank design is that the output of the filter bank is a delayed version of the input. This is the perfect reconstruction condition imposed on the design. Biorthogonal wavelets of Cohen, Daubechies and Feauvears are examples of this class of wavelet bases. One should know that in filter bank design, there may not be a scaling function or wavelet associated with designed filters, because the iterative procedure (described in the Appendices) may not converge to produce a graph of the wavelet. In other words, there may not be nested V spaces with the FIR filters as their two-scale sequences. For this reason, the two-channel filter bank is an excellent alternative for digital signal processing. Some properties of these biorthogonal wavelets are considered here

Given two biorthogonal basis pairs $\{\phi_j, \widetilde{\phi}_j\}$ and $\{\psi_j, \widetilde{\psi}_j\}$, If $\phi_j(x)$, $j \in Z$ spans an approximation space V_j and $\psi_j(x)$, $j \in Z$ spans an wavelet space W_j , we have a biorthogonal MRA

For a given function $f(t) \in V_0$, we may expand it in terms of the biorthogonal basis

$$f(t) = \sum_k c_k \phi_k(t) \tag{2.59}$$

and get the coefficients are obtained by

$$c_k =< f, \widetilde{\phi}_k > . \tag{2.60}$$

Similarly, if $f(x) \in \widetilde{V}_0$, we may also expand it in terms of the dual basis and get the coefficients through the biorthogonal basis. One can see clearly that V_0 and $\widetilde{V_0}$ are two different approximation subspaces. Each of which may generate a MRA of its own.

2.3 Multiwavelets

In [13] , it is stated that in order to reach the best performance in wavelet-based image compression, wavelet filters need many desirable properties including orthogonality and symmetry. However, these desirable properties are mutually exclusive and cannot be attained by the design of a single wavelet. The design objectives can be accomplished if more wavelets associated with the scaling function are considered in MRA. Hence, a multiwavelet set offers more design options and achieves several desirable properties. As stated in [15], multiwavelets (sometimes called the vector wavelets) offer the possibility of better performance for an image processing application than that from using scalar wavelets. It has been shown that the GHM mutliwavelet constructed by Donovan et.l. defeats scalar wavelet in signal denoising in [14] and [4]. However, the GHM multwavelet has no advantages in image compression [15]. In this section, we introduce a new multiwavelet family, the "optimum time-frequency resolution (OPTFR)" multiwavelet that is designed for image compression [11]. In [12], compression experiments show that the results from using OPTFR multiwavelets are better than using Daubechies' orthogonal wavelets and Daubechies' least asymmetric wavelets. The OPTFR performs even better than scalar (9-7)-tap biorthogonal wavelet for some images. Some generalities on multiwavelets and the GHM multiwavelets are presented here. Basic concepts on the construction of the OPTFR will be given in later sections

A set of functions $\psi_1, \psi_2, \cdots \psi_r \in L^2(R)$ is called "*orthogonal multiwavelet of multiplicity* r" if $\{\psi_1(2^j x - k), \cdots, \psi_r(2^j x - k), j, k \in Z\}$ forms an orthogonal basis of $L^2(R)$. The MRA discussed earlier in this chapter plays a vital role in multiwavelet construction. A MRA of *multiplicity* r is a nested sequence of closed subspaces $\{V_j\}$ in $L^2(R)$ satisfying the following conditions [11].

1.$V_j \subset V_{j+1}, j \in Z$.

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

3.$\overline{\bigcup_{j \in Z}} V_j = L^2(R)$

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

5.There exist r functions ϕ_1, $\phi_2, \cdots, \phi_r$ such that the collection of integer translates $\{\phi_j(x-k), 1 \leq j \leq r, k \in Z\}$ forms a Riesz basis of V_0.
The functions $\phi_1, \phi_2, \cdots, \phi_r$ are called *scaling functions*. If there exists a set of compactly supported approximation functions whose integer translates form an orthogonal basis of V_0, then $\{\phi_j(x-k), 1 \leq j \leq r, k \in Z\}$ is called an orthogonal basis of V_0. The $\{V_j\}$ is called an orthogonal MRA. For an orthogonal MRA $\{V_j\}$, subspaces $\{W_j\}$ is defined by $W_j = V_{j+1} \ominus V_j$. If there exists a set of functions $\psi_1, \cdots, \psi_r$ such that the integer translates of them form an orthogonal basis of W_0, then $\psi_1, \cdots, \psi_r$ are called *orthonormal multiwavelets*

Let $\Phi = (\phi_1, \cdots, \phi r)^T$ be a vector scaling function in a MRA of *muliplicity* r, there exist $r \times r$ matrix filter $H = \{H_k\}$ such that

$$\Phi(x) = 2 \sum_{k \in Z} H_k \Phi(2x - k). \tag{2.61}$$

The frequency response of the matrix filter is defined by $\widehat{H}(\omega) = \sum_{k \in Z} H_k e^{-jk\omega}$ so that we have $\widehat{\Phi}(\omega) = H(\omega/2)\widehat{\Phi}(\omega/2)$ in the frequency domain. Let $G = \{G_k\}$ be another matrix filter such that $\Psi(x) = 2 \sum\limits_{k \in Z} G_k \Phi(2x - k)$, where $\Psi = (\psi_1, \cdots, \psi_r)^T$ is the vector wavelet. The quantity (H, G) is called a multiwavelet filter bank, where H is called low-pass vector filter and G is called high-pass vector filter. The modulation matrix of an FIR multifilter

bank (H, G) is defined by

$$H_m(\omega) = \begin{bmatrix} H(\omega) & H(\omega+\pi) \\ G(\omega) & G(\omega+\pi) \end{bmatrix}. \tag{2.62}$$

Since a large value for *muliplicity* r brings a huge burden in computation, mathematicians and engineers design only *multiwavelet of multiplicity* 2. This multiwavelet is good enough to achieve many wavelet design objectives. For the remaining of this section, we focus on *multiwavelet of multiplicity* 2.

The GHM multiwavelet is the first example of orthogonal multiwavelets that was constructed by J. Geronimo, D. Hardin and P. Massopust as shown in [14] and [4]. The GHM multiwavelet has $r = 2$ and the filter length is 4. The two approximation functions are shown in Fig. 2.2 and Fig. 2.3, and the two wavelets are shown in Fig. 2.4 and Fig. 2.5.

GHM low-pass matrix filters are

$$\begin{aligned} H_0 &= \begin{bmatrix} \frac{3}{5} & \frac{4\sqrt{2}}{5} \\ -\frac{1}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix}, H_1 = \begin{bmatrix} \frac{3}{5} & 0 \\ \frac{9}{10\sqrt{2}} & 1 \end{bmatrix} \\ H_2 &= \begin{bmatrix} 0 & 0 \\ \frac{9}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix}, H_3 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{10\sqrt{2}} & 0 \end{bmatrix} \end{aligned} \tag{2.63}$$

and the high-pass matrix filters are

$$\begin{aligned} G_0 &= \frac{1}{10}\begin{bmatrix} -\frac{1}{\sqrt{2}} & -3 \\ 1 & 3\sqrt{2} \end{bmatrix}, G_1 = \frac{1}{10}\begin{bmatrix} -\frac{9}{\sqrt{2}} & -10 \\ -9 & 0 \end{bmatrix} \\ G_2 &= \frac{1}{10}\begin{bmatrix} -\frac{9}{\sqrt{2}} & -3 \\ 9 & -3\sqrt{2} \end{bmatrix}, G_3 = \frac{1}{10}\begin{bmatrix} -\frac{1}{\sqrt{2}} & 0 \\ -1 & 0 \end{bmatrix}. \end{aligned} \tag{2.64}$$

It has been shown that GHM multiwavelet has many useful properties. Two scaling functions are symmetric; one wavelet is symmetric and the other one is antisymmetric. One scaling function is compactly supported in [0,1] and another scaling function and wavelet functions are compactly supported in [0,2]. The system achieves approximation of order 2. The following procedure demonstrate the use of multiwavelet in processing a signal

Given a signal $x(t) \in V_j$, we expand the signal into components with scaling functions

$$x(t) = 2^j \sum_n \sum_{i=1}^{2} s_{ni}^j \phi_i(2^j t - n), \tag{2.65}$$

where $s_{ni}^j = \langle x(\cdot), 2^j \phi_i(2^j \cdot -n) \rangle$ are the scaling function coefficients

The coefficient set $\left\{ s_n^j = \left(s_{n1}^j, \cdots, s_{nr}^j \right)^T : n \in Z \right\}$ in the above function completely determines $x(t)$. Since $V_j = V_{j-1} \oplus W_{j-1}$, we write $x(t)$ as

$$x(t) = 2^{(j-1)/2} \sum_n \sum_{i=1}^{2} s_{ni}^{j-1} \phi_i(2^{j-1} t - n) + 2^{(j-1)/2} \sum_n \sum_{i=1}^{2} d_{ni}^{j-1} \psi_i(2^{j-1} t - n). \tag{2.66}$$

where $\left\{ d_n^j = \left(d_{n1}^j, \cdots, d_{nr}^j \right)^T : n \in Z \right\}$ is the set of multiwavelet coefficients

Replacing $x(t)$ in (2.66) by (2.65), multiplying both sides of the equation with $2^{(j-1)/2} \phi_i(2^j t - k)$, and taking an integral over R, we obtain

$$s_k^{j-1} = \sqrt{2} \sum_n H_{n-2k} s_n^j. \tag{2.67}$$

Multiplying both sides of the above equation with $2^{(j-1)/2} \psi_i(2^j t - k)$, and taking an integral over R, we obtain

$$d_k^{j-1} = \sqrt{2} \sum_n G_{n-2k} s_n^j. \tag{2.68}$$

The reconstruction formula is given by the expression,

$$s_n^j = \sqrt{2} \sum_n H_{k-2n} s_n^{j-1} + \sqrt{2} \sum_n G_{k-2n} d_n^{j-1}. \tag{2.69}$$

In processing signals with multiwavelets, the user needs to generate multiple input data streams for the vector filter. It is very important to map the one-dimensional signal into a multi-dimensional data format. This mapping procedure is called *multiwavelet prefiltering*. For $r = 2$, the basic *multiwavelet prefilter* is a 2×2 matrix such that

$$s_k^0 = P \begin{pmatrix} x_{2k-1} \\ x_{2k} \end{pmatrix}. \tag{2.70}$$

The data stream is partitioned into a sequence of two $one-dimensional$ vectors and applied to a filter defined by 2×2 matrix P. The postfilter Q satisfies the following condition $PQ = I$, where I is the identity matrix. It is necessary to apply prefiltering before MWT and postfiltering after inverse MWT. If the MWT is nothing but an identity matrix, the output of the pre- and post-filtering will produce a signal identical to the input. Two commonly used prefilters are listed as follows.

1. The subsample prefilter $P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

2. The Xia prefilter $P = \begin{bmatrix} 2 + \frac{\sqrt{2}}{10} & 2 - \frac{\sqrt{2}}{10} \\ \sqrt{2} + \frac{3}{20} & \sqrt{2} - \frac{3}{20} \end{bmatrix}$

The diagram of 1-D DMWT is shown in the Fig. 2.6. It is trivial to generalize 1-D DMWT to 2-D DMWT. The diagram of 2-D DMWT (2-level) is shown in Fig 2.7.

The procedure for processing an image using the 2-D DMWT is given below

1. A prefilter is applied to all rows of an image. Every row is partitioned into two data streams

2. The DMWT is applied to each prefiltered row. The first half of each row contains coefficients corresponding to the two lowpass filters. The second half of each row contains coefficients corresponding to the two highpass filters

3. The same prefilter is applied to all columns of the image. Every column is partioned into two data streams

4. Perform the same DMWT to all columns as in step 2

The output of 1-level 2D-DMWT is shown in Table 2.1.

The identity of the data H_1L_2 contains high-frequency components corresponding to the first wavelet filter and low-frequency components from the second lowpass filter. Just as in 2-D DWT, it is straightforward to process multi-level 2-D DMWT. The second level of 2-D DMWT is obtained by applying the DMWT to the sub-image composed of

L_1L_1, L_2L_1, L_1L_2,and L_2L_2. Fig. 2.9 shows an example of 2-D DMWT on a SAR image shown in Fig. 2.8.

2.4 Wavelet Design

2.4.1 Orthogonal Wavelet Construction from Orthonormal Quadratic Mirror Filter

Not all wavelets are suitable for a particular application. Different applications require different wavelets with specified properties for efficient and effective processing. In the process of constructing a wavelet, there are choices to be made on certain parameters so that the wavelet may be designed to achieve certain processing goals. This section highlights the procedure for construction of wavelets from an orthonormal Quadratic Mirror Filter bank (QMF). In section 2.4.1, an orthonormal QMF filter is denoted by (H, G), where H is the low-pass filter and G is the corresponding high-pass filter

Assuming the filter length $L > 0$ is given, the perfect reconstruction condition for H is given by $\sum_{l=0}^{L-1} h_l h_{l+2k} = \delta_{0k}$ and the normalizing condition is given by $\sum_{l=0}^{L-1} h_l = \sqrt{2}$. Combining both conditions, one obtains an equation system for $H = (h_0, h_1, \cdots, h_{L-1})$. There are L unknown variables and $(\frac{L}{2} + 1)$ equations in this system. That means $\frac{L}{2} - 1$ free parameters can be chosen in the wavelet design process. In her paper [1], Daubechies gives a method for constructing compactly supported orthogonal wavelet filterbanks $\{H, G\}$ from QMF filters. Given the low past filter H, the highpass filter $G = (g_0, g_1, \cdots, g_{L-1})$ is determined from $g_i = (-1)^i h_{L-1-i}$. She adds the highest vanishing moments to the QMF filterbank and generates the famous Daubechies wavelet family. Since then, several optimal wavelet design approaches have been proposed that have relaxed the vanishing moment constraints for wavelet designs based on other new criteria. In [8], a new class of wavelets has been designed with interesting approximation properties. In [9], the optimal wavelets for image representation have been constructed. In [10], the signal-adapted wavelet design is formulated as a linear semi-infinite programming (SIP). The SIP technique is capable of locating the globally optimal wavelet

for a given input signal.

Parameterization of FIR filters satisfying the orthogonal condition and normalized conditon is given by

$$\begin{aligned} H_L(z) &= \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} h_k z^k, \\ G_L(z) &= \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} g_k z^k. \end{aligned} \tag{2.71}$$

By induction, one obtains

$$\begin{pmatrix} H_L(z) \\ G_L(z) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} P_{\frac{L}{2}}(z) \\ (-1)^{\frac{L}{2}} Q_{\frac{L}{2}}(z) \end{pmatrix} \tag{2.72}$$

, where $P_i(z), Q_i(z)$ are determined by the following equations

$$\begin{pmatrix} P_i(z) \\ Q_i(z) \end{pmatrix} = \begin{pmatrix} \cos(\alpha_{i-1}) & \sin(\alpha_{i-1}) \\ -\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z^2 \end{pmatrix} \begin{pmatrix} P_{i-1}(z) \\ Q_{i-1}(z) \end{pmatrix} \tag{2.73}$$

and

$$\begin{pmatrix} P_1(z) \\ Q_1(z) \end{pmatrix} = \begin{pmatrix} \cos(\alpha_0) & \sin(\alpha_0) \\ -\sin(\alpha_0) & \cos(\alpha_0) \end{pmatrix} \begin{pmatrix} 1 \\ z \end{pmatrix}, \tag{2.74}$$

where $(\alpha_0, \alpha_1, \cdots, \alpha_{\frac{L}{2}-1})$ are constrained by $\sum_{j=0}^{\frac{L}{2}-1} \alpha_j = \pi/4$. The parametric form of orthonormal QMF filter for $L = 4$ is shown here as an example.

$$\begin{aligned} H(\alpha_0, \alpha_1) = \{&\cos(\alpha_0)\cos(\alpha_1), \cos(\alpha_1)\sin(\alpha_0), \\ &-\sin(\alpha_0)\sin(\alpha_1), \cos(\alpha_0)\sin(\alpha_1)\}. \end{aligned} \tag{2.75}$$

We remark here that not all orthonormal QMF filters generate orthonormal wavelet bases in $L^2(R)$ because of non-convergence. In [3], Lawton proposed a necessary and sufficient condition under which the orthonormal filter determines orthonormal wavelet bases. We outline a simple approach for orthonormal wavelet design with some specific criteria.

1. Set an optimal criterion of wavelet based on specific requirements from the application.

Vanishing moment is a good example.

2. Generate the parameterization for the orthonormal QMF filter.

3. Optimize the parameters to determine the best filter.

4. Check if the optimized filter satisfies the Lawton condition.

2.4.2 Construction of OPTFR Multiwavelet

In section 2.4.1, we briefly introduce the approach to construct an orthogonal wavelet form Quadratic Mirror Filter bank based on certain optimal criterion. In this section, we introduce a procedure for the design of orthogonal multiwavelets with optimal time-frequency resolution. A good time-frequency localization property of multiwavelet is very important for the signal processing such as image compression

Setting the mathematical proof, we introduce the optimized multiwavelet design method proposed in [12]. The parameterization of multiwavelet filterbank is given by the following. One defines $H_1(\omega)$ and $H_2(\omega)$ as well as $G_1(\omega)$ and $G_2(\omega)$ as follows

$$\begin{bmatrix} H_1(\omega) \\ G_1(\omega) \end{bmatrix} = 1/2 \begin{bmatrix} 1 & 0 \\ \cos\theta_0 & \mp\sin\theta_0 \\ 0 & \pm 1 \\ \sin\theta_0 & \pm\cos\theta_0 \end{bmatrix} + 1/2 \begin{bmatrix} 1 & 0 \\ -\cos\theta_0 & \mp\sin\theta_0 \\ 0 & \mp 1 \\ -\sin\theta_0 & \pm\cos\theta_0 \end{bmatrix} e^{-j\omega} \tag{2.76}$$

where $\theta_0 \in (-\pi, pi]$. and

$$\begin{bmatrix} H_2(\omega) \\ G_2(\omega) \end{bmatrix} = 1/4 \begin{bmatrix} 1 & -1 \\ -\sqrt{2}\cos\theta_0 & \sqrt{2}\cos\theta_0 \\ \pm 1 & \mp 1 \\ -\sqrt{2}\sin\theta_0 & \sqrt{2}\sin\theta_0 \end{bmatrix} + 1/2 \begin{bmatrix} 1 & 0 \\ 0 & \mp\sqrt{2}\sin\theta_0 \\ \mp 1 & 0 \\ 0 & \pm\sqrt{2}\cos\theta_0 \end{bmatrix} e^{-jw}$$

$$+1/2 \begin{bmatrix} 1 & 1 \\ \sqrt{2}\cos\theta_0 & \sqrt{2}\sin\theta_0 \\ \pm 1 & \pm 1 \\ \sqrt{2}\sin\theta_0 & \sqrt{2}\sin\theta_0 \end{bmatrix} e^{-j2\omega}. \tag{2.77}$$

where $\theta_0 \in (-\pi, pi]$. Following this, we parameterize the orthogonal FIR multifilter banks as shown below. The whole design method starts from this parameterization.

$$\begin{bmatrix} H_{2N}(\omega) \\ G_{2N}(\omega) \end{bmatrix} = V_{N-1}(z^2)\cdots V_1(z^2) \begin{bmatrix} H_2(\omega) \\ G_2(\omega) \end{bmatrix}, \tag{2.78}$$

and

$$\begin{bmatrix} H_{2N+1}(\omega) \\ G_{2N+1}(\omega) \end{bmatrix} = V_N(z^2)\cdots V_1(z^2) \begin{bmatrix} H_1(\omega) \\ G_1(\omega) \end{bmatrix} \tag{2.79}$$

where $z = e^{j\omega}$, $V_K(z) = I_4 + (Z^{-1} - 1)A_k$ and

$$A_k = 1/2 \begin{bmatrix} 1 & \cos\theta_k & 0 & \mp\sin\theta_k \\ \cos\theta_k & 1 & \sin\theta_k & 0 \\ 0 & \sin\theta_k & 1 & \pm\cos\theta_k \\ \mp\sin\theta_k & 0 & \pm\cos\theta_k & 1 \end{bmatrix}. \tag{2.80}$$

with $\theta_k \in (0, \pi]$

An orthonormal multiwavelet Ψ is said to be balanced if its corresponding scaling function Φ satisfies $\widehat{\Phi}(0) = (1,1)^t/\sqrt{2}$. Experiments have shown that a balanced multiwavelet is essential for effective signal processing. As mentioned in the previous section, before DMWT is applied to the input signal, it must be partitioned into two data streams by prefiltering. If Ψ is not balanced, the simple prefiltering techniques such as subsampling may lead to mixing of low-pass components with high-pass components [15]. This mixing may cause strong oscillations in the reconstructed signal after signal processing. Suppose $\{H_N, G_N\}$ is a multiwavelet filterbank given by equations (2.78) or (2.79). $\Phi_N = \{\phi_{N,1}, \phi_{N,2}\}$ and $\Psi_N = \{\psi_{N,1}, \psi_{N,2}\}$ are corresponding scaling functions and wavelets which are supported in the interval $[0, N]$. The balanced mutliwavelet is constructed with the following procedure.

Let

$$\Phi_N^b = R_0 \Phi_N \tag{2.81}$$

and

$$\Psi_N^b = R_0 \Psi_N \tag{2.82}$$

where $R_0 = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$ be the balanced scaling function and the balanced wavelet. The multiwavelet filter bank corresponding to Φ_N^b, Ψ_N^b is

$$H_N^b(\omega) = R_0 H_N(\omega) R_0^T \tag{2.83}$$

and

$$G_N^b(\omega) = R_0 G_N(\omega) R_0^T. \tag{2.84}$$

Let us recall the window measure described earlier. The time-width of a window function w is defined by

$$\triangle_w := \left(\int_R (t - t^*)^2 |w(t)|^2 dt \right)^{1/2} \tag{2.85}$$

where t^* is the center of the window in the time domain. The spectral width of w is denoted by $\triangle_{\widehat{w}}$. It is well known from the uncertainty principle that $\triangle_w \triangle_{\widehat{w}} \geq \frac{1}{2}$. This time-bandwidth product is called the resolution cell of w. If the resolution cell of a function is closer to the lower bound (i.e. $\frac{1}{2}$), the resulting resolution (image resolution in this case) will be higher. If one considers the wavelet playing the role of a window function (as given in the WFT), the better wavelets are those whose resulting cells have the smallest area. The OPTFR multiwavelet pairs are constructed by minimizing the sum

$$S_N^b = \triangle_{\phi_{N,1}^b} \triangle_{\widehat{\phi_{N,1}^b}} + \triangle_{\psi_{N,1}^b} \triangle_{\widehat{\psi_{N,1}^b}} \tag{2.86}$$

and

$$\widehat{S}_N^b = \triangle_{\widetilde{\phi_{N,1}^b}} \triangle_{\widehat{\widetilde{\phi_{N,1}^b}}} + \triangle_{\widetilde{\psi_{N,1}^b}} \triangle_{\widehat{\widetilde{\psi_{N,1}^b}}} \tag{2.87}$$

under certain constraints. Using this multiwavelet design process, an example of OPTFR

multiwavelet is generated here for reference. Let us denote $J := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, $H_N^b(\omega) = R_0 H_N(\omega) R_0^T$, $G_N^b(\omega) = R_0 G_N(\omega) R_0^T$, For Φ_3^b and Ψ_3^b,we have

$$H_0 = \begin{bmatrix} .00790 & .006236 \\ .00789 & -0.6236 \end{bmatrix},$$
$$H_1 = \begin{bmatrix} .4920 & .0636 \\ -4.9210 & .0063109 \end{bmatrix}.$$

with $H_j = S_0 H_{3-j}, \quad j = 2, 3, \quad G_k = (-1)^{k+1} H_k$, $0 \leq k \leq 3$.
For $\widetilde{\Phi}_3^b$ and $\widetilde{\Psi}_3^b$, we have

$$H_0 = \begin{bmatrix} .00840 & .00647 \\ .0083 & -0.642 \end{bmatrix},$$
$$H_1 = \begin{bmatrix} .4915 & .0636 \\ -4.916 & .0068 \end{bmatrix}$$

with $H_j = S_0 H_{3-j}, \quad j = 2, 3, \quad G_k = (-1)^{k+1} H_k J, \quad 0 \leq k \leq 3$. This multiwavelet has been used for image compression with very good results as mentioned in the introduction of this section [12].

2.5 Appendices

2.5.1 Decomposition and Reconstruction Sequences for Orthgonal Basis

In this section, wavelet decomposition filter banks is denoted by $(a_k, b_k$ and wavelet reconstruction filter bank is denoted by $(p_k, q_k$. The relations between the decomposition sequences $\{a_k\}, \{b_k\}$ with the reconstruction sequences $\{p_k\}, \{q_k\}$ are the simplest for an orthonormal basis. For perfect reconstruction of the signal, we may use the following relations:

$$\begin{aligned} q_k &= (-1)^k p_{1-k} \\ a_k &= -p_k \end{aligned} \tag{2.88}$$

$$b_k = -q_k.$$

Once the p_k sequence has been found through the construction of the two-scale relation of the scaling function, the other sequences are determined.

2.5.2 Decomposition and Reconstruction Sequences for Semi-orthogonal Basis

The sequences for the two-scale relations in the m^{th} order spline space are

$$p_k^m = \begin{cases} \frac{1}{2^{m-1}} \begin{pmatrix} m \\ \ell \end{pmatrix}, 0 \le k \le m \\ 0, \quad \text{otherwise,} \end{cases} \tag{2.89}$$

$$q_k^m = \frac{(-1)^k}{2^{m-1}} \sum_{\ell=0}^{m} \begin{pmatrix} m \\ l \end{pmatrix} N_{2m}(k+1-\ell), \qquad k = 0, ..., 3m-2. \tag{2.90}$$

The two-scale relations for the dual spline and dual wavelets are slightly more complicated. Since the B-spline functions of order $m \ge 2$ are not orthogonal basis, the sequences $\{a_k\}, \{b_k\}$ come from the dual spline and dual wavelets are not finite, They are both infinite sequences with exponential decay.

$$\begin{aligned} a_k &= 2^{-m+1} \sum_{\ell=0}^{m} \begin{pmatrix} m \\ l \end{pmatrix} \sum_{2h=k-\ell-m+1}^{k-\ell+m-1} N_{2m}(m+k-2h-\ell) g_h; \\ b_k &= (-1)^{k+1} 2^{-m+1} \sum_{2h=k-2m}^{k-m} \begin{pmatrix} m \\ k-2h-m \end{pmatrix} g_h, \end{aligned} \tag{2.91}$$

where the z-transform of $\{g_h\}$ is the reciprocal of the z-transform of the Euler-Frobenius-Laurent Polynomial $E_\phi(z)$, namely

$$\begin{aligned} G(z) &= \sum_h g_h z^h := \frac{1}{E_\phi(z)} \\ &= \left[\sum_{h=-m+1}^{m-1} N_{2m}(m+h) z^h \right]^{-1}. \end{aligned} \tag{2.92}$$

2.5.3 Decomposition and Reconstruction Sequences for Biorthogonal Basis

In the case of biorthogonal wavelets, it is not easy to express the sequences explicitly in terms of the lowpass filter sequence $\{a_k\}$. $\{a_k\}$ is double-shift biorthogonal to the lowpass reconstruction sequence $\{p_k\}$.

$$2\sum_k a_k p_{k+2n} = \delta_n. \tag{2.93}$$

Similarly, the highpass decomposition sequence $\{b_k\}$ is double-shift birothogonal to the hiphpass reconstruction sequence $\{q_k\}$.

$$2\sum_k b_k q_{k+2n} = \delta_n \tag{2.94}$$

The biorthogonality between the lowpass and the highpass is also important.

$$2\sum_k a_k q_{k+2n} = 0 \tag{2.95}$$

and

$$2\sum_k b_k p_{k+2n} = 0. \tag{2.96}$$

Many biorthogonal filter banks have already been designed for processing. For more details of biorthogoanl wavelets, readers are referred to standard texts listed in the references.

2.5.4 Graphing the Approximation Functions

(A) Spectral method

Let us rewrite the two-scale relations for the approximation function

$$\phi(t) = \sum_k p_k \phi(2t - k).$$

It can be expressed in the spectral domain as

$$\begin{aligned}
\widehat{\phi}(\omega) &= \frac{1}{2}\sum_k p_k \widehat{\phi}(\frac{\omega}{2}) e^{-j\frac{\omega}{2}} \text{hh} \qquad (2.97)\\
&= P\left(e^{-j\frac{\omega}{2}}\right)\widehat{\phi}(\frac{\omega}{2})\\
&= P\left(e^{-j\frac{\omega}{2}}\right) P\left(e^{-j\frac{\omega}{2^2}}\right)\widehat{\phi}(\frac{\omega}{2^2})\\
&\quad .\\
&\quad .\\
&= \Pi_{k=1}^{N} P\left(e^{-j\frac{\omega}{2^k}}\right)\widehat{\phi}(\frac{\omega}{2^N})\\
&= \Pi_{k=1}^{\infty} P\left(e^{-j\frac{\omega}{2^k}}\right). \qquad (2.98)
\end{aligned}$$

The approximation function $\phi(t)$ can be obtained by finding the IFFT of the infinite product given by (2.98). The inifinte product must be truncated so that $\phi(t)$ may be computed with a finite amount of time

(B) Iterative method

We modify the two-scale equation to fit into an iterative scheme. Consider the expression

$$\phi_{n+1}(t) = \sum_k p_k \phi_n(2t-k),\ n = 0, 1, 2, ... \qquad (2.99)$$

The index n indicates the number of iterative loops. Initially, the user may use a delta sequence $\delta(n)$, or a rectangular pulse. The iteration converges quickly if the regularity of the function is high. In some designs and under certain conditions, the algorithm may not converge due to their low regularity. The final graph may be visualized by using the usual 1-D graphic display. One may consider using the Daubechies filter sequences and the spline sequence to graph the wavelet for experiment to study their properties

The reader may want to use the iterative method to compute the data value for the Daubechies orthonormal approximation function and the compactly supported cubic spline function. The $\{p_k\}$ sequence for these two approximation functions are:

1. Daubechies approximation function $(coefficient), \rightarrow \{p_k\} = \{0.48296, 0.83652, 0.22414, -0.12941$

2. Cubic Spline scaling function
   $, \rightarrow \{p_k\} = \{0.125, 0.5, 0.75, 0.5, 0.125\}$.

(C) Graphing the wavelets
The associated wavelet may be generated by a linear combination of approximation functions based on the 2^{nd} two-scale relation for the wavelets

$$\psi(t) = \sum_k q_k \phi(2t - k). \tag{2.100}$$

The relation between $\{p_k\}$ and $\{q_k\}$ depends on the wavelet type and the approximation function. One may use (2.103) here to generate the graph of different wavelets

(D) Examples
We compute the corresponding wavelets of the scaling function D_4 and N_4. The corresponding $\{q_k\}$sequences are:

1. Cubic Spline scaling function (N_4),
   $\rightarrow \{q_k\} = \frac{1}{8!}\{1, 124, 1677, 7904, 18482, 24264, 18482, 7904, 1677, 124, 1\ \}$.

Coefficients of the Daubechies $(9-7)$ are given in the following table

References

[1] I. Daubechies, "Orthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 41, pp. 909-996, 1988.

[2] A. Jain, "Fundamentals of digital image processing," Prentice-Hall International, Inc. 1989.

[3] W. Lawton, "Tight frames of compactly supported affine wavelets," *J. Math. Phys.*, vol. 31.8, pp. 1898-1901, Aug.1990.

[4] J. Geronimo, D. Hardin and P. Massopust, "Fractal functions and wavelet expansions based on several scaling functions," *J. Approx. Theory*, vol.78, pp. 373-401, 1994.

[5] X. Xia, J. Geronimo, D. Hardin and B. Suter, "Design of prefilters for discrete multi-wavelet transforms," *IEEE Trans. on Signal Processing*, vol. 44, No. 1, pp. 25-35, Jan. 1996.

[6] A. Averbuch, D. Lazar and M. Israeli, "Image Compression using wavelet transform and multiresolution decomposition," *IEEE Trans. on Image Processing*, vol. 5, No.1, pp. 4-15, Jan. 1996.

[7] E. Silva and M. Ghanbari, "On the Performance of linear Phase wavelet transforms in low bit-rate image coding," *IEEE Trans. on Image Proce,* vol. 5. No. 5, pp.689-704, May 1996.

[8] J. Odegard and C. Burrus, "New class of wavelets for signal approximation," *IEEE ISCAS '96.* vol 1.2, pp.189-192, 1996.

[9] Y. Zhuang, "Adaptive wavelet system for image representation", *ICSP 1996,* vol. 1. pp. 291-294, 1996.

[10] P. Moulin, M. Anittescu, K. Kortanek and F. Porta, "The role of linear semi-infinite programming in signal-Adapted QMF bank design", *IEEE Trans. on Signal Processing*, vol. 45, no. 9, pp. 2160-2174 Sep. 1997.

[11] Q. Jiang, "On the design of multifilter banks and orthonormal multiwavelet bases," *IEEE Trans. on Signal Processing*, vol. 46, pp 3292-3303, Dec. 1998.

[12] T. Xia and Q. Jiang, "Optimal multifilter banks: design, related symmetric extension transform, and Application to image compression," *IEEE Trans. on Signal Processing*, vol. 47, No. 7. pp. 1878-1889, Jul. 1999.

[13] M. Martin and A. Bell, "New image compression techniques using multiwavelets and multiwavelets Packets," *IEEE Trans. on Image Processing*, vol. 10 No.4. pp. 500-510, Apr. 2001.

[14] G. Donovan, J. Geronimo, D. Hardin and P. Massopust, "Construction of orthogonal wavelets using fractal interpolation functions," *IAM J. Math. Anal.*, vol. 27. pp. 1158-1192, 1996.

[15] V. Strela, L. Shen, S. Lee and H. Tan, "A general approach for analysis and application of discrete multiwavelet transform," 1997, preprint.

Table 2.1: Percentage heroin measurements in nine illicit heroin preparations.

Sample	% Heroin			
1	2.2	2.3	2.2	2.3
2	8.4	8.7	2.2	2.3
3	7.6	7.5	2.2	2.3
4	11.9	12.6	2.2	2.3
5	4.3	4.2	2.2	2.3
6	1.1	1.0	2.2	2.3
7	14.4	14.8	2.2	2.3
8	21.9	21.1	2.2	2.3
9	8.8	8.4	2.2	2.3

Table 2.2: Daubechies biorthogonal 9-7 wavelet filterbank

k	a_k	b_k	p_k	q_k
-5				
-4	0.0267			
-3	-0.0169		-0.09127	-0.0535
-2	-0.0782	-0.0456	-0.0575	-0.0337
-1	0.2669	0.0288	0.5913	0.1564
0	0.6029	0.2956	1.1151	0.5337
1	0.26687	-0.5575	0.5913	-1.2059
2	-0.07822	0.2956	-0.0575	0.5337
3	-0.0169	0.0288	-0.0913	0.1564
4	0.0267	-0.0456		-0.0337
5				-0.0535

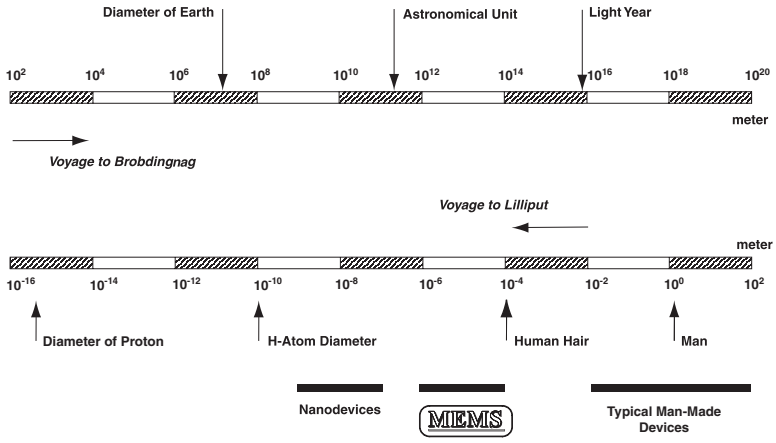

Figure 2.1: Daubechies D_2 Wavelet

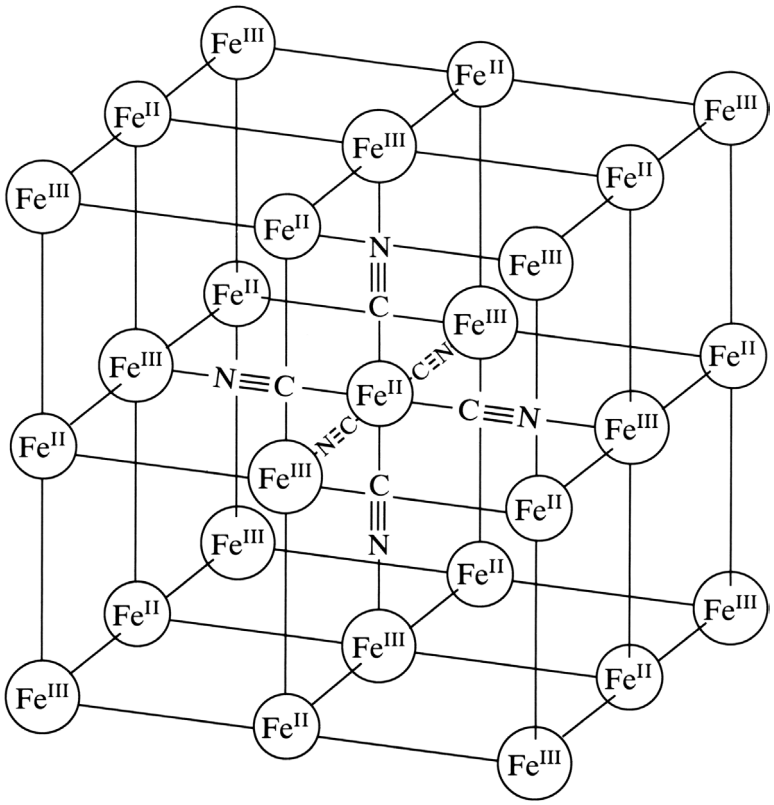

Figure 2.2: GHM approximation function 1

Figure 2.3: GHM approximation function 2

Figure 2.4: GHM wavelet function 1

Figure 2.5: GHM wavelet function 2

Figure 2.6: 1-D DMWT

Figure 2.7: 2-D DMWT

Figure 2.8: A SAR image

Figure 2.9: Multiwavelet decomposition of the SAR image shown in Figure 2.8.

Chapter 3

Wavelet Algorithms and Associated Techniques

We introduced the CWT in the last chapter. It provides a time-scale map of the signal that display the time and scale (frequency) information contained in the signal simultaneously. The frequency content of any event in the time domain can be easily seen in the map. The CWT and the Short-Time Fourier Transform (STFT) are similar in nature since their coefficients are computed from an integral. Computation-wise, they are both quite cumbersome and are not efficient by any measure. The conventional technique for computing the CWT involves convolution or FFT. Obviously, the CWT cannot be carried out in real-time. Compared with CWT, DWT is much more computationally efficent. These are some of the hindrances for further development of CWT. However, it is still a powerful analytical tool. For applications where rich information is desired, the data set is small and the computation time is not an issue, CWT is still preferred over the DWT.

3.1 Discrete Wavelet Transform and Filter Bank Algorithm Based on MRA

The Multiresolution Analysis (MRA) introduced in Chapter 2 forms the theoretical basis for the development of discrete wavelet algorithms. The two-scale relation and the decomposition relation are the foundations for the discrete-time computation for the wavelet transform. By eliminating the redundancy in the CWT, the DWT and IDWT (Inverse DWT) algorithms are very efficient and particularly suitable for real-time computation and hardware implementation. In DSP system representation, the DWT and IDWT algorithms are equivalent to the filter-bank algorithm in multirate signal processing.

3.1.1 Decimation and Interpolation

In multirate signal processing, a signal can be represented by more than one sampling rate. The basic techniques to archive sampling rate change are the decimation and interpolation. An M-point decimation is to select samples that are integer multiples of M from the given signal.

$$y(n) = x(nM) \quad \text{for } n \in Z \tag{3.1}$$

The frequency spectrum of the decimated signals is given by

$$\widehat{y}(e^{j\omega}) = \ 1/M \sum_{k=0}^{M-1} \widehat{x}(e^{j[(\omega-2\pi k)/M]}). \tag{3.2}$$

The diagram of decimation operation is shown in Figure 3.1, where $\{x(n)\}$ is the input discrete signal and $\{y(n)\}$ is the decimated output

For $M = 2$, the decimation operation is archived by taking every other data point (the even indexed sample).

$$\{y(n)\} = \{x(2n)\}. \tag{3.3}$$

The spectral domain relation of a decimator (M=2) can be obtained

$$\widehat{y}(e^{j\omega}) = 1/2[\widehat{x}(e^{j(\omega/2)}) + \widehat{x}(-e^{j(\omega/2)}) \ ]. \tag{3.4}$$

In terms of a matrix operator, the equation 3.3 can be rewritten as

$$\begin{bmatrix} . \\ . \\ y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ . \\ . \end{bmatrix} = \begin{bmatrix} . \\ . \\ x(-4) \\ x(-2) \\ x(0) \\ x(2) \\ x(4) \\ x(6) \\ . \\ . \end{bmatrix} = \begin{bmatrix} & & & & & & & & & \\ & & & & & & & & & \\ 1 & 0 & 0 & & & & & & & \\ 0 & 0 & 1 & 0 & 0 & & & & & \\ & & 0 & 0 & 1 & 0 & 0 & & & \\ & & & & 0 & 0 & 1 & 0 & 0 & \\ & & & & & & 0 & 0 & 1 & 0 \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{bmatrix} \begin{bmatrix} . \\ . \\ x(-2) \\ x(-1) \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ . \\ . \end{bmatrix} \tag{3.5}$$

or in a compact format

$$[y] = [D_{\downarrow 2}][x] \ . \tag{3.6}$$

Interpolation is the inverse operation of decimation. It inserts additional zeros between data to increase the sampling rate. The definition of interpolation by M in the time-domain is given by

$$y(n) = \begin{cases} x(n/M) & \text{if } n = kM \quad k \in Z \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

The diagram of interpolation is shown in Figure 3.2. It can also be represented by the discrete-time convolution

$$y(n) = \sum_k x(k)\delta(n - kM). \tag{3.8}$$

The spectrum of the output is

$$\widehat{y}(e^{j\omega}) \ = \ \sum_n \sum_k x(k)\delta(n - kM)e^{-jn\omega} \tag{3.9}$$

$$\begin{aligned} &= \sum_k x(k)e^{-jkM\omega} \\ &= \widehat{x}(e^{-jM\omega}). \end{aligned}$$

From Equation (3.9), the spectrum of the interpolation output is shrunk by a factor of M on the $\omega-$axis. Unlike the decimator, there is no problem of aliasing in the interpolation. For $M = 2$, we have

$$y(n) = \begin{cases} x(n/2) & \text{for } n = 2l \quad l \in Z \\ 0 & \text{for } n = 2l+1 \quad l \in Z \end{cases} \tag{3.10}$$

The matrix representation of this interpolator is

$$\begin{bmatrix} \cdot \\ \cdot \\ y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ x(-1) \\ 0 \\ x(0) \\ 0 \\ x(1) \\ 0 \\ x(2) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & 1 & 0 & & & & & \\ \cdot & \cdot & 0 & 0 & & & & & \\ \cdot & \cdot & 0 & 1 & 0 & & & & \\ \cdot & \cdot & 0 & 0 & 0 & 0 & & & \\ \cdot & \cdot & 0 & 0 & 1 & 0 & & & \\ \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & & \\ \cdot & \cdot & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \cdot & \cdot & & & & & & & \\ \cdot & \cdot & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ x(-2) \\ x(-1) \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ \cdot \\ \cdot \end{bmatrix} \tag{3.11}$$

or

$$[y] = [I_{\uparrow 2}][x]. \tag{3.12}$$

The interpolator and the decimator are important elements in filter bank (and DWT) algorithms. They are used in conjunction with digital filters for decomposition and recovery of signals. The basic operations in a two-channel perfect reconstruction filter bank consist of convolution; followed by decimation by 2 and the interpolation by 2; followed by convolution (see figure 3.3 and 3.4).

These basic operations will be used repeatedly to build pyramid algorithms and tree

algorithms for wavelets and wavelet packets as well as in two- and three-dimensional signal processing.

Mathematically, the convolution followed by decimation can be represented by

$$\begin{aligned} \{y(n)\} &= \{h(n) * x(n)\}_{\downarrow 2} \\ &= \{u(n) = \sum_k x(k)h(n-k)\}_{\downarrow 2} \\ &= \{u(2n)\} \\ &= \{\sum_k x(k)h(2n-k)\}. \end{aligned} \tag{3.13}$$

This is the decomposition relation given in Chapter 2. Similarly, interpolation followed by convolution can be written as

$$\begin{aligned} \{y(n)\} &= \{h(n) * [x(n)]_{\uparrow 2}\} \\ &= \{h(n) * u(n) \text{ where } u(n) = x(n/2)\} \\ &= \{\sum_k u(k)h(n-k)\} \\ &= \{\sum_{k\in even} x(k/2)h(n-k)\} \\ &= \{\sum_k x(k)h(n-2k)\}. \end{aligned} \tag{3.14}$$

When these two operations are connected in tandem, it constitutes the basic decomposition and recovery of a signal. Depending on the nature of the filters, we have a processing and recovery scheme for the signal within the bandwidth of the filters. The interpolation followed by convolution is the essential part of the wavelet synthesis algorithm. The decomposition provides a very useful tool for image denoising and compression. The algorithm is based on the decomposition relation in MRA discussed in Chapter 2. Let's emphasize several of these relations here for easy reference. Given $x_{s+1}(t) \in V_{s+1}$, $x_s(t) \in V_s$ and $y_s \in W_s$, we can represent all of these functions by their basis:

$$\begin{aligned} x_{s+1}(t) &= \sum_k a_{k,s+1}\phi_{k,s+1}(t) \\ x_s(t) &= \sum_k a_{k,s}\phi_{k,s}(t) \end{aligned} \tag{3.15}$$

$$y_s(t) = \sum_k w_{k,s}\psi_{k,s}(t),$$

where $\{a_{k,s}\}$ is the set of approximation coefficient at resolution s and $\{w_{k,s}\}$ is the wavelet coefficient set at resolution s. From the MRA,

$$V_{s+1} = V_s \oplus W_s, \tag{3.16}$$

there exists $x_s(t) \in V_s$ and $y_s(t) \in W_s$ such that

$$\begin{aligned} x_{s+1}(t) &= x_s(t) + y_s(t) \\ \sum_k a_{k,s+1}\phi_{k,s+1}(t) &= \sum_k a_{k,s}\phi_{k,s}(t) + \sum_k w_{k,s}\psi_{k,s}(t). \end{aligned} \tag{3.17}$$

By substitution of the decomposition relation,

$$\phi(2^{s+1} - l) = \sum_k h_0[2k - l]\phi(2^s t - k) + h_1[2k - l]\psi(2^s t - k) \tag{3.18}$$

into Equation 3.17, we obtain the following wavelet decomposition algorithm

$$\begin{aligned} a_{k,s} &= \sum_l h_0[2k - l]a_{l,s+1} \\ w_{k,s} &= \sum_l h_1[2k - l]a_{l,s+1}. \end{aligned} \tag{3.19}$$

where $(h_0(k), h_1(k))$ is wavelet decomposition filter bank. These two equations correspond to decimation by 2 after convolution. Let $a_s = \{a_{k,s}\}$ $w_s = \{w_{k,s}\}$ $h_0 = \{h_0[k]\}$ and $h_1 = \{h_1[k]\}$, the DWT is shown in Figure 3.5. We can also easily derive the wavelet synthesis transform (IDWT)

$$a_{l,s+1} = \sum_k \{g_0[l - 2k]a_{k,s} + g_1[l - 2k]w_{k,s}\}, \tag{3.20}$$

where $(g_0(k), g_1(k))$ is wavelet reconstruction filter bank. Equation 3.20 corresponds to interpolation followed by convolution. An $L-level$ wavelet decomposition is archived by applying DWT to the approximation coefficients L times. Figure 3.6 shows the implementation of an $L - level$ wavelet decomposition algorithm. Similarly, an $L - level$ wavelet synthesis can be

archived by applying IDWT L times to reconstruct the $L-level$ approximation coefficients.

3.1.2 2-D Wavelet Algorithm

When the input signal is two-dimensional (2-D) such as an image, it is necessary to represent the signal components by two-dimensional wavelets and two-dimensional approximation functions. For any approximation function ϕ and its corresponding wavelet ψ, there are three different 2-D wavelets and one 2-D approximation function constructed by using the tensor product approach. We express the 2-D wavelets in the following manner:

$$\Psi^{[1]}_{i,j}(x,y) = \phi(x-i)\psi(y-j), \tag{3.21}$$

$$\Psi^{[2]}_{i,j}(x,y) = \psi(x-i)\phi(y-j), \tag{3.22}$$

$$\Psi^{[3]}_{i,j}(x,y) = \psi(x-i)\psi(y-j), \tag{3.23}$$

and the 2-D scaling function as

$$\Phi_{i,j}(x,y) = \phi(x-i)\phi(y-j). \tag{3.24}$$

Here the coordinates are x and y while i,j are integer translation in $x-$ and $y-$ direction respectively. In the spectral domain, each of the wavelets and the approximation function occupies a different portion of the 2-D spectral plane. The spectral distribution of each of these four 2-D functions is shown in figure 3.7.

When we analyze a 2-D signal, the decomposition algorithm is applied to the $x-$ and $y-$ direction to generate four components from the input signal. With respect to the spectral domain, the components are called low-high (LH), high-low (HL) and high-high (HH) corresponding to the wavelets $\Psi^{[M]}_{i,j}(x,y), M = 1,2,3$ respectively. The component that corresponds to the approximation function is called the low-low (LL). The terms low- and high- refer to the processing filter; whether it is a low-pass or a high-pass filter. The diagram of 2-D $3-level$ wavelet decomposition is shown in Figure 3.8.

Let us consider a 2-D signal as a rectangular matrix of the signal value. In the case

where the 2-D signal is an image, the signal values are called the pixel values associated with the intensity of the optical reflection. Consider the input signal $c^j(m,n)$ being a $N \times N$ square matrix. We process the signal along the x direction first. That means we decompose the signal row-wise for every row using the 1-D decomposition algorithm. Because of the downsample operation, the resultant two matrices are rectangular of size $N \times \frac{N}{2}$. These matrices are transposed and they are processed row-wise again to obtain four $\frac{N}{2} \times \frac{N}{2}$ square matrices, namely, $a^{j-1}(m,n), w_1^{j-1}(m,n), w_2^{j-1}(m,n)$ and $w_3^{j-1}(m,n)$. The subscripts of the w matrices corresponding to the three different wavelets. This procedure can be repeated for an arbitrary number of times and the total number of coefficients after the decomposition always equals to the initial input coefficient N^2. An example of the decomposition is shown in figure 3.9.

If the coefficients are not processed, the original data can be recovered exactly through the reconstruction algorithm. The procedure is simply the reverse of the decomposition except the sequences are $\{g_0(k), g_1(k)\}$ instead of $\{h_0 k, h_1(k)\}$. One should remember upsample before convolution with the processing sequences.

3.2 Lifting Scheme for Discrete Wavelet Transform

Before the wavelet packet algorithms are discussed, we want to present a fast and simple lifting algorithm to compute the DWT coefficients. The two basic operations in the decomposition and reconstruction algorithms are (1) filtering followed by downsample, and (2) upsample followed by filtering. Both operations are easy to be understood but inefficient in implementation because they take up extra memory for the filters and the intermediate outputs. Recently, Swelden and Daubechies derived a new algorithm called the "lifting algorithm" that is much more efficient with high computational accuracy [10]. The lifting coefficients are wavelet-specific just like different filter coefficients correspond to different wavelets

The lifting scheme of the wavelet transform is shown in Figure 3.10.

First, the signal is subsampled into even and odd data streams. These two data streams can be considered as the approximation signal and the detail signal. A general wavelet-lifting scheme is composed of a sequence of the two basic steps: prediction step and update step. Both steps appear in the lifting sequence alternatively. For the inverse DWT, the procedure is inverted. The two basic steps are applied to the approximation coefficients before they are jointed back together as as one output data stream. It has been shown that the lifting scheme may save up to 60 percent for some wavelet filter banks. The lifting scheme diagram for inverse DWT is shown in Figure 3.11.

The development of the lifting scheme is based on a number of mathematical theories. These theories are discussed one at a time for easy of understanding by the reader.

3.2.1 Laurent polynomial

A FIR filter $h(k) = (h_M, h_{M+1} \cdots, h_{N-1}, h_N)$, $M < N$ is characterized by its $Z-$transform $H(z) = \sum_{M}^{N} h(k)z^{-k}$. The degree of a Laurent polynomial (LP) is defined by $|H(z)| = N - M.$, where $|H(z)|$ is used to represent the degree of a polynomial $H(z)$. In other words, the degree of a polynomial is determined by the length of the corresponding filter. An algebraic polynomial of one term z^K is a Laurent polynomial of degree "0". This example clarifies the new definition well. The degree 0 in LP is assigned to $-\infty$. The LP division is similar to the division in algebraic polynomials

Let $H(z)$ and $G(z)$ be two LPs. Assuming $G(z) \neq 0$ and $|H(z)| \geq |G(z)|$, there exists two Laurent polynomial $Q(z)$ and $R(z)$ such that

$$H(z) = G(z)Q(z) + R(z). \tag{3.25}$$

where $|Q(z)| = |H(z)| - |G(z)|$ is the Quotient and $R(z)$ with $|R(z)| < |G(z)|$ is the Remainder. Like the notations for algebraic polynomial division, $Q(z) = H(z)/G(z)$ represents Quotient and $R(z) = H(z)\%G(z)$ represents Remainder

A LP is said to be invertible if and only if it is a monomial. Unlike the regular polynomial

division, LP division does not hold the property of Uniqueness. This property is important for the development of the lifting scheme

The non-uniqueness of LP division is demonstrated by the following example [10]. Let $H(z) = z^{-1}+6+z$ be divided by $G(z) = 4+4z$. Using 3.25, we have $R(z) = H(z)-G(z)Q(z)$ and $|Q(z)| = 1$.It is easy to check the following two sets of $(Q(z), R(z))$ satisfying 3.25.

$$Q(z) = 1/4(z^{-1} + 5), R(z) = (z^{-1} + 6 + z) - (4 + 4z)(1/4z^{-1} + 5/4) = -4z \tag{3.26}$$

$$Q(z) = 1/4(z^{-1} + 1/4), R(z) = (z^{-1} + 6 + z) - (4 + 4z)(1/4z^{-1} + 1/4) = 4. \tag{3.27}$$

The property of non-uniqueness for LP division is important for wavelet lifting scheme.

3.2.2 Modulation Matrix and Polyphase Matrix

In perfect reconstruction filter bank literature, the modulation matrix $M(z)$ includes the decomposition filters and the reconstruction filters. It is defined by

$$M(z) = \begin{bmatrix} H(z) & H(-z) \\ G(z) & G(-z) \end{bmatrix}. \tag{3.28}$$

The dual of the modulation matrix $\widetilde{M}$ satisfies the relation

$$\widetilde{M}(z^{-1})^t M(z) = 2I \tag{3.29}$$

where I is a 2×2 identity matrix

Polyphase representation of wavelet transform: subsampele the input data based on even and odd indices followed by the dual polyphase matrix filtering. For IDWT, apply the polyphase matrix to the two data streams followed by up sampling before putting the streams together. Now, let us divide the filter$H(z)$ into the even indexed part and the odd indexed part.

$$H(z) = H_e(z^2) + z^{-1}H_o(z^2), \tag{3.30}$$

where $H_e(z) = \sum_k h_{2k} z^{-k}$ and $H_o(z) = \sum_k h_{2k+1} z^{-k}$. The polyphase representation of the filter bank is given by

$$P(z) = \begin{bmatrix} H_e(z) & G_e(z) \\ H_o(z) & G_o(z) \end{bmatrix} \tag{3.31}$$

so that

$$P(z^2)^t = 1/2M(z) \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix}. \tag{3.32}$$

The dual of the polyphase matrix $\widetilde{P}(z)$ may be constructed to satisfy $P(Z)\widetilde{P}(z^{-1})^t = I$. In the theory of filter bank design, perfect reconstruction of a signal requires the elements of the polyphase matrix $P(Z)$ to satisfy $\widetilde{H}_e(z) = G_o(z^{-1})$, $\widetilde{H}_o(z) = -G_e(z^{-1})$, $\widetilde{G}_e(z) = -H_0(z^{-1})$, and $\widetilde{G}_o(z) = H_e(z^{-1})$. These conditions are satisfied by $\widetilde{G}(z) = z^{-1}H(-z^{-1})$ and $\widetilde{H}(z) = -z^{-1}G(-z^{-1})$. The concept of Complementary is important to the lifting algorithm. Two filters (H, G) are said to be Complementary if their polyphase matrix $P(z)$ has unity determinant.

3.2.3 The Euclidean algorithm

The Euclidean algorithm is used to determine the Greatest Common Divisor (GCD) of two LP polynomials [10]. Consider two different LPs $u(z)$ and $v(z)$ with $v(z) \neq 0$ and $|u(z)| \geq |v(z)|$. The quotient polynomial $q(z)$will be of degree ≥ 0. To determine the GCD, we initialize the algorithm by writing $u_0(z) = u(z)$ and $v_0(z) = v(z)$. Assuming that ($v_i(z) \neq 0$), we construct a loop to compute the GCD of $u(z)$ and $v(z)$.

{

$i = i + 1$

$u_i(z) = v_{i-1}(z)$

$v_i(z) = u_{i-1}(z) \% v_{i-1}(z)$

}

$n = i$;

GCD$[(u(z), v(z)] = u_n(z)$

Suppose $|v_{i+1}(z)| < |v_i(z)|$, there exists an m such that $|v_m(z)| = 0$. The process stops for $n = m + 1$. The number of steps thus is limited by $n \leq |v(z)| + 1$. Let $q_{i+1}(z) = u_i(z)\%v_i(z)$.

$$\begin{bmatrix} u_n(z) \\ 0 \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} 0 & 1 \\ 1 & -q_i(z) \end{bmatrix} \begin{bmatrix} u(z) \\ v(z) \end{bmatrix}, \tag{3.33}$$

We obtains

$$\begin{bmatrix} u(z) \\ v(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_n(z) \\ 0 \end{bmatrix} \tag{3.34}$$

and $u_n(z)$ is the common factor of $u(z)$ and $v(z)$. If $u_n(z)$ is a monomial, then $u(z)$ and $v(z)$ are relatively prime. This is a characteristic of LP

Let us use the previous example as an illustration. Let $u(z) = z^{-1}+6+z$ and $v(z) = 4+4z$.

$$u_0(z) = z^{-1} + 6 + z$$

$$v_0(z) = 4 + 4z$$

$$u_1(z) = 4 + 4z$$

$$v_1(z) = 4$$

$$q_1(z) = 1/4z^{-1} + 1/4$$

$$u_2(z) = 4$$

$$v_2(z) = 0$$

$$q_2(z) = 1 + z.$$

We may expressed these factors succinctly as in the following.

$$\begin{bmatrix} z^{-1} + 6 + z \\ 4 + 4z \end{bmatrix} = \begin{bmatrix} 1/4z^{-1} + 1/4 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1+z & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix}. \tag{3.35}$$

The number of steps is $n = 2 = |v(z)| + 1$.

3.2.4 Lifting Algorithm

We will show through factorization of a pair of complementary filters (h, g) into the lifting procedure [10]. Since (h, g) is complementary, $\det(P(z)) = 1$. Moreover, $h_e(z)$ and $h_o(z)$ are relative prime. Using Euclidean algorithm, we get the GCD of Laurent polynomials $h_e(z)$ and $h_o(z)$ whose degree is 1. A constant K is used as their GCD.

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix} \tag{3.36}$$

Note that in case $|h_o(z)| > |h_e(z)|$, the first quotient $q_1(z)$ is zero. If n is odd,

$$\begin{cases} h^n = h(z)z \\ g^n(z) = -z^{-1}g(z) \end{cases}$$

The determinant of the new (h^n, g^n) is still unity. In other words, (h^n, g^n) keeps Complementary. Given a filter h, a complementary g^o can be constructed by letting

$$P^o(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_0(z) & g_o^0(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \tag{3.37}$$

From

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix}, \tag{3.38}$$

we have

$$P^0(z) = \prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}. \tag{3.39}$$

The original polyphase matrix is obtained by

$$P(z) = P^0(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}. \tag{3.40}$$

For a complementary filter pair (h, g), there always exists a Laurent polynomial $s_i(z)$ and $t_i(z)$ for $1 \leq i \leq m$ and a non-zero constant K so that

$$P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}. \square \tag{3.41}$$

The dual polyphase matrix is given by

$$\widetilde{P}(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & 0 \\ -s_i(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix}. \tag{3.42}$$

Haar basis is used as an example here. $h(z) = 1 + z^{-1}$, $g(z) = -1/2 + 1/2z^{-1}$, $\widetilde{h}(z) = 1/2 + 1/2z^{-1}$ and $\widetilde{g}(z) = -1 + z^{-1}$. After Euclidean algorithm, we have

$$P(z) = \begin{bmatrix} 1 & -1/2 \\ 1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1/2 \\ 0 & 1 \end{bmatrix} \tag{3.43}$$

$$P(z)^{-1} = \widetilde{P}(1/z) = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}. \tag{3.44}$$

The DWT lifting algorithm is described here [10].

$s_l^{(0)} = x_{2l}$

$d_l^{(0)} = x_{2l+1}$

$d_l = d_l^{(0)} - s_l^{(0)}$

$s_l = s_l^{(0)} + 1/2 d_l^{(0)}$

The $IDWT$ lifting scheme is similarly computed.

$s_l^{(0)} = s_l - 1/2 d_l$

$d_l^{(0)} = d_l + s_l^{(0)} \cdots$

$x_{2l+1} = d_l^{(0)}$

$x_{2l} = s_l^{(0)}$.

3.3 Wavelet Packets-Symmetrical Tree Algorithm

The wavelet packet basis is a natural generalization of the wavelet basis. The wavelet packet subdivides both the approximation subband and the wavelet subband into finer frequency bands. This is a refinement process in the frequency domain and the WPT is a symmetric tree algorithm. The development of the wavelet packets is based on a mathematical theorem (the splitting trick) proved by I. Daubechies in [7]. The theorem is restated here.
If $f(x-k) \mid_{k \in Z}$ forms an orthonormal basis and

$$\begin{aligned} F_1(x) &= \sum_k p_k f(x-k) \\ F_2(x) &= \sum_k q_k f(x-k), \end{aligned} \tag{3.45}$$

then $\{F_1(x-2k), F_2(x-2k); k \in Z\}$ is an orthonormal basis for $E = span\{f(x-n); n \in Z\}$. By a generalization of the two scale relation in the wavelet, we generate the wavelet packet

subspace associated with the following wavelet packet generation.

$$\begin{aligned} \mu_{2\ell}(t) &= \sum_k g_0(k)\mu_\ell(2t-k) \\ \mu_{2\ell+1}(t) &= \sum_k g_1(k)\mu_\ell(2t-k). \end{aligned} \tag{3.46}$$

For $\ell = 0$, we have μ_0 and μ_1 being the approximation function and the wavelet respectively. For $\ell = 1$, we see the wavelet packets μ_2 and μ_3 are generated from the wavelet μ_1. More wavelet packets are generated for a large integer l

This process is repeated so that many wavelet packets can be generated from the two-scale relations. Some Haar wavelet packets are shown in Figure 3.12 and 3.13.

More available basis functions allow a given signal to be represented by a selected set of bases from different levels of resolution. One may choose the "best" bases set via an optimization criterion. The best-bases and best-level are two well-known algorithms for the wavelet packet representation of a given signal.

3.3.1 Wavelet packet algorithms

The wavelet packet algorithm extends the wavelet algorithm to all subbands. In other words, the two-channel decomposition algorithm is applied to the approximation coefficients as well as the wavelet coefficients. Consequently, for every decomposition process, the number of subband is double that of the previous resolution (2^n, n is the level of decomposition.) Suppose the input discrete signal $x(n)$ has been mapped into the approximation space V_n. That is, $x\,(n) \longleftrightarrow a_{j,k}$. Using the decomposition sequences $h_0(n)$ and $h_1(n)$, we obtain the approximation function series coefficients $a_{j-1,k}$ and the wavelet series coefficients $w_{j-1,k}$. The coefficient sequences $a_{j-1,k}$ and $w_{j-1,k}$ are decomposed through $h_0(n)$ and $h_1(n)$ into four sequences

$$a_{j-2,k}$$

$$w_{j-2,k}$$

$$u_{j-2,k}$$
$$v_{j-2,k} \tag{3.47}$$

in which the last two sequences $u_{j-2,k}$ and $v_{j-2,k}$ are wavelet packet series coefficients. This process may be repeated to generate eight coefficient sequences corresponding to eight wavelet packet components of the original signal $x(n)$. The decomposition and reconstruction block diagrams are given in Figure 3.14 and 3.15.

An example of a one-dimensional signal decomposition is shown in figure 3.16. A two-dimensional extension of the wavelet packete algorithms is very straightforward. The two-dimensional wavelet algorithm is applied to any band (low-low, low-high, high-low, high-high) at any level. The total number of sub-image is 4^n.

3.4 Markov Random Field

Markov Random Field (MRF) has been recognized as a powerful technique for image segmentation and noise removal [4][9]. One of its most useful functions is to retrieve the original (undistorted) signal from noisy data. The *Maximum a posterior* (MAP) estimation technique is used to achieve segmentation in the image plane. The MRF is particularly useful for speckle noise in SAR images. Take the images in Figure (3.17) for example, the goal of applying the MRF to the observed noisy image is to obtain a resultant image with the maximum probability of being the original image.

Let the observed signal be a one-dimensional vector $Y = \{y_i\}$, $i = 1, 2, \cdots, W \times H$, W and H are the width and height of an image if the signal is in two dimensions. The MRF segments the signal to produce an output $X = \{x_i\}$, $i = 1, 2, \cdots, W \times H$, The output signal consists of K different labels. The notation $x_i = k$ means the pixel at i belongs to label k (or the k^{th} class) in the resultant image X. The label index $k \in 1, 2, \cdots, K$ and K is the total number of distinct labels in the segmented image. Two additional symbols used in this discussion are (1) $X_{|x_i}$ denotes all labels of image X except the single pixel x_i and (2) X_{N_i} denotes the neighborhood pixels around pixel x_i. Two constraints are placed on the

probability of the pixels in the resultant image. That forms the important features of the MRF.

$$P(X) > 0, \qquad \text{(positivity)} \tag{3.48}$$

$$P(x_i|X_{|x_i}) = P(x_i|X_{N_i}) \qquad \text{(Markovianity)}. \tag{3.49}$$

The positivity guarantees that all probability is greater than zero and the Markovianity shows that the probability associated with a given pixel depends only on its neighborhood pixels.

Neighborhood System and Clique

The *neighborhood system* and *cliques* are often used in MRF. We use Figure 3.18 to describe the first order neighborhood system, second order neighborhood system and fifth order neighborhood system for a particular pixel x_i located at the center. Figure3.18(c) is a representation of a fifth order neighborhood system. In this figure, the numbers $n = 1, 2, \cdots$ in each neighborhood system indicate the outermost neighboring pixels of the n^{th} order neighborhood system.

A clique is a subset of pixels within a neighborhood system. They are grouped together to be called single-pixel clique, two-pixel cliques and so on according to their sizes. (see Figure3.18) They are denoted by symbols $C_1,\ C_2,\ C_3$, and defined mathematically as

$$C_1 = \{x_i | x_i \in X\}, \tag{3.50}$$

$$C_2 = \{\{x_i, x_{i'}\} | x_{i'} \in X_{N_i},\ x_i \in X \text{are neighbors to each other}\} \tag{3.51}$$

$$C_3 = \{\{x_i, x_{i'}, x_{i''}\} | x_{i'}, x_{i''} \in X_{N_i},\ x_i \in X \text{are neighbors to each other}\}. \tag{3.52}$$

where X is the set of all pixels in the image, x_i is the i^{th} center pixel of a given neighborhood system N_i, $x_{i'},\ x_{i''}$ are pixels within that neighborhood system, and X_{N_i} is the set of all pixels in the neighborhood system. Figure 3.19 gives the picture of single-pixel cliques (a), double-pixel cliques (b), triple-pixel cliques (c) and quad-pixel cliques (d) for

a second-order neighborhood system. The second-order neighborhood system is adequate in most cases. The processing efficiency may be further increased if one includes only the single-pixel cliques and double-pixel cliques, as shown in Figure 3.20 from a second-order neighborhood system for computation. The clique α is single-pixel clique, β_1 and β_2 are horizontal and vertical cliques, while β_3 and β_4 are diagonal cliques. We remark that the pixels in a clique are ordered. Clique $\{x_i, x_{i'}\}$ is not the same as clique $\{x_{i'}, x_i\}$.

Basic Concepts

The Hammersley-Clifford theorem states that the probability of X has a Gibbs distribution given by [1][2].

$$P(X) = Z^{-1} e^{-\frac{1}{T}U(X)} \tag{3.53}$$

$$Z = \sum_{x \in X} e^{-\frac{1}{T}U(X)} \tag{3.54}$$

$$U(X) = \sum_{c \in C} V_c(X), \tag{3.55}$$

where Z is a normalization constant to make $P(X)$ less than unity. T is called the *temperature* which is set to unity, and $U(X)$ is equal to the sum of all clique potentials $V_c(X),\ c \in C$ for all possible cliques C.

$$U(X) = \sum_{\{i\} \in X} V_1(x_i) + \sum_{\{i,i'\} \in C_2} V_2(x_i, x_{i'}) + \sum_{\{i,i',i''\} \in C_3} V_3(x_i, x_{i'}, x_{i''}) + \cdots. \tag{3.56}$$

The potential function $V_c(X), c \in C$ depends on the clique configuration. The potential function for two-pixel cliques is defined as:

$$V_2(x_i, x_{i'}) = \begin{cases} -\beta_{i'} & \text{if } x_i = x_{i'} \text{ and } i, i' \in C_2 \\ \beta_{i'} & \text{if } x_i \neq x_{i'} \text{ and } i, i' \in C_2 \end{cases} \tag{3.57}$$

where $\beta_{i'} > 0$ is a constant associated with pixel $x_{i'}$. The potential function of single-pixel clique is defined as:

$$V_1(x_i) = \alpha_k \text{ if } x_i = k \text{ and pixel } i \in X \tag{3.58}$$

where α_k is a constant associated with label k. The constant is different for a different label k in a heterogeneous system.

Principles of Operation

The Bayes' theorem on the conditional probability states that

$$P(X|Y) \propto P(Y|X)P(X) \tag{3.59}$$

where $P(X)$ is the *a priori* probability of the image X, and $P(Y|X)$ is the conditional probability of the observed image Y given the image X. The principal goal is to maximize the *a posteriori probability* mass function for X which, according to equation 3.53, can be expressed as:

$$P(X|Y) = Z^{-1}e^{-U(X|Y)}. \tag{3.60}$$

The corresponding energy function in equation 3.60 is obtained:

$$U(X|Y) = U(X) + U(Y|X), \tag{3.61}$$

where $U(X)$ can be computed via equation 3.56. Maximization of $P(X/Y)$ is the same as minimization of $U(X/Y)$. Assuming additive Gaussian noise so that the observed image Y is the sum of the original image X and an independent Gaussian noise $\{e_i, \quad 0 < i < W \times H, \quad e_i \sim N(x_i,$

From equation 3.56, minimization $U(X)$ is the same as minimizing of $V_2(x_i, x_{i'})$. The energy function is increased when two pixels in the same clique belong to different labels. To reduce the energy toward a minimum, one should incline to set the same label on pixels along a given direction. If the pixels in a label do not have directional characteristics, the $\beta_i's$ are set to be the same in the second order neighborhood system to minimize the energy function.

In order to achieve the MAP estimation for MRF, several iterative algorithms have been proposed. In [3], Geman proposed the Simulated Annealing (SA) algorithm to find the MAP estimation of the true image by minimizing the energy function $U(X|Y)$. The computational

cost is high since the algorithm optimizes the probability for every point in the image. The Iterated Conditional Modes (ICM) is more computationally feasible. Besides the SA and ICM, there are also the Maximum of Posterior Marginals (MPM), the Dynamic programming methods (DPM) and the Hierarchical Algorithm. Only the first two algorithms are outlined briefly here.

3.4.1 Simulated Annealing algorithm

SA is based on the classical Metropolis method of simulating systems containing large numbers of particles. It guarantees to find a global optimum for any labelling problems. The steps of SA are given as follows:

1. Initiate with a temperature T.

2. Initialize X by maximizing $P(y_i|x_i)$ for each pixel i. This is the maximum-likelihood estimation (MLE) of pixel label. One may use the K-mean clustering algorithm for this step.

3. Perturb X and compute $Diff = U(X|Y) - U(X'|Y)$. If $Diff > 0$, then replace X by X', otherwise replace X by X' with probability $e^{\frac{Diff}{T}}$.

4. Repeat step 3 N_{inner} times.

5. Replace T by f(T) where $f(.)$ is a decreasing function.

6. Repeat step 3 to step 5 N_{outer} times.

$f(.)$ is the cooling schedule initialized by temperature T, the inner iteration time N_{inner}. The outer iteration time N_{outer} and the perturbation method in step 3 are defined by experienced users. In order to assure the algorithm moving toward a global maximum, the temperature is changed very slowly. (step size may be as little as one percent). For the perturbation from X into X', a single pixel assigned with a random label.

3.4.2 Iterated Conditional Mode

The ICM uses essentially the same approach as in SA. However, it makes use of the Markovianity of the system so that only the pixels in the neighborhood system need to be considered. It computes the local energy function $U(X|Y)$. Given an observed image Y, the algorithm sequentially updates each x_i from x_i^k to x_i^{k+1} by maximizing the local $P(x_i|X_{|x_i}, Y)$ instead of the global $P(X|Y)$. To be more specific, recall that X_{N_i} is defined as the set of pixels in the neighborhood of x_i, then

$$P(x_i|X_{|x_i}, Y) = Z^{-1} \exp(-U(x_i|X_{|x_i}, Y)) \tag{3.62}$$

$$P(x_i|X_{|x_i}, Y) \infty P(y_i|x_i) P(x_i|X_{N_i}) \tag{3.63}$$

Hence, we have $U(x_i|X_{|x_i}, Y) = U(y_i|x_i) + U(x_i|X_{N_i})$ where C in Equation 3.56 defines maximal size of the clique for the given neighborhood system and only pixels within the cliques of x_i are considered for the computation of energy function.

The ICM is simple. It is described by these steps:

1. Use k-mean algorithm to initialize X for segmentation.

2. Update x_i to a new value $x_{i'}$ that minimizes $U(x_i|X_{|x_i}, Y)$ for every pixel of the image.

3. Repeat step 2 N_{outer} times.

3.5 Artificial Neural Networks

The nonlinear properties of an Artificial Neural Network (ANN) are ideal for pattern recognition or texture classification. Successes in pattern analyses using ANN have been demonstrated in many disciplines. [**?**] [**?**] [**?**]. We introduce four types of neural classifiers: Multilayer Perceptron (MLP), Support Vector Machine (SVM), Radial Basis Function (RBF) and Self Organizing Map (SOM). For these four classifiers, we focus primarily on the first two. The MLP is the most popular technique using features from the signal groups for training and

testing. Standard back-propagation algorithm with bias and momentum is the method of training. To overcome the problem of slow convergence, we use the genetic algorithm (GA) for optimizing the feature selection to reduce the feature dimension. GA will be discussed in greater detail in a later chapter. Instead of reducing the input feature dimension, an SVM nonlinearly maps an input vector into a high-dimensional feature space in which optimal hyperplanes for separating these mapped features are constructed. Intuitively, an SVM converges very fast compared to traditional neural networks and potentially very useful for pattern classification. An SVM with a RBF kernel is a very versatile tool, well suited for both pattern recognition and regression applications. The RBF network and the SOM will be discussed at the end of this section.

Multilayer Perceptron

In the 1960s, a complicated multilayer perceptron model was developped based on Single Layer Perceptron (SLP). Theoretically, the MLP may be used to approximate a function to arbitrary degree of accuracy. However, there was no training method available in complicated applications. The back-propagation algorithm was introduced in 1980.

A perceptron, like a neuron cell, is the basic building block of an MLP neural network. The most widely used neuron model is based on McCulloch and Pitt's work [**?**]. Figure 3.21 shows a typical structure of a perceptron where the input vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is weighted by the weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)$. The weighted input are linearly summed to produce the output by

$$y(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x} + w_0) \tag{3.64}$$

where w_0 is the *bias* and $g(\cdot)$ is the *activation function*. Several activation functions have been proposed. The popular ones are:

1. Threshold Function

$$g(u) = \begin{cases} 1 & u > t \\ 0 & u \leq t \end{cases} \tag{3.65}$$

2. Linear Function

$$g(u) = a \cdot u + b \tag{3.66}$$

3. Sigmoidal Function

$$g(u) = \frac{1}{1 + e^{-\alpha u}}. \tag{3.67}$$

A perceptron with a threshold activation function forms a linear separation model for two class classification. The bias value w_0 serves as a threshold for the classification. A multi-class problem requires more perceptrons to form an SLP. The SLP concept is extended by using one perceptron model $y_m(\mathbf{x})$ for each class C_m in a multi-class problem.

$$y_m(\mathbf{x}) = g(\mathbf{w}_m^T \mathbf{x} + w_{m0}). \tag{3.68}$$

The decision boundary for separating class C_m from class C_j is given by $y_m(\mathbf{x}) = y_j(\mathbf{x})$ which, for linear discriminants, corresponds to a hyperplane of the form

$$(\mathbf{x}_m - \mathbf{x}_j)^T \mathbf{x} + (w_{m0} - w_{j0}) = 0. \tag{3.69}$$

Minsky and Papert [**?**] have shown mathematically that the SLP cannot solve all problems. However, one may approach problems with more SLP layers. This type of network is called MLP. It has input, hidden layer and the output layers. There may be more than one hidden layer and they contain many generalized perceptrons. A typical MLP network is shown in Fig. 3.22. It is shown that the structure, the activation function and the weight combine to form a nonlinear mapping between the inputs and outputs.

Based on the MLP shown in Fig. 3.22. the output h_k of the kth hidden neuron is given by

$$h_k = \varphi\left(\sum_{n=1}^{N} x_n w_{nk}\right), \tag{3.70}$$

where $\varphi(\cdot)$ is the activation function of the hidden layer. For the same way, the lth output neuron o_l is computed by:

$$o_l = \phi\left(\sum_{k=1}^{K} h_k v_{kl}\right) = \phi\left(\sum_{k=1}^{K} \varphi\left(\sum_{n=1}^{N} x_n w_{nk}\right) v_{kl}\right), \tag{3.71}$$

where $\phi(\cdot)$ is the activation function of the output layer. Equation 3.70 and 3.71 give the formulas for computing the network outputs. The learning procedure is to train the weights via the *back-propagation* that brought the MLP into popularity. Since the BP algorithm makes the MLP one of the most widely used networks, even sometimes being called a BP network. The BP algorithm computes the output of the network from the input layer after an initial assignment of weights and the second stage updates the weights based on the output error. This procedure is repeated until the MLP reaches a stopping criterion. Various search techniques are available for weight adjustment. The gradient search proves to be the simplest way. The mathematical details are provided in [**?**] [**?**]. The algorithm is succinctly summarized here.

1. Randomly initialize the weights with small values if no prior knowledge is available.
2. Set up an input vector.
3. Compute the network output.
4. Compute the output error.
5. Adjust the weights in the gradient direction to reduce the output error.
6. Repeat 2-5 until the error reaches a pre-determined small value.

One should maximize the advantages offered by MLP when it is chosen to solve a problem. The advantages include: (1) The training sample distribution is not restricted. This increases the adaptivity of the MLP. (2)The number of free parameters is increased with the number of hidden units for complex mappings. (3) Bounded activation functions are well behaved. One should avoid using the MLP if time of convergency is an issue or if under- or over-fitting may occurs.

Support Vector Machine

Support Vector Machine (SVM) takes a different approach than the MLP. Its ability for effective classification has been well recognized. Unlike the MLP which reduces the input dimension for increasing efficiency, an SVM maps nonlinearly an input vector into a high-dimensional *feature space* that is hidden from the input and the output. The user constructs an optimal hyperplane in the feature space that separates the mapped features into two classes to achieve classification. SVM was originally constructed for two-class problems. Multi-class SVM will be discussed in Chapter 6. An SVM is different from a conventional neural network in several ways. (1) Method of classification. An SVM minimizes the probability of misclassifying a randomly drawn data point from an unknown probability distribution. (2) Efficiency and effectiveness. An SVM uses support vectors to carry information of the most important features. Researchers [[**?**],[**?**]] have shown that only three to five percent of the data points are used in the support vectors. (3) SVMs depend on feature information not on large number of data [**?**]. We use a simple two-class example to illustrate the SVM method. Let us assume that we have some training sample sets $\{\mathbf{x}_i, \mathbf{y}_i\}, i = 1, \ldots, l$. Each training sample $\mathbf{x}_i \in \Re^d$, d being the dimension of the input space. Each point $\mathbf{x}_i$ belongings to a class labelled by $\mathbf{y}_i \in \{-1, +1\}$. The goal of two-class classification is to define a hyperplane that separates the positive from the negative samples. The point $\mathbf{x}$ which lies on the hyperplane satisfies $\mathbf{w} \cdot \mathbf{x} + b = 0$, where $\mathbf{w}$ is a vector normal to the hyperplane. In the linear case, the algorithm looks for the separating hyperplane with the largest margin from the nearest points on both sides. In other words, we wish to find W.

$$y_i \left(\mathbf{x}_i \cdot \mathbf{w} + b\right) - 1 \geq 0 \qquad \forall i. \tag{3.72}$$

If a hyperplane satisfying (3.72) is found, the set is linearly separable. Given a linearly separable set $\mathbf{S}$, the optimal separating hyperplane (OSH) is the one for which the distance γ to the closest points of $\mathbf{S}$ is maximized.

$$\gamma = \frac{1}{2}\left(\frac{\mathbf{w} \cdot \mathbf{x}^+}{\| \mathbf{w} \|^2} - \frac{\mathbf{w} \cdot \mathbf{x}^-}{\| \mathbf{w} \|^2}\right) - 1 \geq 0 \qquad \forall i. \tag{3.73}$$

The problem can be equivalently stated as:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}\mathbf{w}\cdot\mathbf{w} \\ \text{Subject to} & y_i(\mathbf{x}_i\cdot\mathbf{w}+b-1)\geq 0 \qquad i=1,\ldots,l \end{array} . \tag{3.74}$$

Lagrange multipliers are often used to find the solution. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_l)$ be the l positive Lagrange multipliers of the inequality constraints in (3.72). The problem is formulated as finding the saddle point of the function

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{l}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1]. \tag{3.75}$$

Making the gradient of $L(\mathbf{w}, b, \alpha)$ with respect to $\mathbf{w}$ and b to vanish generates two equations:

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \tag{3.76}$$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0. \tag{3.77}$$

After resubstitution we obtain two equivalent equations:

$$\sum_{i=1}^{l} y_i\alpha_i x_i = \mathbf{w} \tag{3.78}$$

$$\sum_{i=1}^{l} y_i\alpha_i = 0. \tag{3.79}$$

Substitution of (3.78) and (3.79) into (3.75) yields

$$L(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j < \mathbf{x}_i\cdot\mathbf{x}_j > . \tag{3.80}$$

The linear support vector training is reduced to

$$\begin{aligned} &\text{Maximize} \quad L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j < \mathbf{x}_i \cdot \mathbf{x}_j > \\ &\text{subject to} \quad \sum_{i=1}^{l} y_i \alpha_i = 0, \alpha_i \geq 0 \end{aligned} \tag{3.81}$$

.

The optimal solutions $\bar{\alpha}, (\bar{\mathbf{w}}, \bar{b})$ must satisfy the following conditions: [**?**] [**?**]

$$\bar{\alpha_i}[y_i(\bar{\mathbf{w}} \cdot \mathbf{x}_i + \bar{b}) - 1] = 0, \qquad i = 1, \ldots, l. \tag{3.82}$$

Note that the $\bar{\alpha_i}$ that may be nonzero in Equation (3.82) are those for which the constraints Equation (3.72) are satisfied with the equality sign. All other $\bar{\alpha_i}$ are zero. This means that the vector $\mathbf{w}$ is a linear combination of a small percentage of the points $\bar{\mathbf{x}_i}$. These are the *support vectors* that lie on the hyperplane. The problem of classifying a new data point $\mathbf{x}$ is now solved by examining the value of the hyperplane decision function

$$f(x) = sgn\left(\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{b}\right). \tag{3.83}$$

This analysis may not produce a solution for non-separable data. The non-separable cases in SVM allow training error repeated by $\sum_i \xi_i$ where ξ_i are called slack variables. If $0 < \xi_i < 1$, the *ith* data point is correctly classified. Otherwise, it is given by $\xi_i > 1$. The problem may be reformulated to $\|\mathbf{w}\|^2/2 + C\left(\sum_i \xi_i\right)^k$, where C is a user defined cost parameter. A large value for C corresponds to assigning heavier penalty to classification errors. The generalized OSH is the solution to the reformulation.

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{l} \xi_i \\ &\text{subject to} \quad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i \qquad i = 1 \ldots l \end{aligned} \tag{3.84}$$

Following the sane Lagrange multiplier method yield

$$L(\mathbf{w}, b, \xi, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i - \sum_{i=1}^{l}\alpha_i[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i] - \sum_{i=1}^{l}\mu_i\xi_i, \tag{3.85}$$

where μ_i are the Lagrange multipliers introduced to force the ξ_i to be positive. To generalize this approach to a nonlinear decision function, a simple mapping is needed. We use an XOR problem to illustrate the principle. Consider two points (1,1) and (-1,-1) that belong to the positive class while (1,-1) and (-1,1) belong to the negative class. These points are not linearly separable in $\Re^2$. However, if a mapping

$$\Phi(x) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{pmatrix} \tag{3.86}$$

is used on these points, the two positive points are mapped to (1,1,1) and the two negative points are mapped to (1,-1,1). These two classes are linear separable in $\Re^3$. Hence, the main idea of nonlinear support vector machine is to define a mapping to transform the given samples to some higher dimensional space so that the problem becomes linear and separable. The only appearance of the data in the formulation is in the scalar product $\mathbf{x_i} \cdot \mathbf{x}_j$. The training algorithm depends only on the data through $\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$. If we choose a "kernel function" $\mathbf{K}$ such that $\mathbf{K}(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x_j})$, we only need to use $\mathbf{K}$ in the training algorithm. Some Kernels that have been used in the literature are shown in the following Table 3.1.

Once a kernel $\mathbf{K}$ has been chosen, the objective is to maximize

$$L(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{K}(\mathbf{x_i}, \mathbf{x_j}) \tag{3.87}$$

and the decision function becomes

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{3.88}$$

A nonlinear SVM structure is shown in Fig. 3.24. The vectors $\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x}_\ell$ in the hidden layer are the selcted support vectors. A more thorough analysis on extension to multi-class procedure choosing the kernel function and training an SVM is found in [**?**] [**?**].

When we encounter a multi-class classification problem, we need to choose an appropriate approach to extend SVM for multi-class problem. There are two ways to approach a multi-class problem. The "one-versus-rest method" and the "one-versus-one method." For a k -class classification problem, we can construct k independent SVMs so that each one of these k classes is trained to be separate from all the rest. When the system is tested for classification, each sample is input into all the SVMs. the one SVM with the largest output indicates the class where the input data belongs. This is the one-versus-rest method. The one-versus-one method requires $k(k-1)/2$ SVMs. The decision on a data under test is based on the voting results from all these SVMs. The fundamentals of these two methods are the same and according to a comparative study [**?**], We remark here that SVM that works well for two-class problems may not work as well for multi-class problems. We may use an output layer to linearly combine the SVMs and use weight factors to adjust the performance of the entire system. However, even if we can compare the outputs with the target values for a given input sample data, there is no way to make use the error to adjust the weights. SVM does not have work as an MLP. Hence, we have a need to develop certain schemes to adjust the training parameters of an SVM from the difference of the output compared to the target value. The genetic algorithm optimization is one approach to solve this problem. The natural choice for the fitness function is the classification accuracy.

Radial Basis Function

The Radial basis function neural network has been recognized as an effective feed forward nonlinear neural networks for classification problems. Radial basis functions may also be

used as kernel functions in the SVM network discussed in the last section. An RBF network has three layers: the input layer, the hidden layer and the output layer. The input layer is the same as in an MLP. The hidden layer consists of radial basis function neurons. Network nodes in the output layer combine the weighted outputs from the RBF neurons. The output of the network is given by

$$y = w_0 + \sum_{i=1}^{k} w_i \phi(\| \mathbf{x} - \mathbf{c}_i \|), \tag{3.89}$$

where $\phi(\cdot)$ is the radial basis function, $w_i, i = 1, 2, \ldots, k$ are the output layer weights, w_0 is the bias, $\mathbf{x}$ is the input vector to the network, $\mathbf{c}_i s$ are the centers associated with the basis functions and k is the total number of hidden neurons. A radial basis function is a nonlinear function that is in the form of $\phi(\mathbf{x}) = \phi(\| (\mathbf{x} - \mathbf{c} \|)$ where the $\|\cdot\|$ is the norm of the vector. The function is symmetric with respect to $\mathbf{c}$, and " $\mathbf{c}$ " is called the center of the RBF. Several typical RBFs include:
Gaussian function

$$\phi_g(x) = e^{\frac{-(x-c)^2}{\rho^2}} \tag{3.90}$$

where ρ is the "radius" of the function.

Multiquadric

$$\phi_M(x) = \left[(x-c)^2 + \rho^2\right]^{\frac{1}{2}}. \tag{3.91}$$

Inverse multiquadric

$$\phi_I(x) = \left[(x-c)^2 + \rho^2\right]^{-\frac{1}{2}} / \rho. \tag{3.92}$$

When a data vector is input to an RBF network, each hidden neuron will output a value depending on the relative location of the data with respect to the center of the neuron. The output value is near zero if the input value is away from the center of the RBF neuron. On the other hand, the output value is near 1 if the input value is close to the center. The output is a linear weighted sum of the RBF outputs. By writing the algorithm in matrix

formulation and using the Gaussian as the RBF, we get the following equations

$$\mathbf{Y} = \mathbf{R}^t \mathbf{X} \tag{3.93}$$

where

$$\begin{aligned} \mathbf{Y} &= [y_1, ..., y_m]^t \\ \mathbf{Z} &= [\mathbf{z}_1^t, ..., \mathbf{z}_m^t]^t \\ \mathbf{z}_j &= [1, z_{j1}, .., z_{jk}]^t \\ z_{ji} &= e^{\frac{-\left\|\mathbf{x}_j - \mathbf{c}_i\right\|^2}{\rho_i^2}}. \end{aligned} \tag{3.94}$$

Figure 3.25 shows the structure of an RBF neural network.

The RBF network design is to select the basis function and the parameters associated with the neurons in the hidden layer. One simple design is to use all the training samples as hidden neurons' centers. It means they represent all samples (including the testing samples that had never been used before) very well. However, in most applications, this scheme suffers from over-fitting. Other schemes include adaptively changing the number of hidden layer neurons during the training process.

It has been shown by Park and Sandbery [**?**] [**?**] that radial basis function networks have universal approximation capabilities. This property ensures that RBF networks have at least the same theoretical capabilities as MLP networks with sigmoidal activation functions. Since MLP and RBF are capable of universal approximation, the choice is generally based on classification performance for a particular application.

Self-Organizing Maps

The Self-Organizing Map (SOM) was introduced by Kohonen as a technique for signal classification. The model is based on theories in neurobiology and the network function resembles the working principles of a human brain. Although its structure is easy to use, it has not been analyzed as well as that in MLP or RBF networks. SOM is a type of unsupervised

network. It is useful in clustering analysis and signal compression. The SOM model has only two layers, the input layer and the competitive layer. The dimension and arrangement of the nodes in the input layer are the same as the input vectors. The competitive layer can be one-dimensional or two-dimensional. The weight vectors connect the input layer to all the competitive nodes. Figure 3.26 shows the basic structure of an SOM neural network.

The training process is quite simple. For every input vector, all the neuron outputs in the competitive layer are computed. The one with the highest value is the winner. All the weights connected to the winner and its neighbors are updated. The algorithm is summarized as follows:

1. Choose random values for the initial weight vectors $\mathbf{w}_j(0)$. The only restriction here is that all the weights $\mathbf{w}_j(0)$ $j = 1, 2, \ldots, N$, are distinct with N being the number of neurons in the competitive layer. Similar to the initialization process of MLP, it is desirable to keep the magnitudes of the weights small.

2. Introduce a sample $\mathbf{x}$ into the network. Calculate the neuron values in the competitive layer and get the winner with respect to the minimum Euclidean distance criterion.

3. Find the neurons that need to be updated. Typically, they are the neurons in the neighborhood of the winner and the winner itself. For a two-dimensional competitive layer, the neighborhood area may be a rectangular region or a hexagonal region. The neighborhood area should contain only the nearest neighbors of the winning neuron at the start of the convergence phase, and may eventually shrink to one or no neighboring neurons.

4. Adjust the synaptic weight vectors of all neurons by using the update formula

$$\mathbf{w}_j(l+1) = \mathbf{w}_j(l) + \eta(l)(\mathbf{x}(l) - \mathbf{w}_j(l)) \tag{3.95}$$

where $\eta(l)$ is the learning-rate parameter. For good statistical accuracy, $\eta(l)$ should be maintained at a small value (0.01 or less) during the convergence which may take thousands of iterations and a large amount of time.

5. Return to step 2 until no noticeable changes in the feature map are observed.

After training, the network will cluster different classes of samples to different regions of the competitive layer. If the winner of a testing sample is located in the area where the training samples from the same class are clustered by the network, the network is well-trained. Otherwise, it may be under-or-over trained. The SOM works extremely well for some applications though they are not analyzed thoroughly in mathematics. However, for some applications it may oscillate or not converge if the training parameters are not well initialized and adjusted.

3.6 Anisotropic Diffusion

In the last two decades, a diffusion method has been used for image denoising and restoration. The essential idea of this approach is quite simple: the observed image is embedded into a family of derived images $X(h, w, t)$ obtained by convoluting the original image with a Gaussian Kernel $G(h, w, t)$ of variance t :

$$X(h, w, t) = X^0(h, w) * G(h, w, t). \tag{3.96}$$

The large value of t, the scale-space parameter, corresponds to images at coarser resolution. It has been proved that the diffusion equation is the solution of the heat conduction equation (3.96).

$$X(h, w, t) = \triangle X = (X_{hh} + X_{ww}). \tag{3.97}$$

With the ordinal image as the initial condition $I(h, w, 0) = X^0(h, w, t)$, the diffusion approach does not treat edge pixels and intra-region pixels differently. It is an isotropic diffusion and blurs the image very badly when the noise is removed. A good diffusion approach must satisfy the following requirements [5]:

1. *Causality*: No spurious detail should be generated during diffusion.

2. *Immediate Localization:* During the diffusion process, the edge should be sharp, not

blurred.

3. *Piecewise Smoothing:* During the diffusion process, the intraregion smoothing should occur preferentially over interregional smoothing.

For example, in an image composed of trees and background, the leaf regions should be collapsed into a treetop before being merged with the background. Perona and Malik proposed their anisotropic diffusion equation approach by adding a conduction function or diffusion function in Equation (3.98). The anisotropic diffusion is written as

$$\frac{\partial X}{\partial t} = div(c(h,w,t)\nabla X\ ) = c(h,w,t)\Delta X,\ +\nabla c \cdot \nabla X \tag{3.98}$$

where $div()$ is the divergence operator, ∇ is the gradient and Δ is the Laplacian operator, with respect to the space variable. When $c(h,w,t) = 1$, Equation (3.98) reduces to the isotropic heat diffusion equation (Equaiton 3.96). Supposing that the edges of the image are known at iteration t, the smoothing within a region is expected to be strongly encouraged, and the smoothing across the boundary is expected to be strictly restricted. This can be achieved by setting $c(h,w,t)$ 1 within a region and setting 0 at the edge. However, we do not know the edge location in the image. Hence, anisotropic diffusion should locate the edge and carry out the smoothing process. Since the image gradient operator ∇ is a very simple edge estimation, the conduction function $c(h,w,t)$ can be defined based on ∇X.:

$$c(h,w,t) = c(||\nabla X(h,w,t||). \tag{3.99}$$

To satisfy the requirement of conduction function,

$$\begin{aligned} c(x) &= 1 \text{ if } x = 0, \\ c(x) &\rightarrow 0 \text{ if } |x| \rightarrow \infty. \end{aligned} \tag{3.100}$$

Two conduction functions

$$c(x) = \exp(-(\frac{x}{k_x})^2) \text{ and} \tag{3.101}$$

$$c(x) = \frac{1}{1 + (x/k_x)^2} \tag{3.102}$$

have been proposed by Malik and Perona in [**?**]. Luis Alvarez proposed another nonlinear parabolic differential equations for image denoising [6].

$$\frac{\partial X}{\partial t} = c(|G * \nabla X\ |)|\nabla X\ |div(\frac{\nabla X}{|\nabla X|}), X(h, w, 0) = X_0(h, w), \tag{3.103}$$

where $X_0(h, w)$ is the observation, and $X(h, w, t)$ is the evolving image at time t. G is a smoothing kernel (for instance, a Gaussian); $G * \nabla X$ is therefore a local estimate of ∇X for noise removal. $C(x)$ is a conduction function. The term $|\nabla X\ |div(\frac{\nabla X}{|\nabla X|})$ represents a degenerate diffusion term, which diffuses U in the direction orthogonal to its gradient $|\nabla X\ |$ and does not diffuse at all in the direction of $|\nabla X\ |$.

References

[1] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Royal Stat. Soc.*, Ser. B, vol. 36, pp. 192-236, 1974.

[2] R. Kindermann and J. L. Snell, *Markov random fields and their applications.* Providence, R.I.: Mathematical Society, 1980.

[3] S. Geman and D. Geman, "Stochastic relaxation: Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.6, pp. 721-741, 1984.

[4] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 39-55, Jan. 1987.

[5] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, No. 7, July 1990.

[6] L. Alvarez, P. Louis and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion II," *SIAM Journal Numer. Anal.* vol. 29, No. 3, pp. 845-866, June 1992.

[7] I. Daubechies, "Ten Lectures on wavelets," SIAM, Capital City Press, Montpelier, Vermont, 1992.

[8] S. Li, "Markov random field modeling in computer vision," Springer-Verlag Inc., 1995.

[9] M. Malfait and D. Roose, "Wavelet-based image denoising using a markov random filed a priori model," *IEEE Trans. Image Processing*, vol. 6, No.4, pp.549-565, April 1997.

[10] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," Nov. 1997.

[11] J. Goswami and A. Chan, "Fundamentals of wavlelets," John Wiley & Sons, Inc., 1999.

Table 3.1: Kernel functions for different pattern classification

Kernel Function	Type of Classifier
$K(x,y) = exp(-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2))$	Gaussian RBF
$K(x,y) = (1+\mathbf{x}\cdot\mathbf{y})^d$	Polynomial of degree d
$K(x,y) = tanh(\mathbf{x}\cdot\mathbf{y}-\theta)$	Multi Layer Perceptron

Table 3.2: Percentage heroin measurements in nine illicit heroin preparations.

Sample	% Heroin			
1	2.2	2.3	2.2	2.3
2	8.4	8.7	2.2	2.3
3	7.6	7.5	2.2	2.3
4	11.9	12.6	2.2	2.3
5	4.3	4.2	2.2	2.3
6	1.1	1.0	2.2	2.3
7	14.4	14.8	2.2	2.3
8	21.9	21.1	2.2	2.3
9	8.8	8.4	2.2	2.3

Figure 3.1: Decimation by M.

Figure 3.2: Interpolation by M.

Figure 3.3: Convolution followed by decimation

Figure 3.4: Interpolation followed by convolution

Figure 3.5: Two channel filtering and reconstruction

Figure 3.6: Two channel filtering and reconstruction

Figure 3.7: Regions on the 2-D spectral plane occupied by the 2-D scaling function and wavelets

Figure 3.8: 2-D 3-level wavelet

Figure 3.9: 3-level 2-D wavelet decomposition of a SAR image

Figure 3.10: Forward DWT using lifting scheme

Figure 3.11: Inverse DWT using lifting scheme

Figure 3.12: Haar wavelet packet(1-4).

Figure 3.13: Haar wavelet packet (5-8).

Figure 3.14: Wavelet packet decomposition

Figure 3.15: Wavelet packet reconstruction

Figure 3.16: Wavelet packet decomposition of a 1-d signal

Figure 3.17: The MRF relationship between the observed noisy image and the original unpolluted image

Figure 3.18: The neighborhood systems defined in Markov Random Field. (a) The first order neighborhood system. (b) The second order neighborhood system. (c) The third order neighborhood system.

Figure 3.19: The cliques for second order neighborhood systems. (a) single-pixel clique. (b) double-pixel cliques. (c) triple-pixel cliques. (d) quad-pixel clique.

Figure 3.20: The single-pixel clique and double-pixel cliques for the second order neighborhood system. (a) The second order neighborhood system. i is the center of this neighborhood system, and the other pixels are labelled from 1 to 8. (b) The single-pixel clique, which is the pixel x_i itself. (c) The double-pixel cliques which consist of x_i with another pixel $x_{i'}$. Each of them have a weight $\beta_{i'} = \beta_1 \sim \beta_4$ associated with it.

Figure 3.21: Generalized perceptron structure

Figure 3.22: Multiplayer perceptron structure

Figure 3.23: Hyperplane for linear case

Figure 3.24: Nonlinear support vector machine structure

Figure 3.25: Radial basis function neural network.

Figure 3.26: Self-Organizing Map Model

Index