# APPLICATIONS OF WAVELET TRANSFORMS IN PATTERN RECOGNITION AND DE-NOISING

GUANGYI CHEN

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

JANUARY 1999

0-612-43552-0

# Abstract

## Applications of Wavelet Transforms in Pattern Recognition and De-noising

Guangyi Chen

In this thesis, we study the application of wavelet transforms in two important areas: pattern recognition and de-noising. In the area of pattern recognition, we propose and implement two invariant descriptors for the recognition of 2-D patterns. The first invariant descriptor is concerned with patterns which can be represented by periodic 1-D signals. The method first performs orthonormal shell decomposition on the periodic 1-D signals, then applies Fourier transform on each scale of the shell coefficients. The essential advantage of the descriptor is that a multi-resolution querying strategy can be employed in the recognition process and that it is invariant to rotation of the original 2-D pattern. The second invariant descriptor can be used for any pattern. We first transform the pattern to polar coordinate $(r, \theta)$ using the centre of mass of the pattern as origin, then apply the Fourier transform along the axis of polar angle $\theta$ and the wavelet transform along the axis of radius $r$. The features thus obtained are invariant to translation and rotation. Experimental results show that the two invariant descriptors are efficient representation which can provide for reliable recognition.

In de-noising, we develop a new translation-invariant(TI) multiwavelet de-noising algorithm. Instead of using univariate thresholding developed by Donoho, we adopt bivariate thresholding as pioneered by Downie and Silverman. Numerical simulation shows that TI multiwavelet de-noising is better than TI single wavelet de-noising when soft thresholding is used.

# Acknowledgements

I would like to thank my supervisor, Professor T. D. Bui, for introducing me to the fascinating field of wavelet analysis and for his financial support, advice, and encouragement during this work. He carefully guided me through three and a half years at Concordia, discussed all ideas in this thesis, read all its words and taught me a lot. Thank you very much !

I am also grateful to Dr. A. Krzyzak and Dr. C. Y. Suen whose excellent classes gave me a deeper understanding of pattern recognition. Thanks also go to the analysts in the Computer Science Department for their help during my study at Concordia.

I would further like to express my gratitude to my sponsor, the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche of Québec, whose generous FCAR scholarship makes my life easier.

Finally, special thanks go to my wife, Liping Wang, for her understanding, sacrifice and love.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Preliminaries

In this chapter, we will review some basic aspects of the Fourier and wavelet transforms.

## 1.1 The Fourier Transform

The Fourier transform's utility lies in its ability to analyse a signal in the time domain for its frequency content. The transform works by first translating a function in the time domain into a function in the frequency domain. The signal can then be analysed for its frequency content because the Fourier coefficients of the transformed function represent the contribution of each sine and cosine function at each frequency. An inverse Fourier transform does just what you expect, transform data from the frequency domain into the time domain.

### 1.1.1 The Continuous Fourier Transform

Let $f(x)$ be a continuous function of a real variable. The Fourier transform of $f(x)$ is defined by the equation

$$\hat{f}(u) = \int_{-\infty}^{+\infty} f(x)e^{-iux}dx \tag{1}$$

Given $\hat{f}(u)$, $f(x)$ can be obtained by the inverse Fourier Transform

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(u)e^{iux}du \tag{2}$$

The Fourier transform can be easily extended to a function $f(x,y)$ of two variables:

$$\hat{f}(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-i(ux+vy)}dxdy \tag{3}$$

and

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \hat{f}(u,v)e^{i(ux+vy)}dudv \tag{4}$$

where $u$ and $v$ are the frequency variables.

### 1.1.2 The Discrete Fourier Transforms

The discrete Fourier transform(DFT) estimates the Fourier transform of a function from a finite number of its sampled points. The sample points are supposed to be typical of what the signal looks like at all other times. Suppose that a continuous function $f(x)$ is discretized into a sequence of $\{f(x_0), f(x_0+\Delta x), \ldots, f(x_0+(n-1)\Delta x)\}$ by taking $n$ samples $\Delta x$ apart. We may define

$$f(x) = f(x_0 + x\Delta x)$$

where $x$ now assumes the discrete values 0, 1, ..., $n-1$. With this notation in mind, the discrete Fourier transform pair that applies to sampled functions is given by

$$\hat{f}(u) = \frac{1}{n}\sum_{x=0}^{n-1} f(x)e^{-iux/n} \tag{5}$$

for $u = 0,1,2,\ldots,n-1$, and

$$f(x) = \sum_{u=0}^{n-1}\hat{f}(u)e^{iux/n} \tag{6}$$

for $x = 0,1,2,\ldots,n-1$.

In the two-variable case the discrete Fourier transform pair is given by the equations

$$\hat{f}(u,v) = \frac{1}{n^2}\sum_{x=0}^{n-1}\sum_{y=0}^{n-1} f(x,y)e^{-i(ux+vy)/n} \tag{7}$$

for $u = 0,1,2,\ldots,n-1$, $v = 0,1,2,\ldots,n-1$, and

$$f(x,y) = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1}\hat{f}(u,v)e^{i(ux+vy)/n} \tag{8}$$

for $x = 0,1,2,\ldots,n-1$, $y = 0,1,2,\ldots,n-1$.

### 1.1.3 The Fast Fourier Transform

To approximate a function by samples, and to approximate the Fourier integral by the discrete Fourier transform, requires applying a matrix whose order is the number of sample points $n$. Since multiplying a matrix by a vector costs on the order of $O(n^2)$ arithmetic operations, the problem gets quickly worse as the number of sample points increases. However, if the samples are uniformly spaced, then the Fourier matrix can be factored into a product of just a few sparse matrices, and the resulting factors can be applied to a vector in a total of order $O(n \log_2 n)$ arithmetic operations. This is the so-called Fast Fourier Transform(FFT).

### 1.1.4 Some Properties of the Fourier Transform

Attention is focused in this section on properties of the Fourier transform which will be of value in subsequent discussions.

**Translation:** The translation properties of the Fourier transform pair with one variable or two variables are given by

$$f(x)e^{iu_0x/n} \Longleftrightarrow \hat{f}(u-u_0) \tag{9}$$

$$f(x-x_0) \Longleftrightarrow \hat{f}(u)e^{-iux_0/n} \tag{10}$$

and

$$f(x,y)e^{i(u_0x+v_0y)/n} \Longleftrightarrow \hat{f}(u-u_0,v-v_0) \tag{11}$$

$$f(x-x_0,y-y_0) \Longleftrightarrow \hat{f}(u,v)e^{-i(ux_0+vy_0)/n} \tag{12}$$

It is interesting to note that a shift in $f(x)$ or $f(x,y)$ does not affect the magnitude of its Fourier transform since

$$|\hat{f}(u)e^{-iux_0/n}| = |\hat{f}(u)| \tag{13}$$

$$|\hat{f}(u,v)e^{-i(ux_0+vy_0)/n}| = |\hat{f}(u,v)| \tag{14}$$

This property is especially useful in deriving invariant features in pattern recognition.

**Rotation:** If $f(x,y)$ is rotated by an angle $\theta_0$, then $\hat{f}(u,v)$ is rotated by the same angle. Similarly, rotating $\hat{f}(u,v)$ causes $f(x,y)$ to be rotated by the same angle.

## 1.2 The Wavelet Transform

In this section we briefly review the most important properties of the wavelet transform and the wavelet based multiresolution decompositions. For more detailed information about wavelet transform, readers can be directed to [2], [7], [10], [11], [34]-[38]. Wavelets are functions that satisfy certain mathematical requirements. The very name *wavelet* comes from the requirement that they should integrate to zero, "waving" above and below the $x$-axis. The diminutive connotation of wavelet suggest the function has to be well localised. Other requirements are technical and needed mostly to insure quick and easy calculation of the direct and inverse wavelet transform.

The orthonormal basis of compactly supported wavelets of $L^2(R)$ is formed by the dilation and translation of a single function $\psi(x)$

$$\psi_{j,k}(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - k),$$

where $j,\ k \in Z$. The function $\psi(x)$ has a companion, the scaling function $\phi(x)$, and these functions satisfy the following relations:

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k\phi(2x - k), \tag{15}$$

$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k\phi(2x - k), \tag{16}$$

where $h_k$ and $g_k$ are called low-pass and high-pass filter coefficients respectively, and

$$g_k = (-1)^k h_{L-k-1}, \quad k = 0, \ldots, L-1$$

$$\int_{-\infty}^{+\infty} \phi(x)dx = 1.$$

The filter coefficients are assumed to satisfy the orthogonality relations:

$$\sum_n h_n h_{n+2j} = \delta(j), \tag{17}$$

and

$$\sum_n h_n g_{n+2j} = 0. \tag{18}$$

for all $j$, where $\delta(0) = 1$ and $\delta(j) = 0$ for $j \neq 0$.

The vanishing moments property simply means that the basis functions are chosen to be orthogonal to the low degree polynomials, namely, if the set of functions $\{\psi(x-k)\}_{k\in Z}$ is an orthonormal basis of $W_0$, then

$$\int_{-\infty}^{+\infty} \psi(x)x^m dx = 0, \quad m = 0,\ldots,M-1. \tag{19}$$

The number of coefficients $L$ in (15) and (16) may be related to the number of vanishing moments $M$. However, no matter what conditions are imposed, $L$ is always even.

### 1.2.1 Multi-resolution Analysis

The wavelet basis induces a multi-resolution analysis on $L^2(R)$. i.e., the decomposition of the Hilbert space $L^2(R)$, into a chain of closed subspaces

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots \tag{20}$$

such that

1. $\cap_{j\in Z} V_j = \{0\}$ and $\cup_{j\in Z} V_j$ is dense in $L^2(R)$
2. For any $f \in L^2(R)$ and any $j \in Z$, $f(x) \in V_j$ if and only if $f(2x) \in V_{j-1}$
3. For any $f \in L^2(R)$ and any $k \in Z$, $f(x) \in V_0$ if and only if $f(x-k) \in V_0$
4. There exists a function $\psi \in V_0$ such that $\{\psi(x-k)\}_{k\in Z}$ is an orthogonal basis of $V_0$.

Let us define the subspaces $W_j$ as an orthogonal complement of $V_j$ in $V_{j-1}$,

$$V_{j-1} = V_j \oplus W_j \tag{21}$$

and represent the space $L^2(R)$ as a direct sum

$$L^2(R) = \bigoplus_{j\in Z} W_j \tag{22}$$

Selecting the coarsest scale $J$, we may replace the chain of the subspaces (20) by

$$V_J \subset \cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots \tag{23}$$

and obtain

$$L^2(R) = V_J \bigoplus_{j \leq J} W_j \tag{24}$$

If there is a finite number of scales then, without loss of generality, we set $j = 0$ to be the finest scale and consider

$$V_J \subset \cdots \subset V_2 \subset V_1 \subset V_0, \quad V_0 \subset L^2(R) \tag{25}$$

On each fixed scale $j$, the wavelets $\{\psi_{j,k}(x)\}_{j \in Z}$ form an orthonormal basis of $W_j$ and the functions $\{\phi_{j,k}(x) = 2^{-\frac{j}{2}}\phi(2^{-j}x - k)\}_{j \in Z}$ form an orthonormal basis of $V_j$. The coefficients $H = \{h_k\}_{k=0}^{L-1}$ and $G = \{g_k\}_{k=0}^{L-1}$ are quadrature mirror filters. Once the filter $H$ has been chosen, it completely determines the functions $\psi$ and $\phi$. Let us define the $2\pi$−periodic function

$$m_0(\mu) = \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} h_k e^{ik\mu}. \tag{26}$$

then the function $m_0(\mu)$ satisfies the equation

$$|m_0(\mu)|^2 + |m_0(\mu + \pi)|^2 = 1. \tag{27}$$

for the coefficients $h_k$.

### 1.2.2 The Fast Wavelet Transform

Daubechies has discovered that the wavelet transform can be implemented with a specially designed pair of Finite Impulse Response(FIR) filters called a "Quadrature Mirror Filter"(QMF) pair. The output of the QMF filter pair are down-sampled by a factor of two, that is, every other output sample of the filter is kept, the others are discarded. The low-frequency filter output is fed into another identical QMF filter pair. This operation can be repeated as a pyramid algorithm, yielding a group of signals that divides the spectrum of the original into octave bands with successively coarser measurements in time as the width of each spectral band narrows and decreases in frequency.

The fast wavelet transform is actually more computationally efficient than the Fast Fourier Transform. As mentioned previously, a FFT of length $n$ (where $n$ is

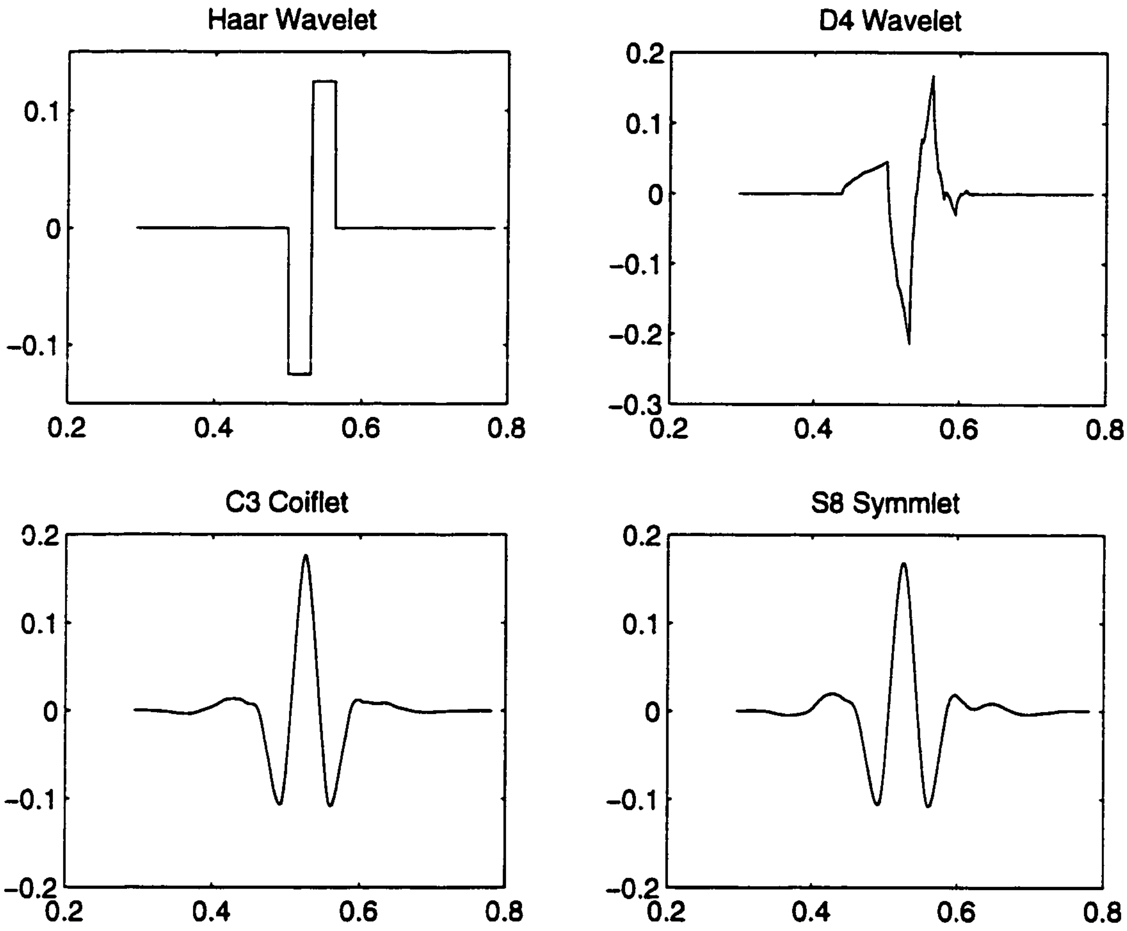

Figure 1: Some single wavelet families used in this thesis

an integral power of 2) takes on the order of $O(n \log_2 n)$ operations. A fast wavelet transform of length $n$ requires approximately $O(n)$ operations - the best possible.

### 1.2.3 Some Wavelet Families

There are many kinds of wavelets. One can choose between smooth wavelets, compactly supported wavelets, wavelets with simple mathematical expressions, wavelets with simple associated filters, etc. In order to test the performance of different wavelet families in our pattern recognition experiments, we use the following wavelet filters. Figure 21 shows these wavelets reproduced from WAVELAB developed by D. L. Donoho.

The **Haar** filter is discontinuous, and can be considered a Daubechies-2. Its scaling filter is

$$H = (1/\sqrt{2}, 1/\sqrt{2}).$$

The **Daubechies-4** filter has its advantage on its most compact support of 4 and its orthogonality. The size 4 is indeed shortest even span in which the second derivatives are computable. Its scaling filter is

$$H = (0.482962913145,\quad 0.836516303738,\quad 0.224143868042,\quad -0.129409522551).$$

The **Coiflet** filters are designed to give both the mother and father wavelets 2, 4, 6, 8, or 10 vanishing moments (see Daubechies for the definition of vanishing moments and their usefulness). Here we only test the 6 vanishing moment case. Its scaling filter is

$$\begin{aligned} H = (&0.0385807777748,\quad -0.126969125396,\quad -0.077161555496,\quad 0.607491641386,\\ &0.745687558934,\quad 0.226584265197). \end{aligned}$$

The **Symmlet-8** is the least asymmetric compactly-supported wavelets with 8 vanishing moments. Its scaling filter is

$$\begin{aligned} H = (&-0.107148901418,\quad -0.041910965125,\quad 0.703739068656,\quad 1.136658243408,\\ &0.421234534204,\quad -0.140317624179,\quad -0.017824701442,\quad 0.045570345896). \end{aligned}$$

## 1.3 Fourier versus Wavelet Transform

### 1.3.1 Similarities between Fourier and Wavelet Transform

The fast Fourier transform(FFT) and the discrete wavelet transform(DWT) are both linear operations that generate a data structure that contains $\log_2 n$ segments of various lengths, usually filling and transforming it into a different data vector of length $2^n$.

The mathematical properties of the matrices involved in the transforms are similar as well. The inverse transform matrix for both the FFT and the DWT is the transpose of the original. As a result, both transforms can be viewed as a rotation in function space to a different domain. For the FFT, this new domain contains basis functions that are sines and cosines. For the wavelet transform, this new domain contains more complicated basis functions called wavelets, mother wavelets, or analysing wavelets.

Both transforms have another similarity. The basis functions are localised in frequency, making mathematical tools such as power spectra (how much power is contained in a frequency interval) and scalegrams useful at picking out frequencies and calculating power distribution.

### 1.3.2 Dissimilarities between Fourier and Wavelet Transforms

The most interesting dissimilarity between these two kinds of transforms is that individual wavelet functions are localised in space. Fourier sine and cosine functions are not. This localisation feature, along with wavelets localisation of frequency, makes many functions and operators using wavelets sparse when transformed into the wavelet domain. This sparseness, in turn, results in a number of useful applications such as data compression, detecting features in images, and removing noise from time series.

One way to see the differences in time-frequency resolution between the Fourier transform and the wavelet transform is to look at the basis function coverage of the time-frequency plane. Figure 2 shows a windowed Fourier transform, where the window is simply a square wave. The square wave window truncates the sine or cosine function to fit a window of a particular width. Because a single window is used for all frequencies in the Fourier transform, the resolution of the analysis is the same at all locations in the time-frequency plane.

An advantage of wavelet transforms is that the windows vary. In order to isolate signal discontinuities, one would like to have some very short basis functions. At the same time, in order to obtain detailed frequency analysis, one would like to have some very long basis functions. A way to achieve this is to have short high-frequency basis functions and long low-frequency ones. This happy medium is exactly what you get with wavelet transforms. Figure 3 shows the coverage in the time-frequency plane with one wavelet function, the Haar wavelet.

The Wavelet representation also provides a coarse-to-fine matching strategy in pattern recognition, called *multi-resolution matching*. The matching starts from the coarsest scale and moves on to the finer scales. The costs for different scales are quite

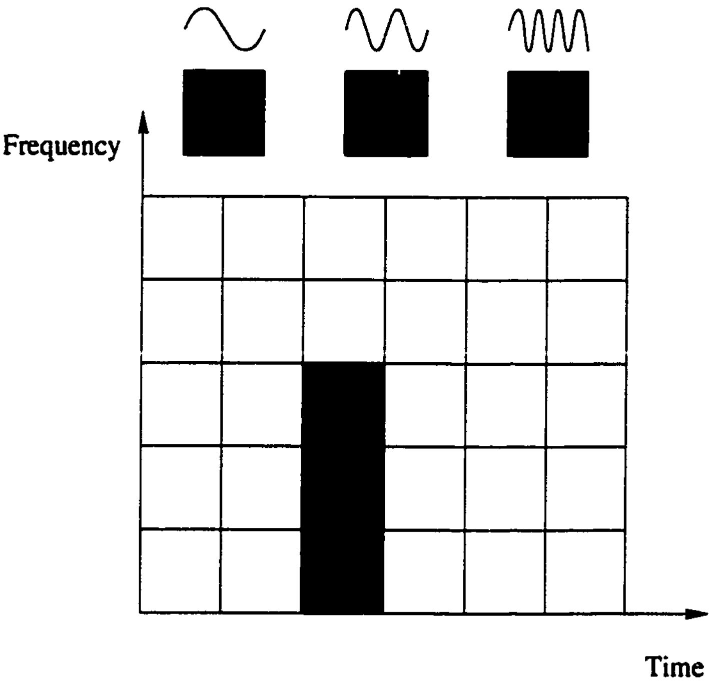

Figure 2: Fourier basis function, time-frequency tiles, and coverage of time-frequency plane

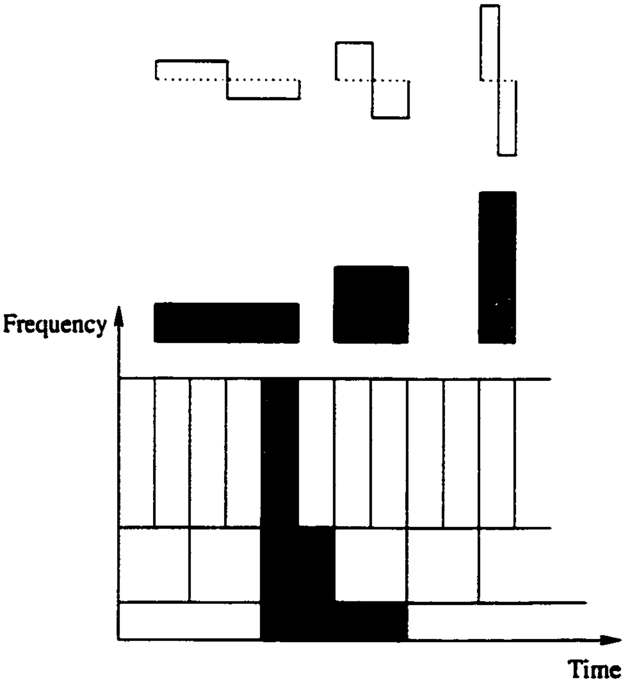

Figure 3: Haar wavelet basis functions, time-frequency tiles, and coverage of the time-frequency plane

different. Since the coarsest scale has only a small number of coefficients, the cost at this scale is much less than for finer scales. In practice, the majority of patterns can be unambiguously identified during the coarse scale matching, while only few patterns will need information at finer scales to be identified. Therefore, the process of multi-resolution matching will be faster compared to the conventional matching techniques.

One more thing to remember is that wavelet transforms do not have a single set of basis functions like the Fourier transform, which utilises just the sine and cosine functions. Instead, wavelet transforms have an infinite set of possible basis functions. Thus wavelet analysis provides immediate access to information that can be obscured by other time-frequency methods such as Fourier analysis.

## 1.4 Outline of the Thesis

In this thesis, we consider new applications of wavelet transform in pattern recognition and signal de-noising. We propose and implement two invariant descriptors for the recognition of 2-D patterns and one translation invariant de-noising scheme by using multiwavelets. The organisation of the thesis is as follows.

Chapter 2 proposes an invariant descriptor for recognising 2-D patterns which can be represented by periodic 1-D signals such as the contour of an object, the ring-projection, the line-moment, etc. The descriptor performs orthonormal shell decomposition on the periodic 1-D signal, then applies Fourier transform on each scale of the shell coefficients. The essential advantage of the descriptor is that a multi-resolution querying strategy can be employed in the recognition process and that it is invariant to rotation of the original 2-D pattern. The translation- and scale-invariant properties can be obtained while extracting the 1-D periodic signal from the original 2-D pattern. Experimental results show that the descriptor proposed is a reliable tool for pattern recognition and it is robust to white noises.

Chapter 3 presents a novel descriptor for recognising complex patterns such as aircrafts, keys, road signs, printed Chinese characters, etc. We first transform the pattern to polar coordinate $(r, \theta)$ using the centre of mass of the pattern as origin, then apply the Fourier transform along the axis of polar angle $\theta$ and the wavelet transform along the axis of radius $r$. The features thus obtained are invariant to translation and rotation. As an example, we apply the method to a database of 85 printed Chinese characters. The result shows that the descriptor using a combination of Fourier and wavelet transforms is an efficient representation which can provide for reliable recognition.

Chapter 4 gives a translation-invariant(TI) multiwavelet de-noising scheme using bivariate thresholding rule. TI single wavelet de-noising was developed by Coifman and Donoho [13] and it is better than the non-TI single wavelet de-noising. Strela et al. [47] has found that non-TI multiwavelet de-noising gives better results than the non-TI single wavelets. In this thesis we extend Coifman's TI single wavelet de-noising scheme to multiwavelets. The experiment results show that TI multiwavelet de-noising is better than the single wavelet case for soft thresholding.

Finally, Chapter 5 gives the conclusions of current work and propose the future work in the area of pattern recognition and de-noising.

# Chapter 2

# Multi-resolution Orthonormal Shell-Fourier Descriptor for Pattern Recognition

## 2.1 Introduction

In pattern recognition the dimension of the feature vector is very important in recognition process. A very big feature vector may require a large amount of computation time. Therefore, it is desirable to have a smaller feature vector while keeping the original pattern's important features as much as possible. Among all the existing techniques, one major approach is to transform the original 2-D pattern into a 1-D periodic signal, say, the contour of a pattern, the line moment of a pattern, the ring-projection, and so forth.

Fourier transform has been a powerful tool for pattern recognition ([1], [21], [31], [50], [54]). However, Fourier basis is not local in the spatial domain. A local variation of the input signal can affect all the Fourier coefficients. In addition, Fourier descriptor does not have a multi-resolution representation. Whatever the size of the target and the noise level, matching has to be performed at a designed scale. Wavelet descriptor([5], [30], [32]-[42], [52], [38]), on the other hand, provides better global shape features at the low scales as well as detail features at high differential scales. Besides, wavelet descriptor offers a natural multi-resolution representation of the signal so that we can employ a multi-resolution matching strategy. It has been claimed that wavelet descriptor can provide higher recognition rate for hand-printed character recognition([29], [52]) and aircraft recognition [30].

One intuitive way is to apply wavelet transform directly on the 1-D signal in our application. However, wavelet transform is not translation-invariant. A small shift in the input signal may cause totally different wavelet coefficients. Therefore we have to find a way to overcome this shift-variance problem before wavelet descriptor can be used here.

Basically speaking, there are two ways to overcome the shift problem. One is to normalise the input signal so that the starting point is fixed at a specific position. Another way is to use shift invariant wavelet decomposition. Several methods based on the best-basis-selection approach have been proposed in order to get shift-invariant wavelet decomposition([32], [33], [39]). In those methods, a cost function is used as a measure of shift-invariant property. The input signal is decomposed into wavelet coefficients for all possible shifts by means of a tree structure. The set of wavelet coefficients that minimises the given cost function is then chosen as the shift invariant wavelet representation. Another kind of shift-invariant wavelet transform is proposed by Y. Hui et al. ([25], [26]). They establish a framework for shift-invariant filter bank by connecting the relation between the polyphase representation and shift-invariant property of the filter banks. The proposed framework is independent of input signal and yields filters with shift-invariant property without changing the structure of dydiac wavelet transform.

Even though the above mentioned methods can deal with the shift problem in wavelet transform, they could not be successfully applied in pattern recognition applications. The reason is that they are very sensitive to noise in the original signal. Little noise in the original signal may cause totally different decision. Therefore, we do not adopt the above mentioned methods. Instead, we use a combination of the concept of orthonormal shell decomposition introduced by Saito and Beylkin [42] and the Fourier transform to eliminate the shift problem.

The basic idea proposed in this chapter is as follows. Following Saito and Beylkin [42], we decompose the 1-D signal into multi-resolution scales using orthonormal shell expansions (wavelet transform without down sampling). In order to eliminate the shifts in each scale, we perform Fourier transform on it and obtain the spectrum. The feature obtained in this way is invariant to the original pattern's orientation. The translation- and scale-invariance can be guaranteed during the process of extracting the periodic 1-D signal. Experimental results show that the new descriptor is better than the Fourier descriptor for recognising printed Chinese characters.

It should be mentioned that both Fourier transform and the orthonormal shell decomposition (wavelet transform) used in this chapter are discrete transforms.

## 2.2 The Orthonormal Shell Expansion

This section reviews briefly the concept of orthonormal shell developed in [42]. It has been mentioned in the introduction that the coefficients of orthogonal wavelet expansions are not shift-invariant. However, if all the wavelet coefficients of $n$ circulant shifts of a vector (signal) are computed, we may use them when shift invariance is important. Based on this observation, the notion of a shell (much more redundant than a frame) was introduced in [42] to obtain a redundant but shift-invariant family of functions.

In our applications, there is always the finest and the coarsest scales of interest and therefore the number of scales is finite and we can consider only shifts by multiples of some fixed unit. Assuming that the finest scale is described by the $n$-dimensional subspace $V_0$ and consider only circulant shifts in $V_0$. Let $V_{j_0}$ be the subspace describing the coarsest scale $(1 \leq j_0 \leq J)$ where $n = 2^J$, and let $\psi_{j,k}(x) = 2^{-\frac{j}{2}}\psi(2^j(x-k))$ and $\phi_{j,k}(x) = 2^{-\frac{j}{2}}\phi(2^{-j}(x-k))$. Therefore, the functions $\{\psi_{j,k}\}_{1\leq j\leq j_0, 0\leq k\leq 2^{J-j}-1}$ and $\{\phi_{j_0,k}\}_{0\leq k\leq 2^{J-j_0}-1}$ generate the coefficients $s_k^j$ and $d_k^j$:

$$s_k^j = \int f(x)\phi_{j,k}(x)dx, \tag{28}$$

and

$$d_k^j = \int f(x)\psi_{j,k}(x)dx. \tag{29}$$

for $j = 1, 2, \ldots, j_0$ and $k = 0, 1, \ldots, 2^{J-j} - 1$. The coefficients $\{d_k^j\}_{1\leq j\leq j_0, 0\leq k\leq 2^{J-j}-1}$ are known as orthonormal wavelet coefficients.

In the case of orthonormal shell decomposition, Saito and Beylkin defined the functions $\{\psi_{j,k}(x)\}_{1\leq j\leq j_0, 0\leq k\leq 2^J-1}$ and $\{\phi_{j_0,k}(x)\}_{0\leq k\leq 2^J-1}$ as a shell of the orthonormal wavelets for shifts in $V_0$. As a consequence, the coefficients $\{d_k^j\}_{1\leq j\leq j_0, 0\leq k\leq 2^J-1}$ and $\{s_k^{j_0}\}_{0\leq k\leq 2^J-1}$ are called the orthonormal shell coefficients. Clearly the set of coefficients in the orthonormal shell is much more abundant and overly redundant compared to the set of coefficients in the orthogonal wavelet transform. However, this redundancy is needed for our shift invariant property.

Assuming that the orthonormal wavelet coefficients of the finest scale $\{s_k^0\}_{0\leq k\leq n-1}$ are given as an original signal and let us consider the function $f = \sum_{l=0}^{n-1} s_k^0 \phi_{0,k}$. The orthonormal shell coefficients of this function $f$ are obtained from the quadrature mirror filters $H = \{h_l\}_{0\leq l\leq L-1}$ and $G = \{g_l\}_{0\leq l\leq L-1}$ (associated with the orthonormal basis of compactly supported wavelets)

$$s_k^j = \sum_{l=0}^{L-1} h_l s_{k+2^{j-1}l}^{j-1}, \tag{30}$$

and

$$d_k^j = \sum_{l=0}^{L-1} g_l s_{k+2^{j-1}l}^{j-1}. \tag{31}$$

for $j = 1,\dots,j_0$, $k = 0,\dots,2^{J-j}-1$. The complexity of (30) and (31) is $O(n\log n)$.

It is easy to show that the recurrence relations (30) and (31) compute the orthonormal wavelet coefficients of all circulant shifts of the function $f$. For $d_k^j$, the first scale is:

$$d_k^1 = \sum_{l=0}^{L-1} g_l s_{k+l}^0 = \sum_{l=0}^{L-1} g_l s_{2k+l}^0 + \sum_{l=0}^{L-1} g_l s_{2k+1+l}^0 = d_{2k}^1 + d_{2k+1}^1. \tag{32}$$

for $k = 0,\dots,\frac{n}{2}-1$.

It is clear that the sequence $\{d_{2k}^1\}$ contains all the orthonormal wavelet coefficients that appear if $f(x)$ is circularly shifted by even numbers and the sequence $\{d_{2k+1}^1\}$ contains all the orthonormal wavelet coefficients for odd shifts.

Similarly, at the $j$-th scale

$$d_{2^jk+m}^j = \sum_{l=0}^{L-1} g_l s_{2^{j-1}(2k+l)+m}^{j-1}, \tag{33}$$

and

$$s_{2^jk+m}^j = \sum_{l=0}^{L-1} h_l s_{2^{j-1}(2k+l)+m}^{j-1}. \tag{34}$$

for $k = 0,1,\dots,2^{J-j}$, $m = 0,1,\dots,2^j-1$ .

The sequences $\{d_{2^jk}^j\}$, $\{d_{2^jk+1}^j\}$, ..., $\{d_{2^jk+2^j-1}^j\}$ contain the orthonormal wavelet coefficients of the $j$-th scale of the signal shifted by $0,1,\dots,2^j-1$, respectively. Therefore, the set $\{d_k^j\}_{1\leq j\leq j_0, 0\leq k\leq 2^{J-j_0}-1}$ and $\{s_k^{j_0}\}_{0\leq k\leq 2^{J-j_0}-1}$ contain all the coefficients of the orthonormal wavelet expansion of $f(x)$, $f(x+1)$, ..., $f(x+n-1)$. This set of

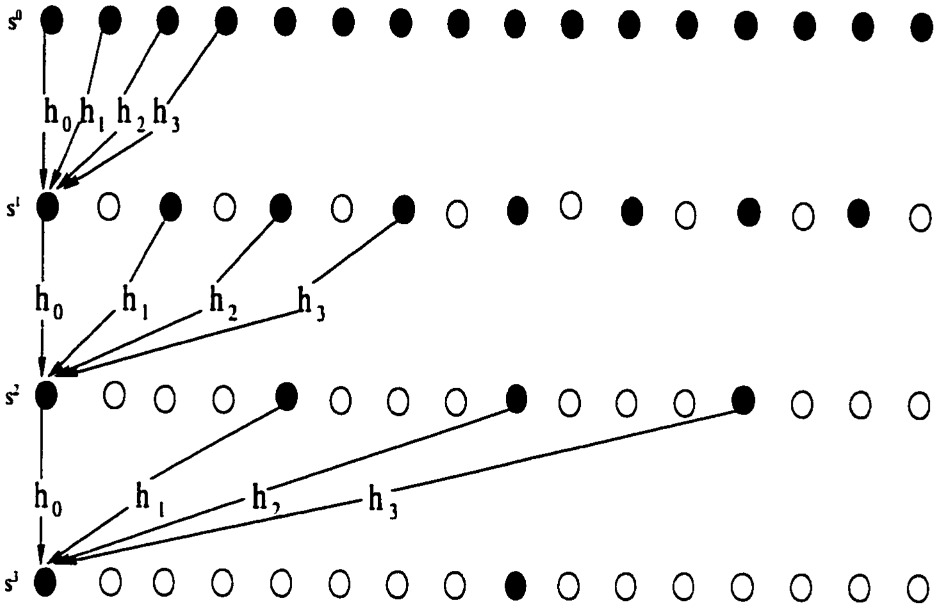

Figure 4: A scheme illustrating the algorithm for expanding a signal into multiresolution scales using the quadrature mirror filter $H = (h_0, h_1, h_2, h_3)$.

coefficients defines the orthonormal shell decomposition. The diagram for computing these coefficients is illustrated in Figure 4.

## 2.3 Extracting Periodic 1-D Signal

The problem of discriminating planar object is one of the most familiar and fundamental problems in pattern recognition. Among all the existing techniques one major approach is to transform the original planar object into 1-D signal and then extracting features from the signal. The major advantage of this approach is that the size of the feature vector is reduced, and so is the computation workload. There are many ways to get 1-D signals from the original planar object. One of the most compelling methods is to extract the outer contour of the object by means of chain code [18], angular bend functions [54], or parametric curves [21].

The chain code first described by Freeman [18] approximates a continuous contour by a sequence of piecewise linear fits that consist of eight standardised line segments. The code of a contour is then the chain V of length $k$

$$\mathrm{V} = a_1 a_2 a_3 \ldots a_k,$$

where each link $a_i$ is an integer between 0 and 7 oriented in the direction $\frac{\pi}{4} a_i$ (as measured counter-clockwise from the $x$-axis of the $x$-$y$ coordinate system) and of length 1 or $\sqrt{2}$ depending, respectively, on whether $a_i$ is even or odd.

The angular bend functions in [54] are defined as follows. We assume $\gamma$ is a clockwise-oriented simple closed curve with the parametric representation $(x(l), y(l)) = z(l)$ where $l$ is arc length and $0 \leq l \leq L$. Denote the angular direction of $\gamma$ at point $l$ by the function $\theta(l)$. Define then the cumulative angular function $\phi(l)$ as the net amount of angular bend between starting point $l = 0$ and point $l$. So $\phi(l) = \theta(l) - \theta(0)$ except for possible multiples of $2\pi$ and $\phi(L) = -2\pi$. Finally define the $\phi^*(t)$ as $\phi^*(t) = \phi(\frac{Lt}{2\pi}) + t$ where $t$ ranges from 0 to $2\pi$. It is clear that $\phi^*(t)$ is invariant under translation and scaling.

The parametric curve can be represented by parameterised $x$- and $y$-coordinates [21]. Let us denote the contour of the object by a clockwise-oriented simple closed curve with the parametric representation $(x(l), y(l)) = z(l)$, where $l$ is the arc length along the contour. A point moving along the boundary generates the complex function $u(l) = x(l) + iy(l)$ which is periodic with period $L$. This representation is translation-invariant only. However, the scale-invariance can be achieved by normalising the arc length to a fixed length.

The chain code, angular bend function, and parametric curve are only suitable for patterns with a simple contour. For complex patterns, it is difficult to trace their boundary or recognise them in terms of a single closed curve. Therefore, other methods such as ring-projection or line moment is introduced.

The ring-projection is used by Y. Y. Tang et al. [48] for optical character recognition. Suppose a 2-D pattern is represented by a binary image. One can first transform the pattern into polar coordinates by using the centroid of mass as the origin. For any fixed $r \in [0, M]$, where $M$ is the longest distance of the pattern point to the centroid, we compute the following integral

$$p(r) = \int_0^{2\pi} f(r\cos\theta, r\sin\theta) d\theta.$$

The signal $p(r)$ is equal to the total mass along the circular rings. It is clear that $p(r)$ is translation- and rotation-invariant.

The line moment is defined by Wang et al. [50]. For object $O$, we form $N$ angularly equispaced radial vectors $r_k$ departing from the centroid $C$ where angular step is $\frac{2\pi}{n}$ and $k$ is the index of $n$ vectors. The end point of $r_k$ is object's boundary point in the direction of $r_k$. Each $r_k$ includes black line segment $B_{kl}$ and white line segments

$W_{kl}$. Black line segment $B_{kl}$ represents a line segment of $r_k$ where only object exists and $1 \leq l \leq N_k$ in which $N_k$ is the number of black line segments of $r_k$. For black line segment $B_{kl}$ one can obtain its line length $L_{kl}$ and its line centroid $C_{kl}$. Thus we have the moment of $r_k$ defined by $m_k = \sum_{l=1}^{N_k} |C_{kl} - C| L_{kl}$. The line moment $m_k$ is invariant to translation only. However, we can normalise it by $\hat{m}_k = \frac{m_k}{\sum m_k}$ so that $\hat{m}_k$ is also scale-invariant.

## 2.4 Orthonormal Shell-Fourier Descriptor

Feature extraction is a crucial step in pattern recognition. In this section we describe an invariant descriptor based on a periodic 1-D signal obtained from the original unknown pattern as we described in the previous section. In general, good features must satisfy the following requirements: First, *intraclass variance* must be small, which means that features derived from different samples of the same class should be close (e.g., numerically close if numerical features are selected). Secondly, the *inter-class* separation should be large, i.e., that features derived from samples of different classes should differ significantly. Furthermore, features should be independent of the size and location of the characters. This independence can be achieved by processing or by extracting features that are translation-, rotation-, and scale-invariant.

As we explained in the section 2.2, the orthonormal shell expansion is a linear operation that decomposes a signal into components that appear at different scales. This transform is based on the convolution of the 1-D signal with a dilated filter. From the discussion of section 2.2 we know that if $\tilde{f}(x)$ is a periodically shifted version of $f(x)$, then $\tilde{d}^j(x)$ and $\tilde{s}^j(x)$ are also circularly shifted versions of $d^j(x)$ and $s^j(x)$ for all scales $j = 1, \ldots, j_0$. Since the spectrum of Fourier transform of a signal is the same no matter how much the signal is shifted, we apply Fourier transform on $d^j(x)$ at each scale $j$ ($1 \leq j \leq j_0$) and the average $s^{j_0}(x)$ and then get the spectra, which are invariant to rotation. Please note that the translation- and scale-invariance can be obtained while extracting the periodic 1-D signal from the unknown pattern. The feature vector is constructed from the low and intermediate scale wavelet coefficients and the low pass coefficients.

We can summarise the Orthonormal Shell-Fourier Descriptor as follows:

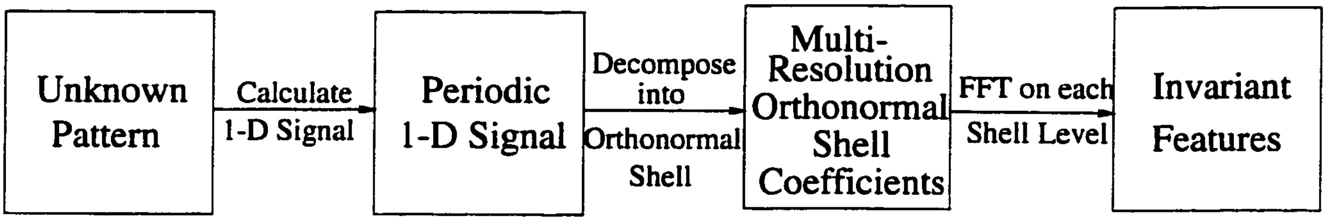

Figure 5: Block diagram of the Orthonormal Shell Fourier descriptor

1. Get the translation- and scale-invariant 1-D periodic signal $f(x)$ from the unknown pattern.

2. Apply orthonormal shell decomposition on the signal $f(x)$ up to scale $j_0$ to get $d^j(x)(1 \leq j \leq j_0)$ and $s^{j_0}(x)$.

3. Perform Fourier transform at $d^j(x)(1 \leq j \leq j_0)$ and $s^{j_0}(x)$ to get the spectra.

4. Use the resultant Fourier coefficients to query the database at different scales.

Figure 5 is the block diagram of the algorithm. Figure 6 illustrates the whole process of the algorithm. Figure 6(a) is the original character used in the experiment. Figure 6(b) is the calculated line moments $f(x)$ defined in [50]. Figure 6(c) is the multi-resolution orthonormal shell decomposition of the line moment, denoted by $d^j(x)(1 \leq j \leq 5)$ and $s^5(x)$, and Figure 6(d) shows the Fourier spectra of the decomposition coefficients at different scales. Please note that in Figure 6(c) we use $-j(1 \leq j \leq 5)$ to represent $d^j(x)$ and $-6$ the average $s^5(x)$.

The computational complexity of the algorithm can be estimated as follows. Let $n$ be the number of sample points in the periodic 1-D signal. The multi-resolution orthonormal shell decomposition of the signal will take $O(n \log n)$ operations, since we can decompose the signal up to $\log n$ scales. For each resolution shell scale, we have to perform 1-D Fourier transform, so we need $O(n \log n) \times \log n = O(n(\log n)^2)$ operations for the third step of the algorithm. In total, we need $O(n \log n + n(\log n)^2) = O(n(\log n)^2)$ operations.

## 2.5 Fast Implementation of the Algorithm

The implementation of the previous algorithm spends a lot of computation time when the original signal size is large. In fact, we can reduce the complexity of the algorithm

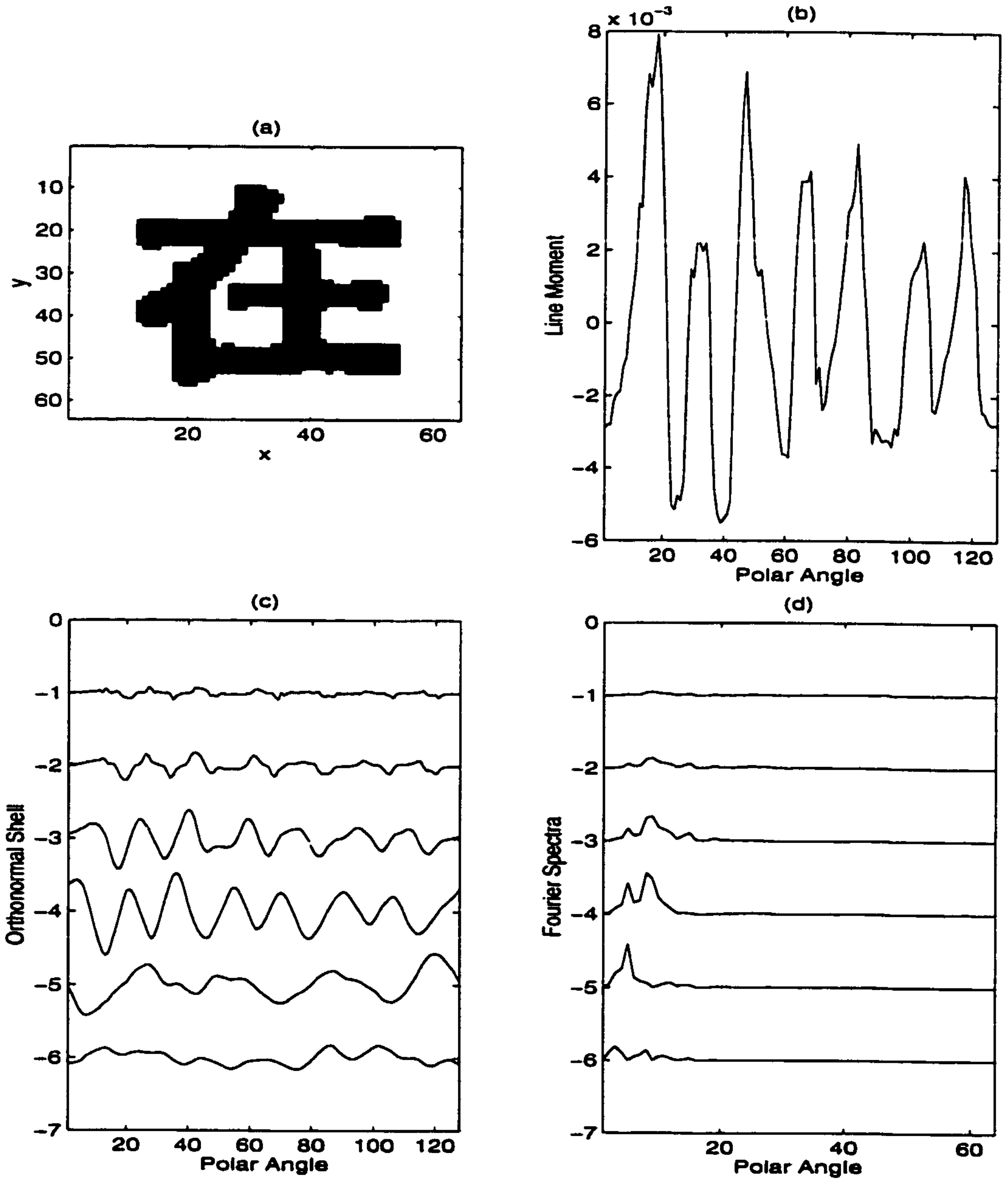

Figure 6: Representations of a pattern at each step of the new algorithm. (a) The original character in the experiment. (b) The line moment calculated from the character. (c) Multi-resolution orthonormal shell decomposition of the line moment. Please note that we use $-j(1 \leq j \leq 5)$ to represent $d^j(x)$ and $-6$ the average $s^5(x)$. (d) The spectra of Fourier transform at each scale and the average.

from $O(n(\log n)^2)$ to $O(n \log n)$.

Let us rewrite (30) and (31) in the Fourier domain

$$\hat{s}^j(\mu) = \sqrt{2}\, \overline{m_0(2^{j-1}\mu)} \hat{s}^{j-1}(\mu), \tag{35}$$

and

$$\hat{d}^j(\mu) = \sqrt{2}\, \overline{m_1(2^{j-1}\mu)} \hat{s}^{j-1}(\mu). \tag{36}$$

for $j = 1, \ldots, j_0$.

Here the $m_0(\mu)$ and $m_1(\mu)$ are defined as

$$m_0(\mu) = \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} h_k e^{ik\mu}, \tag{37}$$

and

$$m_1(\mu) = \frac{1}{\sqrt{2}} \sum_{k=0}^{L-1} g_k e^{ik\mu}. \tag{38}$$

Taking the modulus of (35) and (36), we have

$$|\hat{s}^j(\mu)| = \sqrt{2} |m_0(2^{j-1}\mu)| |\hat{s}^{j-1}(\mu)|, \tag{39}$$

and

$$|\hat{d}^j(\mu)| = \sqrt{2} |m_1(2^{j-1}\mu)| |\hat{s}^{j-1}(\mu)|. \tag{40}$$

for $j = 1, \ldots, j_0$.

We know that $s^0$ is the original 1-D signal, so by recursively applying (39) and (40) we can have a much cheaper implementation of the algorithm:

1. Get the translation- and scale-invariant 1-D periodic signal $f(x)$ from the unknown pattern.

2. Calculate $|\hat{d}^j|$ $(1 \leq j \leq j_0)$ and $|\hat{s}^{j_0}|$ by recursively applying (39) and (40).

3. Use the resultant coefficients to query the database at different scales.

Applying (39) and (40) will take $O(n)$ operations because $m_0(\mu)$ and $m_1(\mu)$ have only $L$ terms. As such, we need $O(n \log n)$ operations to recursively apply (39) and (40). In total, the fast algorithm requires $O(n \log n)$ operations.

## 2.6 Multi-resolution Matching

In theory, the matching process can be done from coarse to fine scales. For each scale, we match the features of the target with those of the patterns in the database and we have three decisions to make:

1. Accept the target as a specific pattern.
2. Reject the target.
3. Mark those entries in the database that are similar to the target as to be determined and begin the next iteration.

If the target is accepted or rejected, the matching process is then terminated. If the target is undetermined, we continue the matching process to the next finer scale, but only those entries that are marked as to be determined are used. Since there are less entries in the finer scale, the matching process can be conducted much cheaper than the traditional matching method. We can use either $L_1$ or $L_2$ distance metric in each scale. Figure 7 illustrates the multi-resolution matching process.

Even though the above mentioned multi-resolution approach is similar to humans' simultaneous interpretation of visual information, it is not trivial to apply it in real applications. There are two aspects we have to consider. First, we have to determine a threshold in order to give a guideline for acception, rejection, and to be determined. There is no way to give an optimal threshold mathematically. One has to choose it by doing a lot of experiments. Second, significant features are lost at very low resolution scales and it is likely that for very high resolution scales the intraclass variance becomes larger because of the deformation of the pattern and the accumulation errors during the transformations. Therefore, it is desirable to use only intermediate resolution scales, say 3 - 5, during the classification phase.

## 2.7 Experimental Results

In order to test the efficiency of the proposed algorithm, we use the 1-D line moments of an object defined in [50] in our experiments. The definition of moments in this experiment is nearly the same as [50] except we normalise it to have zero mean. The experiments are done under Sun Solaris 2.5. There are 85 printed Chinese characters

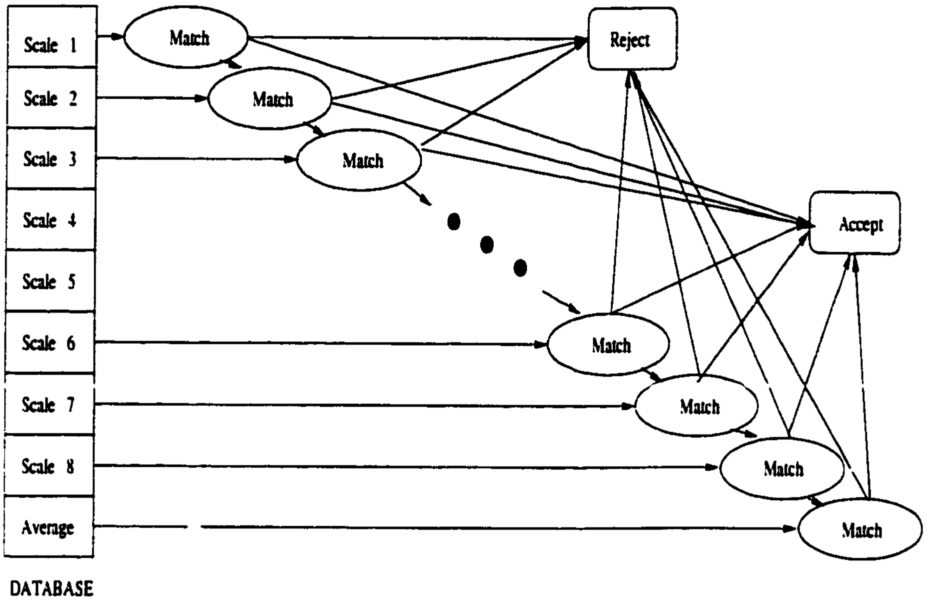

Figure 7: The multi-resolution Matching

in our experiments. Each character is represented by 64 × 64 pixels, with 1 denoting the object and 0 the background. We use 128 samples points in the computed 1-D line moment and all its multi-resolution Orthonormal Shell decomposition signals. Since the spectra of Fourier transform are symmetric, we only keep 64 coefficients at each scale. We use $L_1$ distance in the recognition process.

We tested the algorithm when rotation and scaling are present. The tested rotation angles are 30°, 60°, 90°, 120°, 180°, and 270°, while the scaling factors are 0.5, 0.6, 0.7, 0.8, 0.9, and 1.2. For illustration, we show the six rotation angles in Figure 8, the six scaling factors in Figure 9, and a combination of rotation and scaling in Figure 10 for the printed Chinese character "zai". The wavelet families used in our experiment are Haar, Daubechies-4, Coiflet-6, and Symmlet-8.

The Orthonormal Shell coefficients of a signal have multi-resolution representation of the original signal. Generally speaking, the coarse scale coefficients represent the global feature of the signal, while the fine scale coefficients represent the details. Due to noise introduced in the pattern, it is advantageous to use only a few low and intermediate scale coefficients. As the Fourier coefficients are becoming smaller as the index goes up, we keep only 20 low frequency Fourier coefficients for each scale. As such, the total coefficients stored in the database for each character are reduced significantly and we can make the querying process faster.

In our experiment, we test the performance of the Orthonormal-Fourier descriptor when only three scales are used. For each scale we only use the magnitude of 20 low frequency Fourier coefficients. When only rotation is presented to the Orthonormal-Fourier descriptor, we obtain 100% recognition rates for all different rotation angles by using different kind of wavelets. When only scaling is added to the testing pattern, we get the results in Table 1 by using features $d^4$, $d^5$ and $s^5$. In addition, we obtain the experimental results in Table 3 by using features $d^3$, $d^4$ and $d^5$. When a combination of rotation and scaling factors are added to the testing pattern, we get the experimental results in Table 2 by using features $d^4$, $d^5$ and $s^5$. Also, we obtain the experimental results in Table 4 by using features $d^3$, $d^4$ and $d^5$.

We do the same experiments by using the Fourier descriptor. Since we use the Fourier spectrum of the 1-D signal in the Orthonormal Shell-Fourier descriptor, we also use the Fourier spectrum as invariant features for the Fourier descriptor. From Table 3 and Table 4 one can see that the Orthonormal Shell-Fourier descriptor is much better than the Fourier descriptor in the sense of recognition rate, and it is very robust to geometric distortion.

We also test the robustness of the new algorithm against pseudo-random noise. In our experiments we add white noise to the 1-D periodic signal with different signal to noise ratios (SNR). The SNR is defined as $\sqrt{var(f)/\sigma^2}$, where $var(f)$ is the variance of the signal $f(t)$. Figure 11 shows the signal (solid line) and the noisy signal (dash dot line) with white noise added. Table 5 lists the recognition rates of the Orthonormal Shell-Fourier Descriptor for different SNR's. It is clear that the new descriptor is very robust to noise.

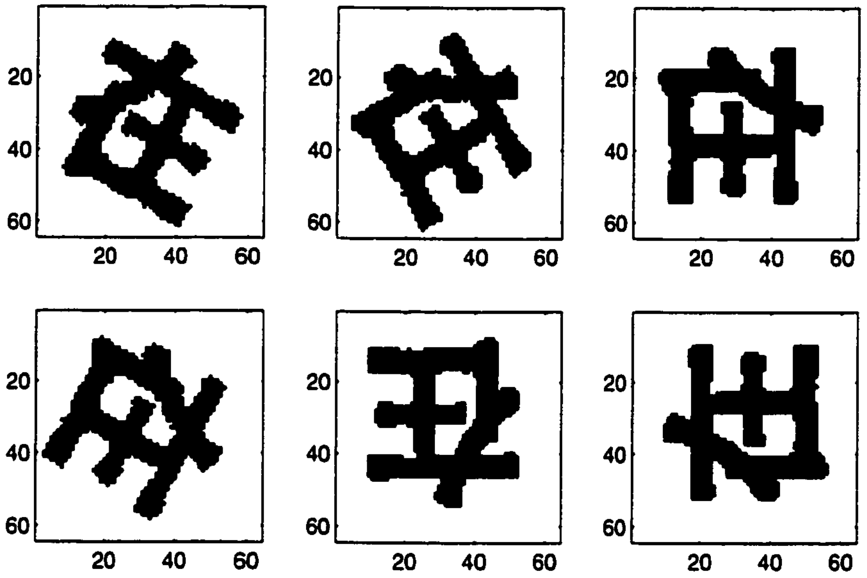

Figure 8: The six rotation angles for character "zai"

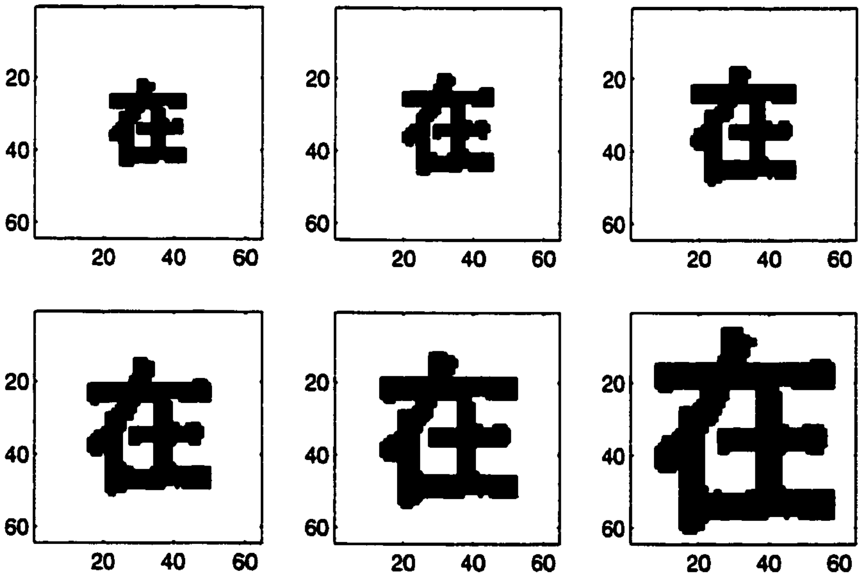

Figure 9: The six scaling factors for character "zai"

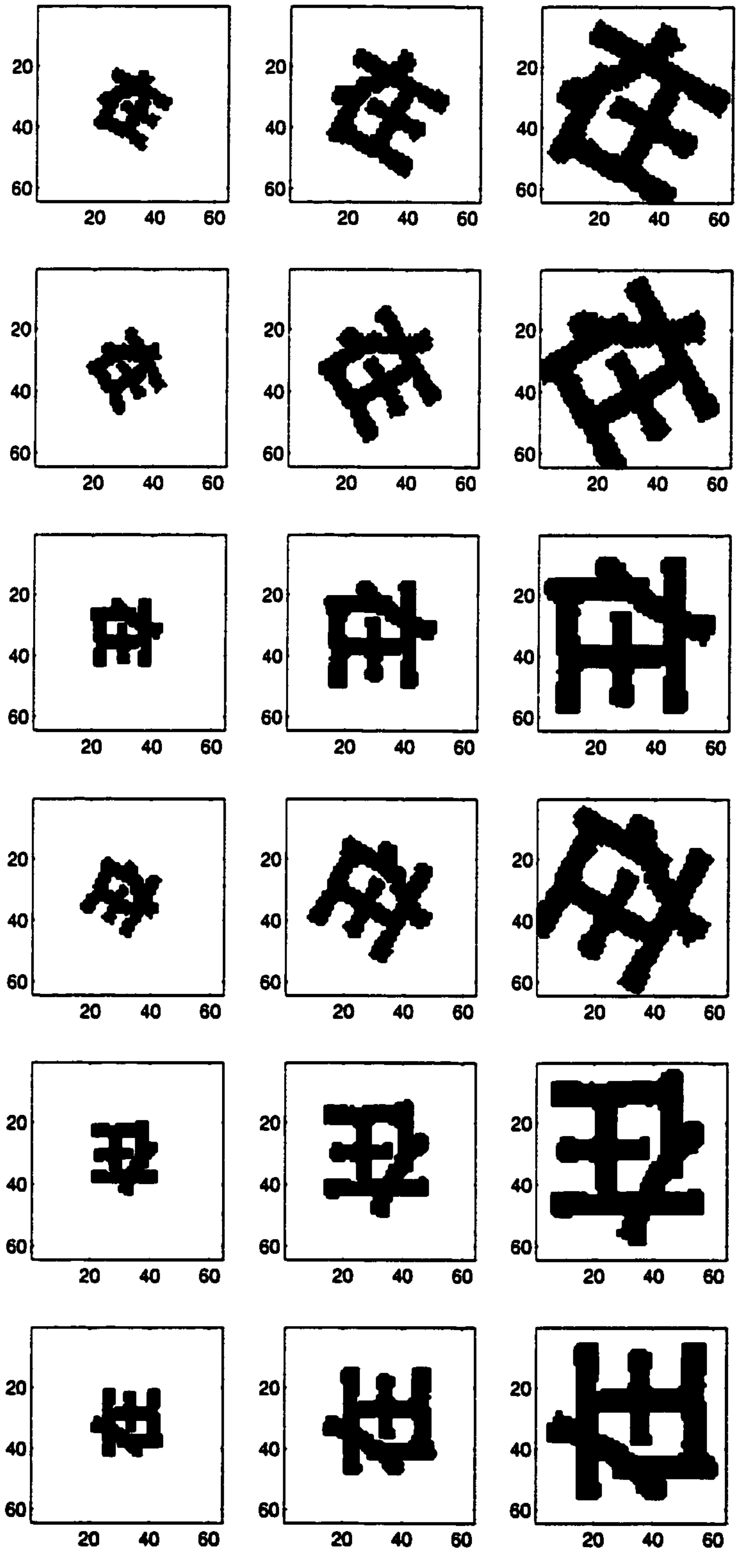

Figure 10: A combination of rotation and scaling factors for character "zai"

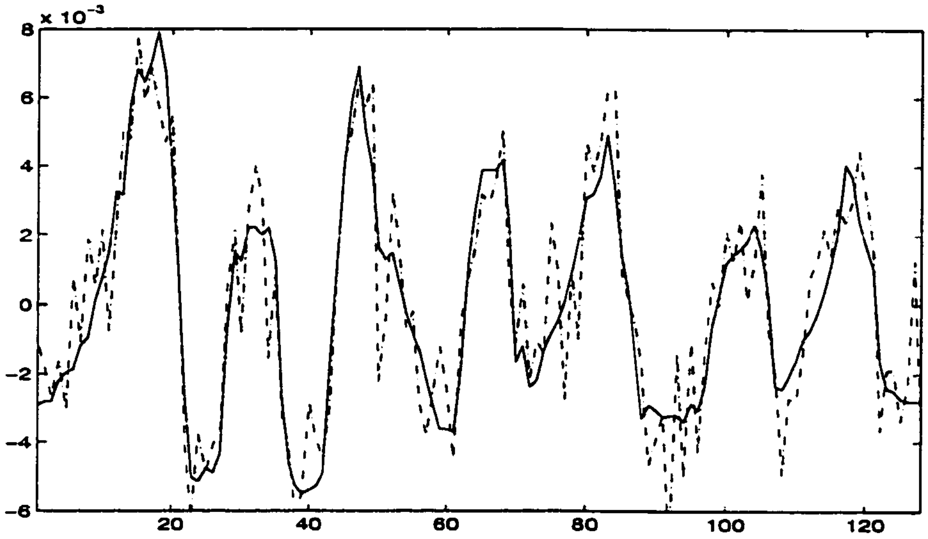

Figure 11: A periodic signal: with (dash dot line) and without (solid line) noises

| Descriptor | Scaling Factor | | | | | |
|---|---|---|---|---|---|---|
| | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.2 |
| Haar | 97.65 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 97.65 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 97.65 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 1: The recognition rates of the Orthonormal-Fourier descriptor for different scaling factors. The features used in this experiment are from scales $d^4$, $d^5$ and $s^5$. For each resolution scale, only the first 20 low frequency Fourier coefficients are used.

| Descriptor | Scaling Factor | Rotation Angle | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30° | 60° | 90° | 120° | 180° | 270° |
| Haar | 0.5 | 97.65 % | 94.12 % | 90.59 % | 83.53 % | 91.76 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 0.5 | 97.65 % | 95.29 % | 90.59 % | 85.88 % | 92.94 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 0.5 | 94.12 % | 87.06 % | 87.06 % | 87.06 % | 94.71 % | 96.47 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 0.5 | 94.12 % | 88.24 % | 88.24 % | 85.88 % | 99.41 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 2: The recognition rates of the Orthonormal-Fourier descriptor for a combination of rotation angles and scaling factors. The features used in this experiment are from scales $d^4$, $d^5$ and $s^5$. For each scale, only the first 20 low frequency Fourier coefficients are used.

| Descriptor | Scaling Factor | | | | | |
|---|---|---|---|---|---|---|
| | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.2 |
| Haar | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Fourier | 94.12 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 3: The recognition rates of the Orthonormal-Fourier descriptor and the Fourier descriptor for different scaling factors. The features used in the Orthonormal-Fourier descriptor are from scales $d^3$, $d^4$ and $d^5$. For each scale, only the first 20 low frequency Fourier coefficients are used.

| Descriptor | Scaling Factor | Rotation Angle | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30° | 60° | 90° | 120° | 180° | 270° |
| Haar | 0.5 | 97.65 % | 96.47 % | 92.94 % | 87.06 % | 90.59 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 0.5 | 98.82 % | 96.47 % | 92.94 % | 87.06 % | 88.24 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 0.5 | 97.65 % | 97.65 % | 91.76 % | 89.41 % | 85.88 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 0.5 | 98.82 % | 97.65 % | 92.94 % | 88.24 % | 87.06 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Fourier | 0.5 | 97.65 % | 95.29 % | 87.06 % | 85.88 % | 85.88 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 4: The recognition rates of the Orthonormal-Fourier descriptor and the Fourier descriptor for a combination of rotation angles and scaling factors. The features used in the Orthonormal-Fourier descriptor are from scales $d^3$. $d^4$ and $d^5$. For each scale, only the first 20 low frequency Fourier coefficients are used.

| SNR | 40 | 20 | 10 | 5 | 4 | 3 | 2 |
|---|---|---|---|---|---|---|---|
| Recognition Rates | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % | 97.65 % |

Table 5: Recognition rates when noise is added to the 1-D signal but no rotation or scaling of the original pattern is included.

# Chapter 3

# Invariant Pattern Recognition by a Combination of Wavelet and Fourier Transforms

## 3.1 Introduction

Feature extraction [49] is a crucial processing step for pattern recognition, and most research has been devoted to finding measures that concisely represent a pattern and that at the same time contain enough information to ensure reliable recognition. In general, good features must satisfy the following requirements: First, *intraclass variance* must be small, which means that features derived from different samples of the same class should be close(e.g., numerically close if numerical features are selected). Secondly, the *interclass* separation should be large, i.e., that features derived from samples of different classes should differ significantly. Furthermore, features should be independent of the size and location of the patterns. This independence can be achieved by processing or by extracting features that are translation-, rotation-, and scale-invariant. In this chapter, we introduce a second algorithm for invariant pattern recognition based on a combination of wavelet transform and Fourier transform.

## 3.2 Rotation-Invariant Features Derived by a Combination of Wavelet and Fourier Transforms

Consider the rotation of an image through angle $\alpha$. If the rotated image is denoted by $\tilde{f}$, the relationship between the original and rotated images in the same polar coordinates is $\tilde{f}(r,\theta) = f(r,\theta-\alpha)$. The transform from Cartesian coordinate system to polar coordinate system can be done as follows. Let $d$ be the radius of the pattern. We draw $n$ concentric circles centred at the centroid $(\bar{x},\bar{y})$ with radius $\frac{d \times i}{n}, i = 1,2,...,n$. Also, we form $n$ angularly equi-spaced radial vectors $\theta_i$ departing from $(\bar{x},\bar{y})$ with angular step $\frac{2\pi}{n}$. For any small region:

$$S_{ij} = \{(r,\theta)|\ r_i < r \leq r_{i+1}, \theta_j < \theta \leq \theta_{j+1}\}$$

we calculate the average value of $f(x,y)$ over this region, and assign the average value to $g(r,\theta)$ in the polar coordinate system.

The replacement of Cartesian coordinate system to polar coordinate system does not add any deformation to the original rotative distortion. The sample patterns of $f(x,y)$ and $g(r,\theta)$ are different, which result in two different matrices. However, since both $f(x,y)$ and $g(r,\theta)$ are discrete formats of the same image function, regardless of digitisation error, a replacement between them should not affect the geometric nature of the image function itself. Hence, the change of coordinate system has transformed the rotation into a translation along the axis polar angle $\theta$. In order to eliminate rotational variance, we apply 1-D Fourier transform along the axis of the polar angle $\theta$ of $g(r,\theta)$ to obtain its spectrum $G(r,\varphi)$. Since the spectra of Fourier transform of circularly shifted signals are the same, we obtain a feature which is also rotation invariant.

As we know the wavelet coefficients represent pattern features in an efficient way, so we can apply wavelet transform along the axis of radius of the resulting $G(r,\varphi)$. The dydiac wavelet transform decomposes a signal into a set of signals at different resolution scales. The information at the fine scales is strongly affected by noise and quantisation errors which are due to the use of the rectangular grids in digital images. In order to reduce such effects, only a few low resolution scales should be used for distinguishing among objects. This makes the representation robust in a noisy environment and reduces the amount of computation.

The matching task is to recognise an unknown object in an image as one of a number of patterns whose representation are known. The process consists of two phases: learning and classification. In the learning phase, the system constructs the object representation in a noise-free environment. Only a few selected intermediate scales should be used in the matching process. Under uniformly distributed noise, the signal to noise ratio at these scales is improved and the effect of noise is reduced significantly. Therefore, choosing a suitable number of scales makes the representation more robust. In the classification phase, the representation of unknown object is compared with all the objects in the database using a distance metric. An object is recognised as one of the known patterns if the distance between the object and that pattern is the smallest.

The idea of polarisation followed by a combination of wavelet and Fourier transforms can be accomplished in two ways by interchanging the order of wavelet transform and Fourier transform. We call the Polarisation-Fourier-Wavelet process as the PFW algorithm. Instead, we call the Polarisation-Wavelet-Fourier process as the PWF algorithm.

The steps of the PFW algorithm can be summarised as follows:

1. Normalise the pattern $f(x, y)$ so that it is translation- and scale-invariant.

2. Transform $f(x, y)$ into polar coordinate system to obtain $g(r, \theta)$.

3. Conduct 1-D Fourier transform on $g(r, \theta)$ along the axis of polar angle $\theta$ and obtain its spectrum:

$$G(r, \varphi) = |FT_\theta(g(r, \theta))|.$$

4. Apply 1-D wavelet transform on $G(r, \varphi)$ along the axis of radius $r$:

$$WF(r, \varphi) = WT_r(G(r, \varphi)).$$

5. Use the wavelet coefficients to query the pattern feature database at different scales.

The steps of the PWF algorithm can be summarised as follows:

1. Normalise the pattern $f(x, y)$ so that it is translation- and scale-invariant.

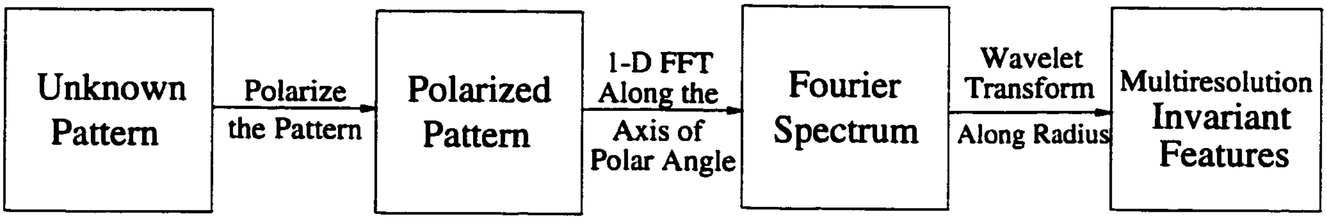
Unknown Pattern
Polarize the Pattern
Polarized Pattern
1-D FFT Along the Axis of Polar Angle
Fourier Spectrum
Wavelet Transform Along Radius
Multiresolution Invariant Features

Figure 12: Block diagram of the PFW algorithm

2. Transform $f(x, y)$ into polar coordinate system to obtain $g(r, \theta)$.

3. Apply 1-D wavelet transform on $g(r, \theta)$ along the axis of radius $r$:

$$WT(r, \theta) = WT_r(g(r, \theta)).$$

4. Conduct 1-D Fourier transform on $WT(r, \theta)$ along the axis of polar angle $\theta$ and obtain its spectrum:

$$FW(r, \varphi) = |FT_\theta(WT(r, \theta))|.$$

5. Use the wavelet coefficients to query the pattern feature database at different scales.

Figure 12 and Figure 14 are the block diagrams of the PFW and PWF algorithms, respectively. Figure 13 depicts how a printed Chinese character is transformed after each step of the PFW algorithm. Figure 13 (a) is the character in $(x, y)$-coordinate system. Figure 13 (b) is the polarised character $g(r, \theta)$ in polar coordinate system where each unit in the axis of the Polar Angle represents 6 degrees. Figure 13 (c) shows the spectrum density of the Fourier transform $G(r, \varphi) = |FT_\theta(g(r, \theta))|$, and Figure 13 (d) shows the wavelet coefficients $WF(r, \varphi) = WT_r(G(r, \varphi))$.

Figure 15 shows how a printed Chinese character is transformed after each step of the PWF algorithm. Figure 15 (a) is the character in $(x, y)$-coordinate system. Figure 15 (b) is the polarised character $g(r, \theta)$ in polar coordinate system where each unit in the axis of the Polar Angle represents 6 degrees. Figure 15 (c) shows the wavelet coefficients $WT(r, \theta) = WT_r(g(r, \theta))$. Figure 15 (d) illustrates the spectrum density of the Fourier transform $FW(r, \varphi) = |FT_\theta(WT(r, \theta))|$.

The computational complexity of the PFW and PWF algorithms can be calculated as follows. Let $n$ be the number of discretized sample points along one axis of the

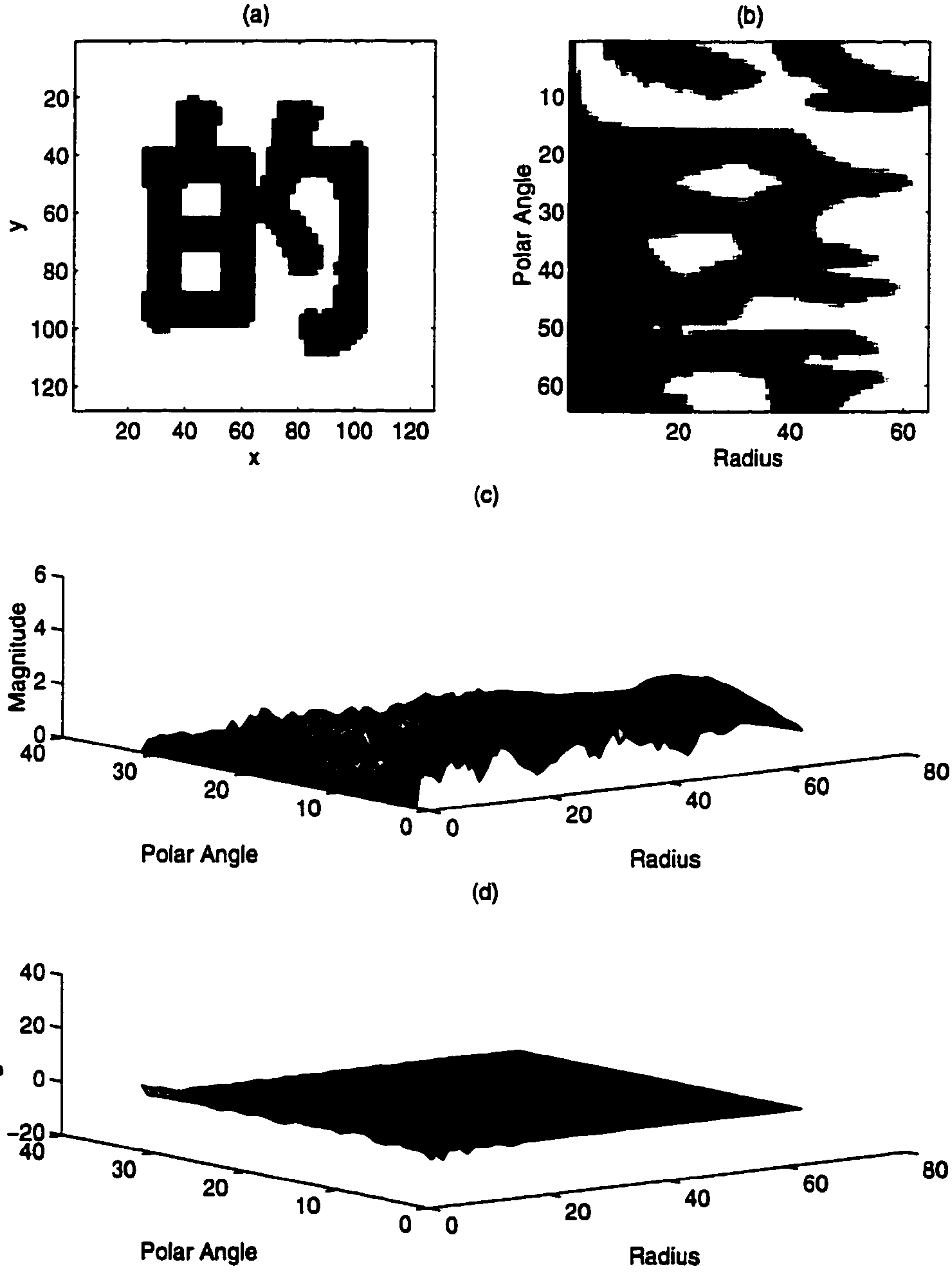

Figure 13: An illustration of how a printed Chinese character is transformed after each step of the PFW algorithm. (a) The original printed Chinese character in Cartesian coordinates (b) The polarised character in polar coordinates where each unit in the axis of the Polar Angle represents 6 degrees (c) The Fourier spectrum of the polarised character (d) The wavelet coefficients based on the Fourier spectrum

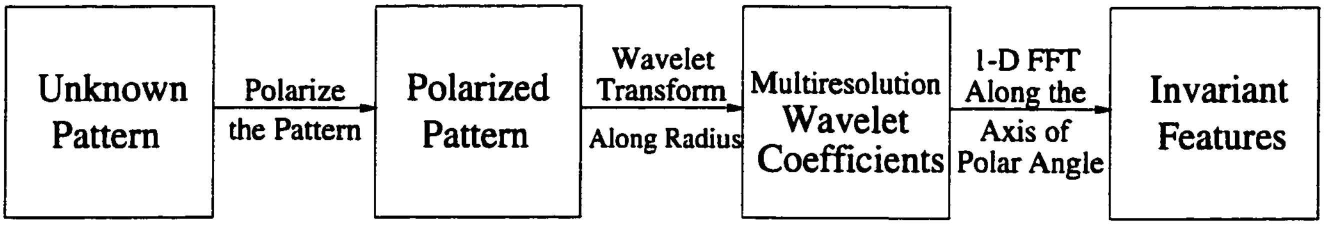
Unknown Pattern
Polarize the Pattern
Polarized Pattern
Wavelet Transform Along Radius
Multiresolution Wavelet Coefficients
1-D FFT Along the Axis of Polar Angle
Invariant Features

Figure 14: Block diagram of the PWF algorithm

original image. we can finish the replacement of Cartesian coordinates into polar coordinates in $O(n^2)$ operations. Since the complexity of FFT on a signal of length $n$ is $O(n \log n)$ and we have $O(n)$ rows in the polarised image $g(r, \theta)$, the Fourier transform along the axis of polar angle $\theta$ takes $O(n^2 \log n)$ operations. As we know, the wavelet transform of a signal of length $n$ takes $O(n)$ operations. We have in total $n$ signals to be processed by wavelet transform. Therefore, the third step of the algorithm requires $O(n^2)$ operations. In summary, the number of operations for the algorithms is

$$O(n^2) + O(n^2 \log n) + O(n^2) = O(n^2 \log n)$$

It is noted that the features extracted by the PFW algorithm are a superset of that of the Ring-Projection approach. Tang et al (1996) introduces the Ring-Projection of a pattern by

$$P(r) = \int_0^{2\pi} f(r\cos\theta, r\sin\theta)d\theta$$

where $r$ is the radius of the ring. It is shown that $P(r)$ is equal to the pattern mass distributed along circular rings.

From Fourier transform we have

$$G(r, \varphi) = \frac{1}{n}\sum_{\theta=0}^{n-1} g(r, \theta)e^{-\frac{i\theta\varphi}{n}}.$$

When $\varphi = 0$, we get the average value along the axis of radius $r$

$$G(r, 0) = \frac{1}{n}\sum_{\theta=0}^{n-1} g(r, \theta).$$

i.e.

$$G(r, 0) = P(r).$$

The PFW algorithm extracts more features from the pattern than the Ring-Projection approach does. Therefore, we can expect that the PFW algorithm gives higher recognition rate.

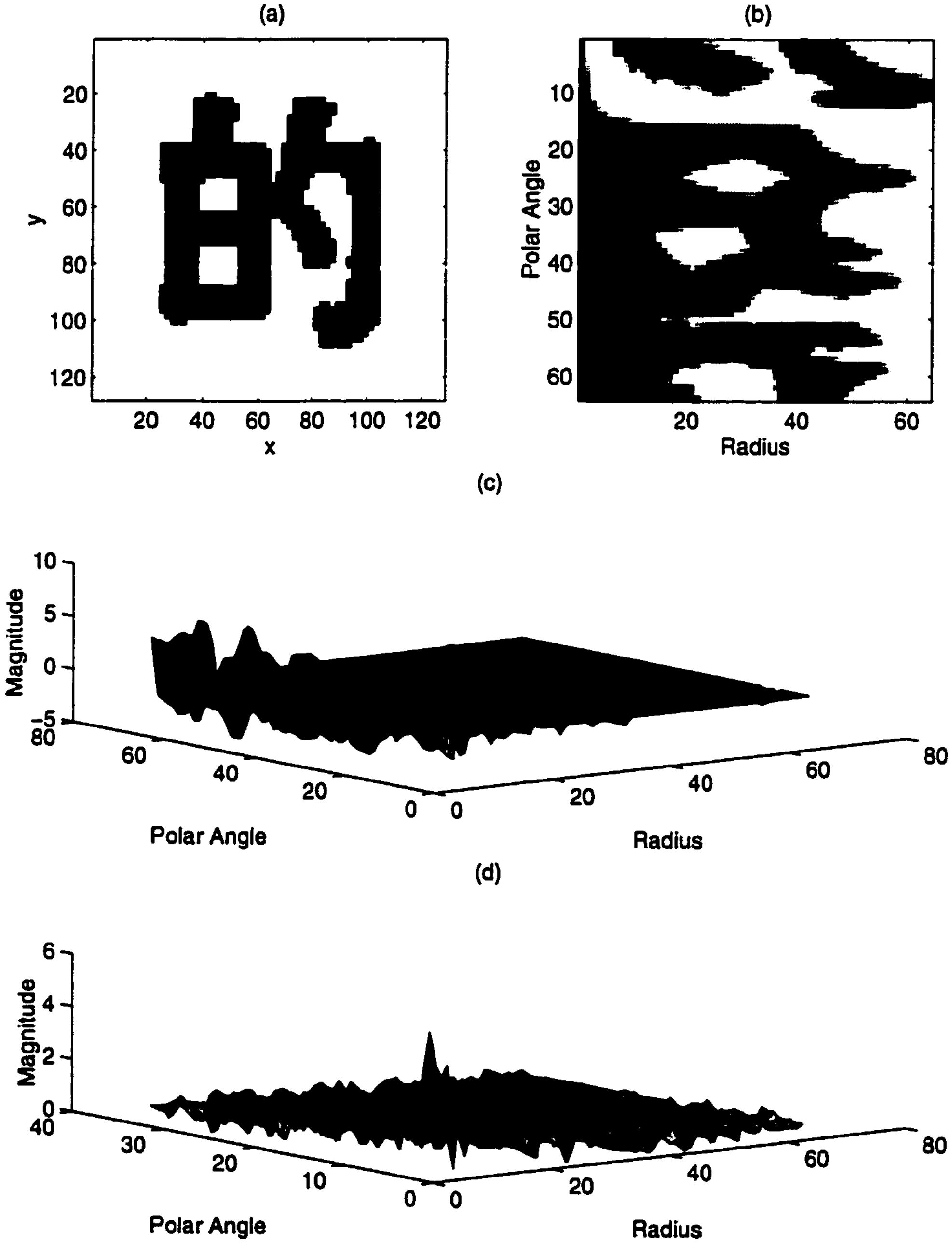

Figure 15: An illustration of how a printed Chinese character is transformed after each step of the PWF algorithm. (a) The original printed Chinese character in Cartesian coordinates (b) The polarised character in polar coordinates where each unit in the axis of the Polar Angle represents 6 degrees (c) The wavelet coefficients after wavelet transform along radius (d) The Fourier spectrum based on the wavelet coefficients

## 3.3 Translation and Scaling Normalisation

As discussed in the previous section, the proposed descriptors are only rotation invariant. However, the considered images have scaling and translation differences as well. Therefore, prior to extraction of features using a combination of wavelet and Fourier transforms, these images should be normalised with respect to scaling and translation.

To achieve scaling and translation uniformity, the regular moments $m_{pq}$ of each image can be utilised. The regular moments $m_{pq}$ are defined as $m_{pq} = \int_{-\infty}^{+\infty} x^p y^q f(x,y)dxdy$. Translation invariance is achieved by transforming the image into a new one whose first order moments, $m_{01}$ and $m_{10}$, are both equal to zero. This is done by transforming the original $f(x,y)$ image into another one which is $f(x+\bar{x}, y+\bar{y})$, where $\bar{x}$ and $\bar{y}$ are the centroid location of the original image computed from

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}}.$$

In other words, the origin is moved to the centroid before moment calculation.

There are three commonly used methods to achieve scaling-invariance. First, Scaling invariance can be accomplished by enlarging or reducing each shape such that its zeroth order moment $m_{00}$ is set equal to a predetermined value $\beta$ [28]. Note that in the case of binary images $m_{00}$ is the total number of shape pixels in the image. Let $f(\frac{x}{a}, \frac{y}{a})$ represent a scaled version of of the image function $f(x,y)$. Then, the regular moment $m_{pq}$ of $f(x,y)$ and $m'_{pq}$, the regular moment of $f(\frac{x}{a}, \frac{y}{a})$, are related by

$$m'_{pq} = \int_x \int_y x^p y^q f(\frac{x}{a}, \frac{y}{a})dxdy = a^{p+q+2} m_{pq}.$$

Since the objective is to have $m'_{00} = \beta$, one can let $a = \sqrt{\frac{\beta}{m_{00}}}$. Substituting $a = \sqrt{\frac{\beta}{m_{00}}}$ into $m'_{00}$, one obtains $m'_{00} = a^2 m_{00} = \beta$. Thus scaling invariance is achieved by transforming the original image function $f(x,y)$ into a new function $f(\frac{x}{a}, \frac{y}{a})$ with $a = \sqrt{\frac{\beta}{m_{00}}}$. Second, we can achieve scaling invariance by setting $a = \max_{f(x,y)\neq 0} \sqrt{(x-\bar{x})^2 + (y-\bar{y})^2}$, the longest distance from $(\bar{x}, \bar{y})$ to a point $(x,y)$ on the pattern. This method, however, is rarely used in noisy environment since it is very sensitive to noise. The third method is to normalise the image so that the average radius $r_0$ for the activated pixels is a quarter of the input grid dimension; i.e., we have

$$r_0 = \frac{1}{m_{00}} \sum_{j=1}^{n} \sum_{k=1}^{n} f(x_j, y_k)\sqrt{x_j^2 + y_k^2},$$

$$a = \frac{n}{4r_0}.$$

In summary, an image function $f(x, y)$ can be normalised with respect to scaling and translation by transforming it into $f(\frac{x}{a} + \bar{x}, \frac{y}{a} + \bar{y})$, with $(\bar{x}, \bar{y})$ being the centroid of $f(x, y)$ and a suitable scaling factor $a$. Wherever $(\frac{x}{a} + \bar{x}, \frac{y}{a} + \bar{y})$ does not correspond to a grid location, the function value associated with it is interpolated from the values of the four nearest grid locations around it.

## 3.4 Replacement of Cartesian Coordinates into Polar Coordinates

There are two issues in the replacement of Cartesian Coordinate system to Polar coordinate system, i.e. the way of mapping coordinates $(x, y)$ to $(r, \theta)$ and determining the value of the new function $g(r, \theta)$.

In the case of continuous functions, such replacement can be achieved by

$$x = r\cos\theta, \quad y = r\sin\theta$$

and there is a one-to-one correspondence between the two functions $g(r, \theta) = f(r\cos\theta, r\sin\theta)$. With regard to discretized coordinate system, however, the task becomes a little bit more complex. Assume the discretized intervals between two consecutive coordinate values in Cartesian coordinate system $(x, y)$ and polar coordinate system $(r, \theta)$ are $(\Delta x, \Delta y)$ and $(\Delta r, \Delta\theta)$, respectively. Then their image pixels are $\Delta x \Delta y$ and $r_k \Delta r \Delta\theta$, as shown by the shaded area in Figure 16, where $r_k = (k + \frac{1}{2})\Delta r$.

The best way to solve this problem is to divide the image pixel into smaller areas, called sub-pixels. In our experiment, we divide image pixel $\Delta x \Delta y$ into $16^2$, denoted by $p^2$, sub-pixels, so that each sub-pixel is small enough to be mapped into a unique pixel $r_k \Delta r \Delta\theta$. With the division of pixels into sub-pixels, we can calculate $g(r_i, \theta_j)$ as the average value of those sub-pixels that fall into region $S_{ij} = \{(r, \theta) | \ r_i < r \leq r_{i+1}, \theta_j < \theta \leq \theta_{j+1}\}$.

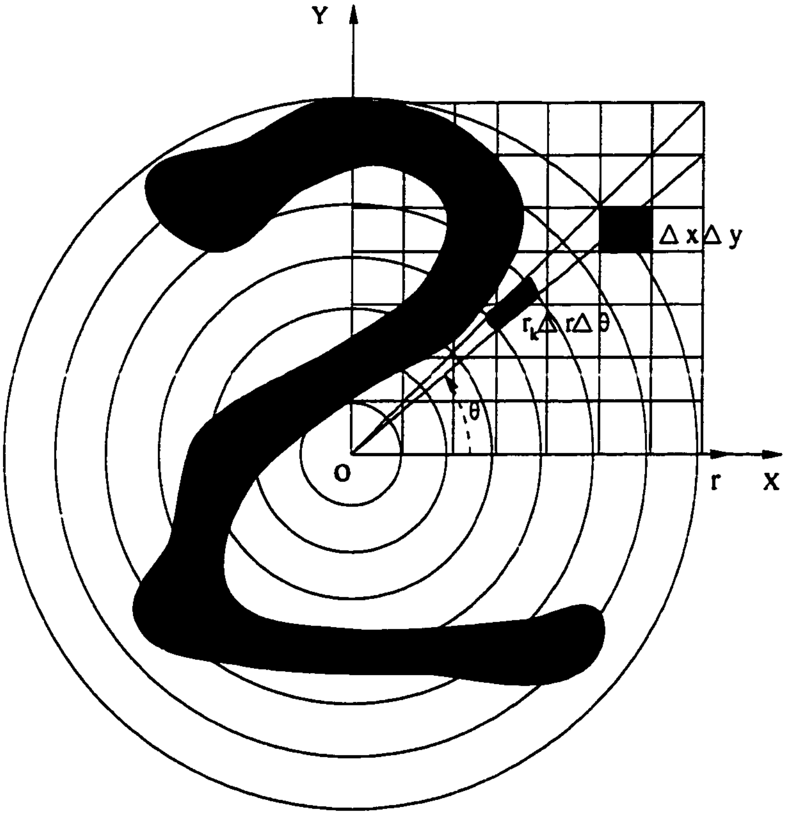

Figure 16: Polarisation of a pattern

## 3.5 Experimental Results

In order to test the efficiency of the PFW and PWF algorithms, we use the same set of 85 printed Chinese characters in our experiment. In Zhang et al. [55], Fourier descriptors together with a new associative memory classifier for recognition were developed and tested on the same set of Chinese characters. The original Chinese character is represented by 64 × 64 pixels, and so is the polarised character. Since the spectrum of 1-D Fourier transform is symmetric, we only keep half of the Fourier coefficients. Therefore, the size of $FW(r,\varphi)$ is $64 \times 32$ and so is the size of the wavelet coefficients $WF(r,\varphi)$.

Because translation will not change the relative position of the centre of mass of the character, our major concern is the system's performance on rotation and scaling. For each character, we tested six rotation angles and six scaling factors. The six rotation angles are 30°, 60°, 90°, 120°, 180° and 270°, and the six different scaling factors are 0.5, 0.6, 0.7, 0.8, 0.9 and 1.2. We use four kinds of wavelet transforms in our experiments, namely, Haar, Daubechies-4, Coiflet-6, and Symmlet-8. We obtain 100% recognition rate for all the rotation angles for both the PFW and PWF algorithms.

The wavelet coefficients of a signal have multi-resolution representation of the original signal. The coarse resolution wavelet coefficients normally represent the global shape of the signal, while the fine resolution coefficients represent the details of the signal. Due to noise introduced in the original image and the errors accumulated in the process of polarisation, the detail coefficients are becoming less important than the coarse scale coefficients. Therefore, it is desirable to use low and intermediate resolution wavelet coefficients as a robust feature in the classification phase. We test the PFW algorithm by using Daubechies-4 wavelet and different scales. Table 10 show the experimental results of the PFW algorithm with regard to recognition rates by using features at different scales. It is clear that the average and the low scales carry significant invariant features.

We test the performance of the PFW and PWF algorithms for different wavelets when only features in a few low scales are used. For each wavelet, we test the performance of the algorithms when only scaling is presented or when a combination of scaling and rotation are presented. When only scaling is used, we obtain the following results. Table 6 shows the recognition results of the PFW algorithm when only features $d^5$, $d^6$ and $s^6$ are used. Table 8 illustrates the recognition results of the PFW algorithm when only features $d^4$, $d^5$ and $d^6$ are used. Table 11 tabulates the recognition results for the PWF algorithm when only features $d^4$, $d^5$, $d^6$, and $s^6$ are used. When a combination of rotation and scaling are presented, we get the following experimental results. Table 7 shows the recognition results of the PFW algorithm when only features $d^5$, $d^6$ and $s^6$ are used. Table 9 illustrates the recognition results of the PFW algorithm when only features $d^4$, $d^5$ and $d^6$ are used. Table 12 tabulates the recognition results for the PWF algorithm when only features $d^4$, $d^5$, $d^6$, and $s^6$ are used. These results demonstrate the effectiveness of the feature extraction algorithms against geometric distortion.

We also test the noise tolerance and sensitivity of the PWF algorithm. The noisy images with different orientations are generated by adding white noises to the noise-free images. The signal to noise ratio (SNR) is defined as

$$\mathrm{SNR} = \frac{\sqrt{\sum_{i,j}(f_{i,j} - avg(f))^2}}{\sqrt{\sum_{i,j}(n_{i,j} - avg(n))^2}}$$

| Descriptor | Scaling Factor | | | | | |
|---|---|---|---|---|---|---|
| | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.2 |
| Haar | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 6: The recognition rates of the PFW algorithm for different scaling factors. The features used in this table are $d^5$, $d^6$, and $s^6$.

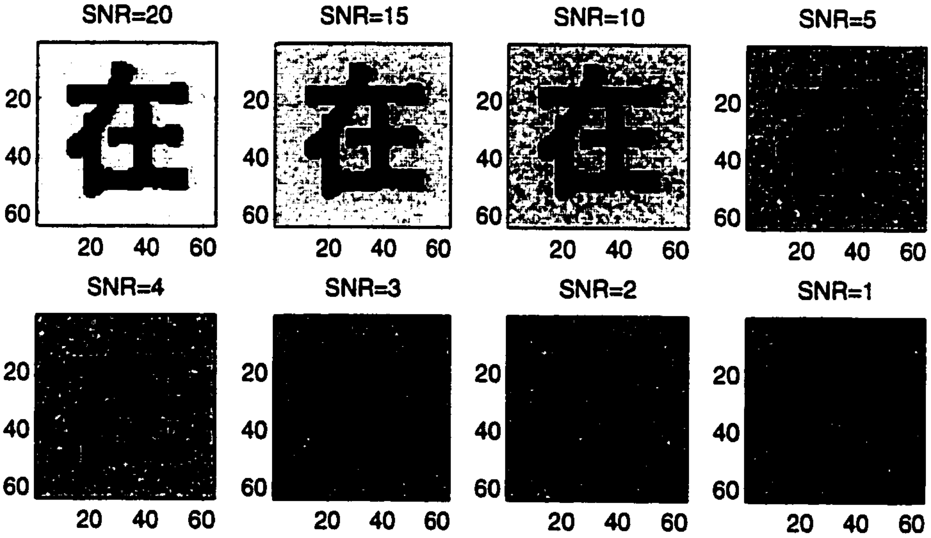

Figure 17: The noisy patterns with SNR = 20, 15, 10, 5, 4, 3, 2, and 1, respectively.

where $f$ is the noise-free image, $n$ is the added white noise, and $avg(f)$ is the average value of the image $f$. Figure 17 shows some noisy patterns for SNR = 20, 15, 10, 5, 4, 3, 2 and 1, respectively. We test the PWF algorithm for SNR = 30, 20, 10, 5 and 2, and for rotation angle = 30°, 60°, 90°, 120°, 180° and 270°. The features $d^4$, $d^5$, $d^6$, and $s^6$ are used in the classification phase. Also, Two different distance metrics, $L_1$ and $L_2$, are employed. The results are tabulated in Table 13 and Table 14, respectively. Our experiments verify that the PWF algorithm is very robust to white noises.

| Descriptor | Scaling Factor | Rotation Angle | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30° | 60° | 90° | 120° | 180° | 270° |
| Haar | 0.5 | 97.65 % | 96.47 % | 92.94 % | 90.59 % | 92.94 % | 95.29 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 0.5 | 97.65 % | 100 % | 96.47 % | 97.65 % | 97.65 % | 95.29 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 0.5 | 98.82 % | 98.82 % | 96.47 % | 96.47 % | 95.29 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 0.5 | 98.82 % | 98.82 % | 96.47 % | 96.47 % | 95.29 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 7: The recognition rates of the PFW algorithm for a combination of rotation angles and scaling factors. The features used in this table are $d^5$, $d^6$, and $s^6$.

| Descriptor | Scaling Factor | | | | | |
|---|---|---|---|---|---|---|
| | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.2 |
| Haar | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 98.82 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 8: The recognition rates of the PFW algorithm for different scaling factors. The features used in this table are $d^4$, $d^5$, and $d^6$.

| Descriptor | Scaling Factor | Rotation Angle | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30° | 60° | 90° | 120° | 180° | 270° |
| Haar | 0.5 | 100 % | 100 % | 98.82 % | 100 % | 97.65 % | 96.47 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 98.82 % | 100 % | 100 % |
| Daubechies 4 | 0.5 | 100 % | 100 % | 100 % | 98.82 % | 100 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 0.5 | 100 % | 100 % | 100 % | 100 % | 98.82 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 0.5 | 100 % | 100 % | 100 % | 100 % | 100 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 98.82 % | 100 % | 100 % |

Table 9: The recognition rates of the PFW algorithm for a combination of rotation angles and scaling factors. The features used in this table are $d^4$, $d^5$, and $d^6$.

| Resolution Scales | Scaling Factor | Rotation Angle | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30° | 60° | 90° | 120° | 180° | 270° |
| $d^2$- $d^6$<br>$s^6$ | 0.5 | 98.82 % | 100 % | 97.65 % | 98.82 % | 92.94 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 98.82 % | 96.47 % | 98.82 % | 94.12 % | 96.47 % |
| $d^3$- $d^6$<br>$s^6$ | 0.5 | 100 % | 100 % | 98.82 % | 98.82 % | 98.82 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| $d^4$- $d^6$<br>$s^6$ | 0.5 | 100 % | 100 % | 100 % | 98.82 % | 100 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| $d^5$- $d^6$<br>$s^6$ | 0.5 | 98.82 % | 100 % | 96.47 % | 97.65 % | 95.29 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| $d^6$<br>$s^6$ | 0.5 | 97.65 % | 97.65 % | 94.12 % | 90.59 % | 91.76 % | 95.25 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 98.82 % | 100 % |
| | 1.2 | 100 % | 100 % | 98.82 % | 98.82 % | 96.47 % | 100 % |
| $s^6$ | 0.5 | 95.29 % | 89.41 % | 77.65 % | 78.82 % | 68.24 % | 85.88 % |
| | 0.8 | 100 % | 100 % | 97.65 % | 95.29 % | 85.88 % | 96.47 % |
| | 1.2 | 100 % | 100 % | 95.29 % | 94.12 % | 89.41 % | 97.65 % |
| $d^4$- $d^6$ | 0.5 | 100 % | 100 % | 100 % | 98.82 % | 100 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| $d^5$- $d^6$ | 0.5 | 100 % | 100 % | 100 % | 97.65 % | 96.47 % | 97.65 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 98.82 % | 98.82 % | 100 % |
| $d^6$ | 0.5 | 94.12 % | 91.76 % | 88.24 % | 78.82 % | 80.00 % | 89.41 % |
| | 0.8 | 100 % | 100 % | 100 % | 98.82 % | 97.65 % | 100 % |
| | 1.2 | 94.12 % | 92.94 % | 94.12 % | 85.88 % | 85.88 % | 89.41 % |

Table 10: The recognition rates of the PFW algorithm by using Daubechies-4 wavelet and features at different scales

| Descriptor | Scaling Factor | | | | | |
|---|---|---|---|---|---|---|
| | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.2 |
| Haar | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Daubechies 4 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Symmlet 8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 11: The recognition rates of the PWF algorithm for different scaling factors. The features used in this table are $d^4$, $d^5$, $d^6$, and $s^6$.

| Descriptor | Scaling Factor | Rotation Angle | | | | | |
|---|---|---|---|---|---|---|---|
| | | 30° | 60° | 90° | 120° | 180° | 270° |
| Haar | 0.5 | 100 % | 100 % | 98.82 % | 100 % | 98.82 % | 98.82 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 98.82 % | 100 % |
| Daubechies 4 | 0.5 | 100 % | 100 % | 100 % | 100 % | 98.82 % | 100 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| Coiflet 3 | 0.5 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 98.82 % | 98.82 % |
| Symmlet 8 | 0.5 | 100 % | 100 % | 100 % | 100 % | 98.82 % | 100 % |
| | 0.8 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |
| | 1.2 | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % |

Table 12: The recognition rates of the PWF algorithm for a combination of rotation angles and scaling factors. The features used in this table are $d^4$, $d^5$, $d^6$, and $s^6$.

| SNR | Rotation Angle | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0° | 30° | 60° | 90° | 120° | 180° | 270° |
| 30 | 100 % | 96.47 % | 100 % | 100 % | 100 % | 100 % | 97.65 % |
| 20 | 100 % | 97.65 % | 97.65 % | 100 % | 100 % | 100 % | 100 % |
| 10 | 100 % | 96.47 % | 96.47 % | 100 % | 100 % | 100 % | 98.82 % |
| 5 | 100 % | 96.47 % | 97.65 % | 100 % | 100 % | 100 % | 100 % |
| 2 | 97.65 % | 95.29 % | 97.65 % | 97.65 % | 96.47 % | 97.65 % | 96.47 % |

Table 13: The recognition rates of the PWF algorithm for different SNR's. The features used in this table are $d^4$, $d^5$, $d^6$ and $s^6$, and the distance metric is $L_1$.

| SNR | Rotation Angle | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0° | 30° | 60° | 90° | 120° | 180° | 270° |
| 30 | 100 % | 96.47 % | 96.47 % | 100 % | 100 % | 100 % | 98.82 % |
| 20 | 100 % | 96.47 % | 96.47 % | 98.82 % | 100 % | 100 % | 98.82 % |
| 10 | 100 % | 95.29 % | 96.47 % | 100 % | 100 % | 100 % | 98.82 % |
| 5 | 100 % | 97.65 % | 96.47 % | 98.82 % | 100 % | 100 % | 98.82 % |
| 2 | 100 % | 97.65 % | 95.29 % | 96.47 % | 97.65 % | 100 % | 98.82 % |

Table 14: The recognition rates of the PWF algorithm for different SNR's. The features used in this table are $d^4$, $d^5$, $d^6$ and $s^6$, and the distance metric is $L_2$.

# Chapter 4

# Translation-Invariant De-noising Using Multiwavelets

## 4.1 Introduction

In the second area of applications, we will discuss the use of translation invariant multiwavelet transform in signal de-noising. This is a very active research area pioneered by recent work of Donoho and his co-workers at Stanford. In de-noising, single orthogonal wavelets with single-mother wavelet function have played an important role (see [14] - [16]). The pioneering work of Donoho and Johnstone ([14], [15]) can be summarised as follows: Let $g(t)$ be the noise-free signal and $f(t)$ the signal corrupted with white noise $z(t)$, i.e., $f(t) = g(t) + \sigma z(t)$, where $z(t)$ has a normal distribution $N(0, 1)$. Donoho and his coworkers proposed the following three-step algorithm:

1. Transform the noisy signal $f(t)$ into an orthogonal domain by discrete single wavelet transform.
2. Apply soft or hard thresholding to the resulting wavelet coefficients by using the threshold $\lambda = \sqrt{2\sigma^2 \log n}$.
3. Perform inverse discrete single wavelet transform to obtain the de-noised signal.

The de-noising is done only on the detail coefficients of the wavelet transform. It has been shown that this algorithm offers the advantages of smoothness and adaptation. However, as Coifman and Donoho [13] pointed out, this algorithm exhibits visual artifacts: Gibbs phenomena in the neighbourhood of discontinuities. Therefore, they propose in [13] a translation invariant (TI) de-noising scheme to suppress such artifacts by averaging over the de-noised signals of all circular shifts. The experimental results in [13] confirm that TI single wavelet de-noising performs better than the traditional single wavelet de-noising. However, it is known that there is a limitation for the time-frequency localisation of a single wavelet functions. Multiwavelets have recently been developed by using translates and dilates of more than one mother wavelet functions ([47]-[53]). They are known to have several advantages over single wavelets such as short support, orthogonality, symmetry, and higher order of vanishing moments. In addition, Strela et al. [47] claimed that multiwavelet soft thresholding offers better results than the traditional single wavelet soft thresholding. Since TI single wavelet de-noising also has better performance than the traditional single wavelet de-noising, it is natural to attempt TI multiwavelet de-noising and compare the results with TI single wavelet de-noising.

The organisation of this chapter is as follows. Section 4.2 gives a short introduction to multiwavelets. Section 4.3 explains how TI multiwavelet de-noising works. Section 4.4 discusses univariate thresholding and bivariate thresholding, respectively. And finally section 4.5 shows some experimental results.

## 4.2 Discrete Multiwavelet Transform

Wavelet transform plays an important role in de-noising. This is because most of the noise will accumulate at the detail scale coefficients of the wavelet transform. In this section, we will give a short introduction to multiwavelet transform. Multiwavelets are generalisation of single wavelets. Multiwavelet basis uses translations and dilations of $M \geq 2$ scaling functions $\{\phi_k(x)\}_{1\leq k\leq M}$ and $M$ mother wavelet functions $\{\psi_k(x)\}_{1\leq k\leq M}$. If we write $\Phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_M(x))^{\mathrm{T}}$ and $\Psi(x) = (\psi_1(x), \psi_2(x), \ldots, \psi_M(x))^{\mathrm{T}}$, then we have

$$\Phi(x) = 2\sum_{k=0}^{L-1} H_k \Phi(2x-k), \tag{41}$$

and

$$\Psi(x) = 2\sum_{k=0}^{L-1} G_k \Phi(2x-k). \tag{42}$$

where $\{H_k\}_{0\leq k\leq L-1}$ and $\{G_k\}_{0\leq k\leq L-1}$ are $M\times M$ filter matrices. The scaling functions $\phi_i(x)$ and associated wavelets $\psi_i(x)$ are constructed so that all the integer translations of $\phi_i(x)$ are orthogonal, and the integer translations and the dilations of factor 2 of $\psi_i(x)$ form an orthonormal basis for $L^2(R)$.

As an example, for $M = 2$, $L = 4$, we give the most commonly used multiwavelets developed by Geronimo, Hardin and Massopust [19]. Let

$$H_0 = \begin{pmatrix} 3/10 & 2\sqrt{2}/5 \\ -\sqrt{2}/40 & -3/20 \end{pmatrix}, \quad H_1 = \begin{pmatrix} 3/10 & 0 \\ 9\sqrt{2}/40 & 1/2 \end{pmatrix},$$

$$H_2 = \begin{pmatrix} 0 & 0 \\ 9\sqrt{2}/40 & -3/20 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 0 & 0 \\ -\sqrt{2}/40 & 0 \end{pmatrix},$$

and

$$G_0 = \begin{pmatrix} -\sqrt{2}/40 & -3/20 \\ -1/20 & -3\sqrt{2}/20 \end{pmatrix}, \quad G_1 = \begin{pmatrix} 9\sqrt{2}/40 & -1/2 \\ 9/20 & 0 \end{pmatrix},$$

$$G_2 = \begin{pmatrix} 9\sqrt{2}/40 & -3/20 \\ -9/20 & 3\sqrt{2}/20 \end{pmatrix}, \quad G_3 = \begin{pmatrix} -\sqrt{2}/40 & 0 \\ 1/20 & 0 \end{pmatrix}.$$

then the two functions $\phi_1(x)$ and $\phi_2(x)$ can be generated via (41). Similarly, the two mother wavelet functions $\psi_1(x)$ and $\psi_2(x)$ can be constructed by (42). Let $V_J$ be the closure of the linear span of $2^{J/2}\phi_l(2^Jx-k), l = 1,2;\ k \in Z$. With the above constructions, it has been proved that $\phi_l(x-k)$, $l = 1,2;\ k \in Z$ form an orthonormal basis for $V_0$, and moreover the dilations and translations $2^{j/2}\psi_l(2^jx-k)$, $l = 1,2;\ j,k \in Z$ form an orthonormal basis for $L^2(R)$ [13]. In other words, the spaces $V_j$, $j \in Z$, form an orthogonal multiresolution analysis of $L^2(R)$. The two scaling functions $\phi_1(x)$ and $\phi_2(x)$ are supported in $[0,1]$ and $[0,2]$, respectively. They are also symmetric and Lipschitz continuous. This is impossible to achieve for single orthogonal wavelets.

The original noisy signal $f(t)$ should be first discretized into the vector $\{f_i\}_{1\leq i\leq 2^J}$, where $n = 2^J$ = signal length, and prefiltered before it can be used as input of the discrete multiwavelet transform (DMWT). While single wavelet transform has one input stream of size $n \times 1$ which is provided by $\{f_i\}_{1\leq i\leq 2^J}$, multiwavelet transform requires input that consists of $M$ streams each of size $n \times 1$. Therefore a method of mapping the data $\{f_i\}_{1\leq i\leq 2^J}$ to the multiple streams has to be developed. This mapping process is called preprocessing and is done by a prefilter or a repeated signal filter [12]. We will return to the prefilter later in this section.

For the sake of clarity, we use $\{S_{0,k}\}_{0\leq k\leq 2^J-1}$ to denote the multiple streams obtained by applying a prefilter to the original discretized signal $\{f_i\}_{1\leq i\leq 2^J}$. Also, we use $S_{j,k}$ and $D_{j,k}$, which are periodic in $k$ with period $2^{j-1}$, to represent the low-pass and high-pass coefficients. The forward and inverse DMWT can be recursively calculated by:

$$S_{j+1,k} = \sqrt{2}\sum_{n=0}^{L-1} H_n S_{j,(n+2k)mod(2^J)}, \tag{43}$$

$$D_{j+1,k} = \sqrt{2}\sum_{n=0}^{L-1} G_n S_{j,(n+2k)mod(2^J)}, \tag{44}$$

and

$$S_{j,2k+p} = \sqrt{2}\sum_{n=0}^{L/2-1} (H^{\mathrm{T}}_{2n+p} S_{j+1,(n+k)mod(2^{J+1})} + G^{\mathrm{T}}_{2n+p} D_{j+1,(n+k)mod(2^{J+1})}), \tag{45}$$

for $j = 0, 1, \ldots, J-1$; $p = 0, 1$; $k = 0, 1, \ldots$ It is noted that Eq.(45) is different from Eq.(3.7) of [53]. A simple verification can show that the inverse DMWT in Eq.(3.7) of [53] is incorrect. For example, let us consider reconstructing $S_{j,0}$ from $\{S_{j+1,k}\}_k$ and $\{D_{j+1,k}\}_k$. Before passing through the synthesis filters, we have to upsample (i.e. insert zeros) $\{S_{j+1,k}\}_k$ and $\{D_{j+1,k}\}_k$ first. Since the upsampled version is 0 in every other sample, $S_{j,0}$ can only be the sum of the form

$$S_{j,0} = \sqrt{2}(H_0^T S_{j+1,0} + G_0^T D_{j+1,0} + H_2^T S_{j+1,1} + G_2^T D_{j+1,1} + \cdots),$$

that is, with coefficients of $H_0$, $H_2$, $H_4$, ... and similarly with G. However, the equation in [53] has terms with $H_0$, $H_1$, $H_2$, $H_3$, ...

Because a given signal consists of one stream but the DMWT algorithm requires that the input data be multiple streams, a method of mapping the data to the multiple streams has to be developed. This mapping process is called preprocessing and is done by a *prefilter Q* [12]. A *postfilter P* just does the opposite, i.e., mapping the data from multiple streams into one stream. Thus, with $M = 2$,

$$S_{0,k} = \sum_n Q_n \begin{pmatrix} f_{2(n+k)+1} \\ f_{2(n+k)+2} \end{pmatrix}.$$

The postfilter $P$ that accompanies the prefilter $Q$ satisfies $PQ = I$, where $I$ is the identity filter. So, if one applies a prefilter, DMWT, inverse DMWT and postfilter to any sequence the output will be identical to the input. The commonly used prefilters are:

**Identity** prefilter

$$Q_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

**Xia** prefilter

$$Q_0 = \begin{pmatrix} 2+\sqrt{2}/10 & 2-\sqrt{2}/10 \\ \sqrt{2}+3/20 & \sqrt{2}-3/20 \end{pmatrix}.$$

**Minimal** prefilter

$$Q_0 = \begin{pmatrix} 2\sqrt{2} & -\sqrt{2} \\ 1 & 0 \end{pmatrix}.$$

**Repeated signal** prefilter is different from the definition above. It is defined by

$$S_{0,k} = f_{k+1} \begin{pmatrix} \sqrt{2} \\ 1 \end{pmatrix}.$$

It has been shown by Downie and Silverman [12] that the repeated signal filter is very good for de-noising purposes, we also get the same conclusion in our experiments.

Multiwavelets have some advantages in comparison to single ones. For example, such features as short support, orthogonality, symmetry, and higher order of vanishing moments, are known to be important in signal processing. A single wavelet cannot possess all these properties at the same time, but multiwavelets can. It is confirmed that multiwavelets can give better results than the single wavelets in image compression and de-noising [47].

## 4.3 Translation Invariant Multiwavelet De-noising

TI de-nosing suppresses noise by averaging over thresholded signals of all circular shifts. TI table is a fast way of implementation, rather than having to do transform on the original signal $n$ times. We can realize the TI multiwavelet de-noising algorithm by the following steps:

1. Prefilter the original noisy signal into multiple streams by a specific prefilter.

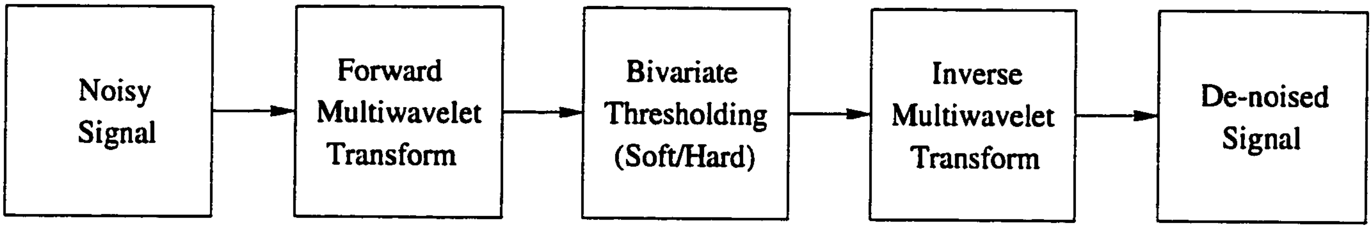
Noisy Signal
Forward Multiwavelet Transform
Bivariate Thresholding (Soft/Hard)
Inverse Multiwavelet Transform
De-noised Signal

Figure 18: The block diagram of the TI multiwavelet de-noising

2. Decompose the multiple streams into a TI table which is similar to the TI table in [13].

3. Apply the univariate or bivariate thresholding (soft/hard) on the TI table.

4. Calculate the de-noised multiple streams from the TI table by reversing the processes in step 2.

5. Postfilter the de-noised multiple streams to get the de-noised signal.

Figure 18 shows the block diagram of the TI multiwavelet de-noising scheme. The TI table in [13] is for discrete single wavelet transform. Below we describe the TI table for discrete multiwavelet transform: The TI table is an $n$ by $WM$ matrix with $0 \leq W \leq \log_2 n$. For convenience, we partition the TI table into $W$ *column groups* with columns $(w-1)M+1$ through $wM$ into *column group* $w$ for $1 \leq w \leq W$. Similarly, the *$w$-th column group* is partitioned into $2^w$ "boxes", where each box is a $n/2^w$ by $M$ matrix. These boxes contain all the multiwavelet coefficients at scale $w$ for different shifts. We also need an $n \times M$ matrix, denoted by $S$, to store the low pass coefficients. This matrix is dynamically filled during each resolution scale.

The fill-in of the TI table and $\alpha$ are fulfilled by a series of decimation and filtering operations. Let $G$ and $H$ stand for the usual down-sampling high pass and low pass operations of the wavelet theory. Also let $R_h$ stand for circular shift by $h$. Set $S_{0,0}$ = the multiple streams by applying a prefilter to $f$, and initialise

$$D_{1,0} = GR_0S_{0,0}; \quad D_{1,1} = GR_1S_{0,0},$$

$$S_{1,0} = HR_0S_{0,0}; \quad S_{1,1} = HR_1S_{0,0}.$$

then, the recursive equations follow

$$D_{j+1,2k} = GR_0S_{j,k}; \quad D_{j+1,2k+1} = GR_1S_{j,k},$$

$$S_{j+1,2k} = HR_0S_{j,k}; \quad S_{j+1,2k+1} = HR_1S_{j,k}.$$

for $j = 1, \ldots, W$ and $k = 0, 1, \ldots, 2^{j-1} - 1$. It should be mentioned that the scale $W$ is normally chosen to be smaller than $\log_2 n$, and thus the scale $j$ does not really go from $j = 1$ up to $j = J - 1$. We should place the vector $D_{w,k}$ in *box* $k$ of *column group* $w + 1$. Also, we need to place the $S_{w,k}$'s in *box* $k$ of the average matrix $S$.

After thresholding on the TI Table, we can reverse the steps during the fill-in process. Let $G^*$ and $H^*$ stand for the usual up-sampling high pass and low pass operations. At scale $j$, for each $k$ in the range $0 \le k \le 2^j - 1$, compute

$$\gamma_k = (R_0G^*S_{j,2k} + R_{-1}G^*S_{j,2k+1})/2, \quad \delta_k = (R_0H^*D_{j,2k} + R_{-1}H^*D_{j,2k+1})/2,$$

and

$$S_{j-1,k} = \gamma_k + \delta_k.$$

for $j = W, \ldots, 1$ and $k = 0, 1, \ldots, 2^{j-1} - 1$. The de-noised multiple stream multiwavelet coefficients are then given by $S_{0,0}$. A postfilter has to be applied in order to get the de-noised signal $\tilde{f}(t)$.

## 4.4 Univariate vs Bivariate Thresholding

The success of wavelet thresholding lies in that usually the signal will be compressed into a few large coefficients whereas the noise component will give rise to small coefficients only. Univariate thresholding is developed by Donoho and Johnstone [15] and two kinds of thresholding methods, *soft* and *hard*, are discussed. The *hard* thresholding will kill all the wavelet coefficients whose magnitudes are less than the threshold to zero while keeping the remaining coefficients unchanged. The *soft* thresholding kills the smaller wavelet coefficients, too. However, all the coefficients whose magnitudes are greater than the threshold will be reduced by the amount of the threshold. Univariate thresholding is successfully used in multiwavelet de-noising by Strela et al. [47].

Even though univariate thresholding will work in multiwavelet de-noising, it does not give enough noise reduction. This is because multiwavelet transform normally produces correlated coefficients. Therefore, we have to use a thresholding method that will treat the multiwavelet coefficient vector as a whole entity. Downie and Silverman developed the bivariate thresholding method by setting the universal threshold to

$\lambda = 2\log n$. The theory behind bivariate thresholding can be briefly reviewed as follows [12].

Suppose we apply the DMWT with an appropriate prefilter to a noisy function, then we get $L$ stream coefficients of the form,

$$D_{j,k} = D^*_{j,k} + E_{j,k}$$

where $E_{j,k}$ has a multivariate normal distributation $N(0, V_j)$. The matrix $V_j$ is the covariance matrix for the error term which depends on the resolution scale j. Using the standard transform

$$\theta_{j,k} = D^T_{j,k} V_j^{-1} D_{j,k}$$

we obtain a positive scalar value which in the absence of any signal component will have a $\chi^2_L$ distribution. It is these values that are thresholded, and the coefficients vectors can then be adapted accordingly. To find the universal threshold, we need to find the asymptotic maximum of $n$ i.i.d. $\chi^2_L$ random variables. If $M_n$ is the maximum of $n$ i.i.d. $\chi^2_p$ variables ($p > 1$) and $\lambda_n$ is the infimum of all sequences $a_n$ that satisfy $P(M_n \leq a_n) \to 1$ as $n \to \infty$, then one can show that

$$\lambda_n = 2\log n + (p-2)\log\log n.$$

For the result to hold, it is sufficient to show that $nP(X > \lambda_n) \to \gamma$ as $n \to \infty$, where $X \sim \chi^2_p$ and $\gamma$ is a positive real number. Using the distribution function for a $\chi^2_p$ distribution and repeated integration by parts, it can be shown that this limit is attained when $\lambda_n = 2\log n + (p-2)\log\log n$. When $p = 2$, the universal threshold for multiple wavelets simplifies to $\lambda_n = 2\log n$. Unlike in the single wavelet thresholding the variance term $\sigma$ does not appear in the universal threshold formula.

Suppose we use a threshold of $\lambda$. Then the *hard* thresholding rule in bivariate thresholding can be written as

$$\hat{D}_{j,k} = \begin{cases} D_{j,k} & \text{if } \theta_{j,k} \geq \lambda \\ 0 & \text{otherwise} \end{cases}$$

where $\theta_{j,k} = D^{\mathrm{T}}_{j,k} V_j^{\mathrm{T}} D_{j,k}$ and $V_j$ is the covariance matrix for the error term depending on the resolution scale $j$. The bivariate *soft* thresholding can be formulated as

$$\hat{D}_{j,k} = \begin{cases} D_{j,k}(1 - \frac{\lambda}{\theta_{j,k}}) & \text{if } \theta_{j,k} \geq \lambda \\ 0 & \text{otherwise} \end{cases}$$

We can use a robust covariance estimation method to estimate $V_j$ directly from the observed coefficients [24]. The pseudo code for robust covariance estimation can be listed as follows:

```
#define mad($y$) 1.4826 * median(abs($y$ - median($y$)))
float $a_1, a_2, b_1, b_2, V_j[2][2]$;
$a_1$        = 1.0/mad ($row_1$);
$a_2$        = 1.0/mad ($row_2$);
$b_1$        = mad ($a_1 * row_1 + a_2 * row_2$);
$b_2$        = mad ($a_1 * row_1 - a_2 * row_2$);
$V_j[1][1]$  = $1/(a_1 * a_1)$;
$V_j[2][2]$  = $1/(a_2 * a_2)$;
$V_j[1][2]$  = $(b_1 - b_2)/((b_1 + b_2) * a_1 * a_2)$;
$V_j[2][1]$  = $V_j[1][2]$;
```

## 4.5 Experimental Results

In our experiments we use the same signals given in [13]: *Blocks*, *Bumps*, *HeaviSine*, and *Doppler*. Gaussian white noise is added to the signals so that the signal-to-noise ratio(SNR) is 7. The SNR is defined as $\sqrt{var(f)/\sigma^2}$, where $var(f)$ is the variance of the signal $f(t)$. The number of sample points for each signal is $n = 2048$. Figure 19 shows the four noise-free signals and the noisy signals. Unless otherwise specified, we use the minimal repeated signal prefilter for the TI multiwavelet de-noising experiments. The coefficients are thresholded using universal univariate threshold $\sqrt{2\sigma^2 \log n}$ and bivariate threshold $2 \log n$. The inverse multiwavelet transform and post-filter are applied to obtain the smoothed estimate of the noise-free signal. All detail scales except the five coarsest scales are thresholded. The mean square error (MSE) is used as the distance measure between the noise-free signal and the de-noised signal.

We compare the performance of the TI GHM multiwavelets with TI Daubechies-4 single wavelets. Figure 20 shows the de-noised signals by using the TI Daubechies-4 wavelets, while Figure 21 illustrates the de-noised signals by using the TI GHM multiwavelets with univariate thresholding. The mean square errors are given in Table 15. It is clear that the TI GHM multiwavelets with soft thresholding gives better results than the TI Daubechies-4 single wavelets. However, the TI GHM multiwavelets with hard thresholding still keeps a lot of noise. As Downie and Silverman [12] pointed out, the DMWT produces correlated coefficients, so a method which accounts for the noise and the signal components within the whole vector should be developed. We implement the TI GHM multiwavelets de-noising with bivariate thresholding [12] and obtain better results. Figure 22 shows the four signals de-noised by using TI GHM multiwavelets with bivariate thresholding, respectively. From Table 15 one can see that TI GHM multiwavelets de-noising with soft bivariate thresholding obtains smaller MSE than both the TI single wavelet de-noising and the TI GHM multiwavelet de-noising with soft univariate thresholding. Also, the TI multiwavelet de-noising with bivariate hard thresholding is better than with univariate hard thresholding. Even though TI multiwavelet de-noising with hard thresholding can suppress more noise than with hard univariate thresholding, it does not always outperform TI single wavelet de-noising. Only *Blocks* and *Doppler* have smaller MSE in our experiments. We also compare the performance of the TI multiwavelet de-noising with the non-TI multiwavelet de-noising by using univariate thresholding and bivariate thresholding, respectively. Generally speaking, TI multiwavelet de-noising is better than non-TI multiwavelet de-noising no matter what thresholding method is used. Furthermore, we can see that bivariate hard thresholding is better than univariate hard thresholding. This confirms the claim by Downie and Silverman. However, bivariate soft thresholding gets bigger MSE for Heavisine and Doppler. Figure 23 and 24 show non-TI univariate de-noising and bivariate de-noising by using GHM multiwavelets.

We test the performance of the TI multiwavelet bivariate thresholding for $SNR = 7$ and different prefilters. The prefilters used in our experiments are Identity, Xia, Minimal, and Repeated row. From Table 16 one can see that the repeated row prefilter gets the smallest MSE for all four signals. We also test the TI multiwavelet bivariate de-noising for different SNR's. Table 17 illustrates the experimental results for $SNR = 3$, 5, 7, and 9. It is meaningful that the TI multiwavelet bivariate de-noising works well when the noise level is high. The comparision between the TI multiwavelet bivariate de-noising and TI single wavelet de-noising is given in Table 18. The single wavelets that are used include Daubechies 4 (D4), Symmelet 8, Haar, and Coiflet 4. TI multiwavelet bivariate de-noising obtains superior performance over all single wavelet de-noising.

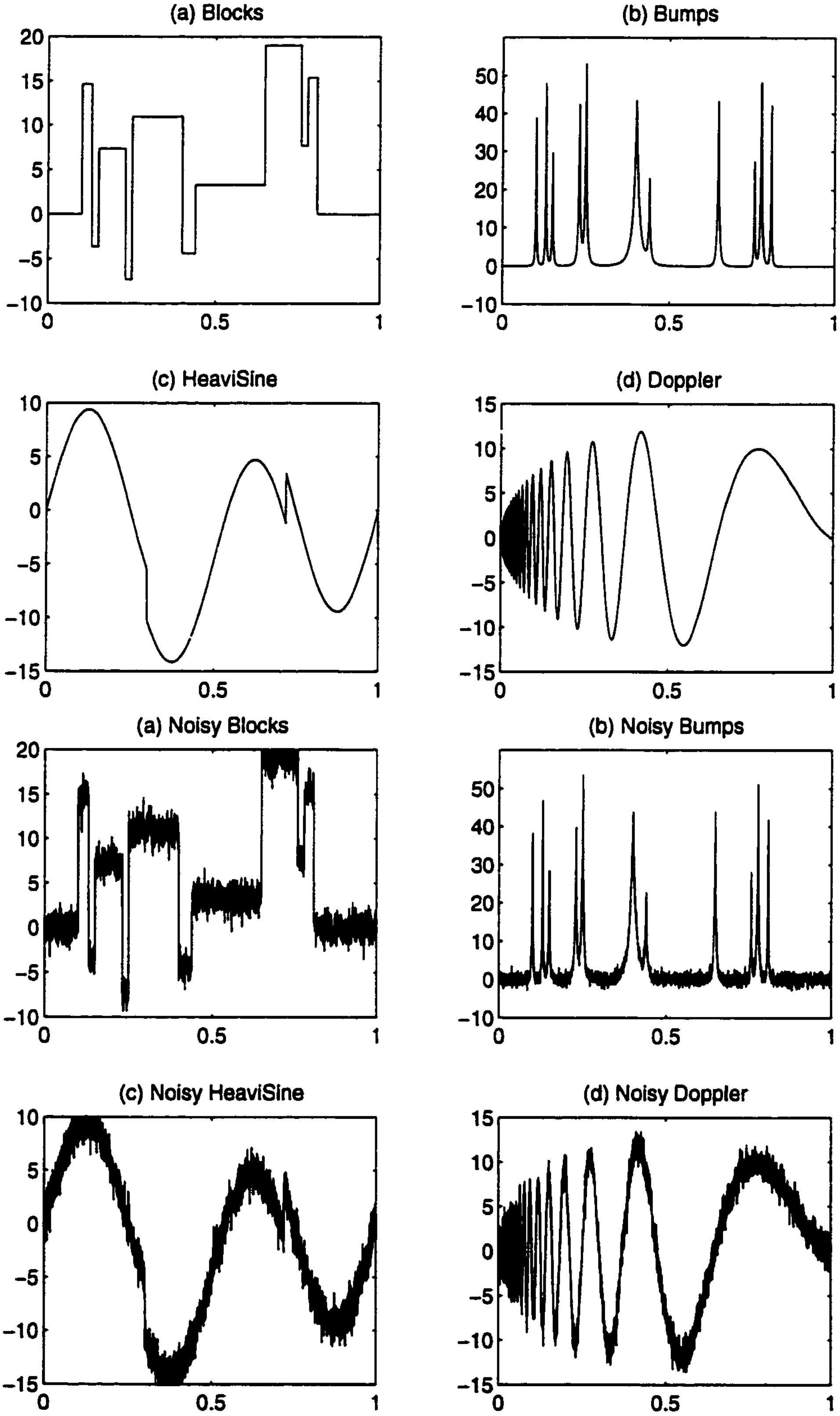

Figure 19: Four noise-free signals and four noisy signals

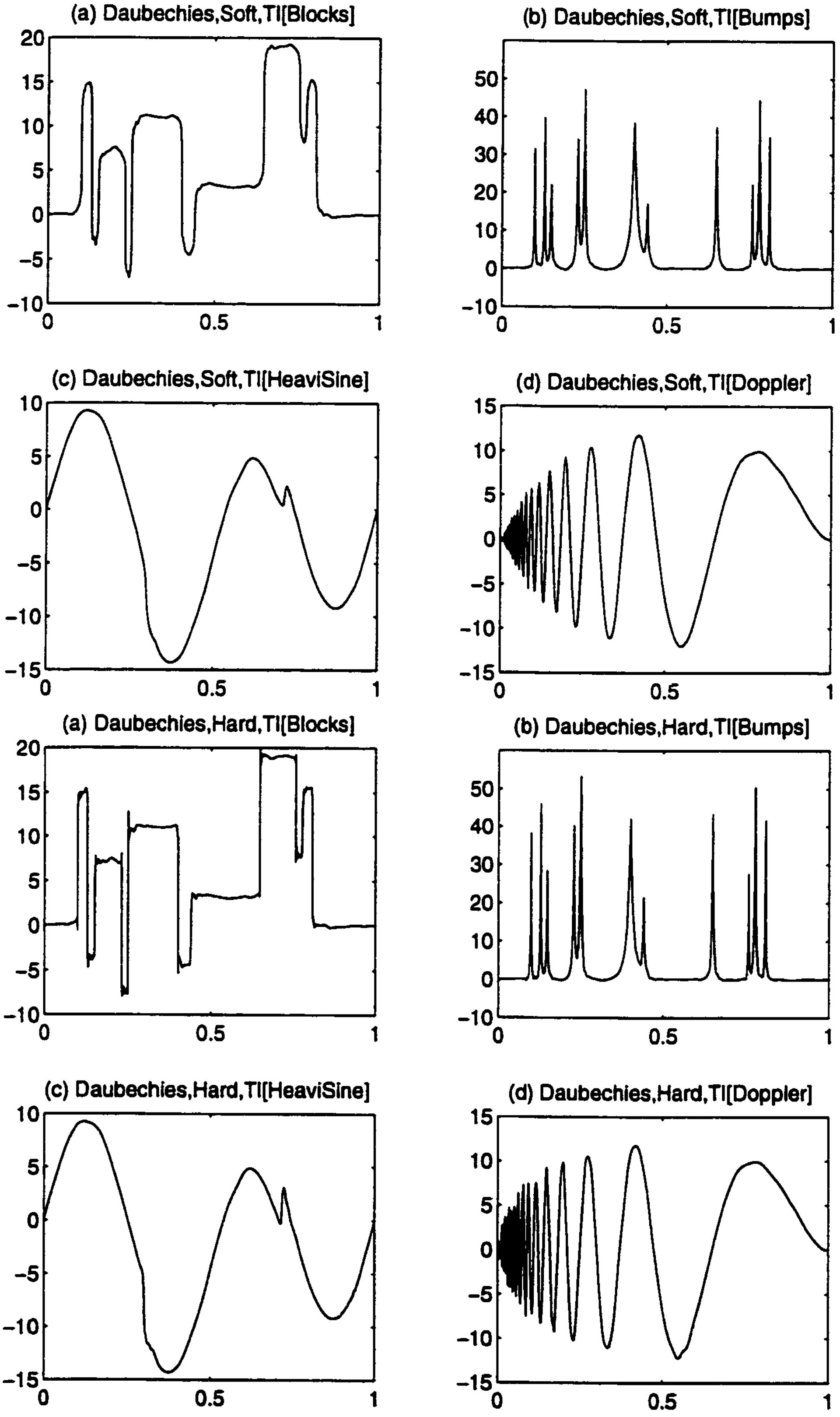

Figure 20: TI D4 Wavelet Shrinkage

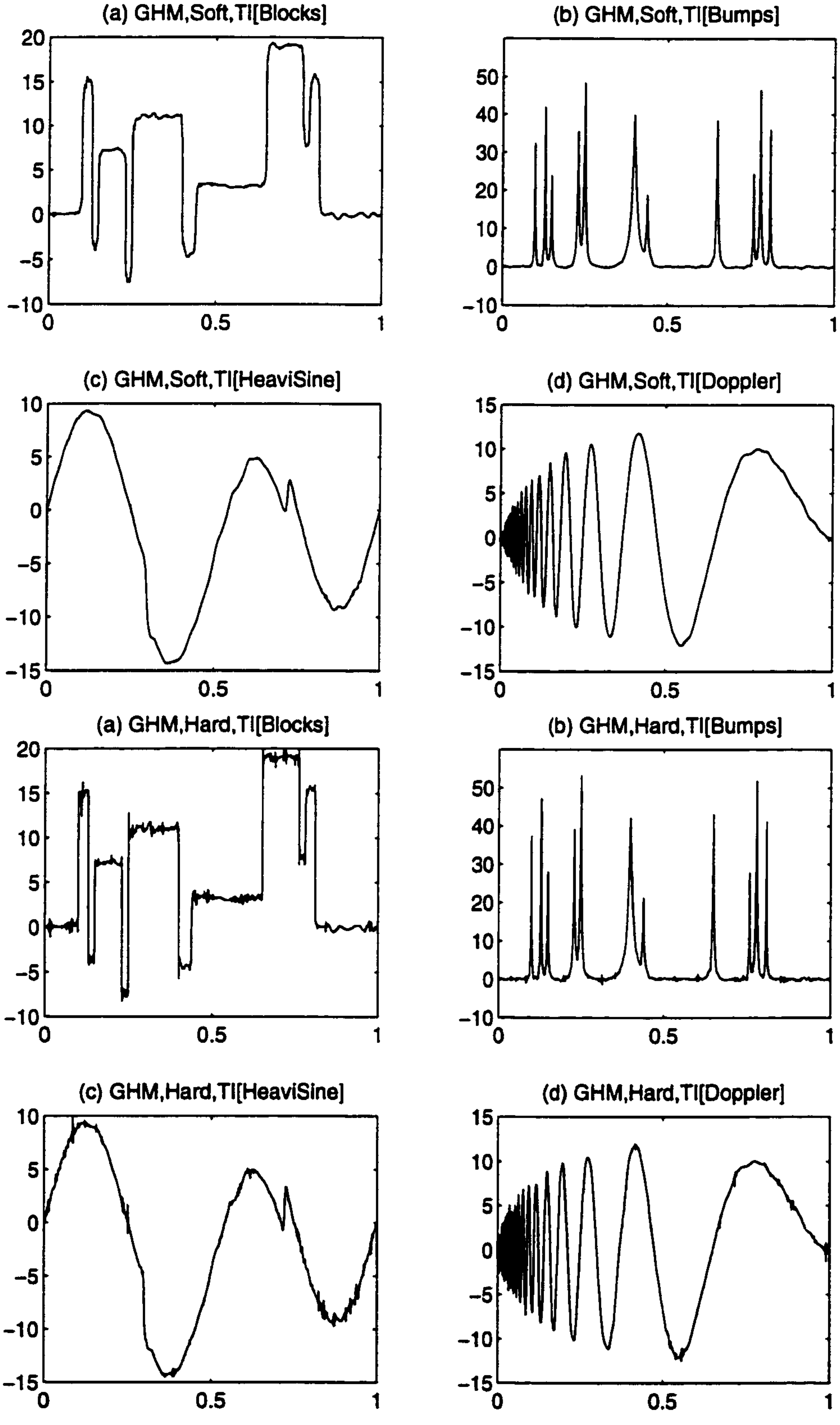

Figure 21: TI GHM MultiWavelet Threshold: Univariate

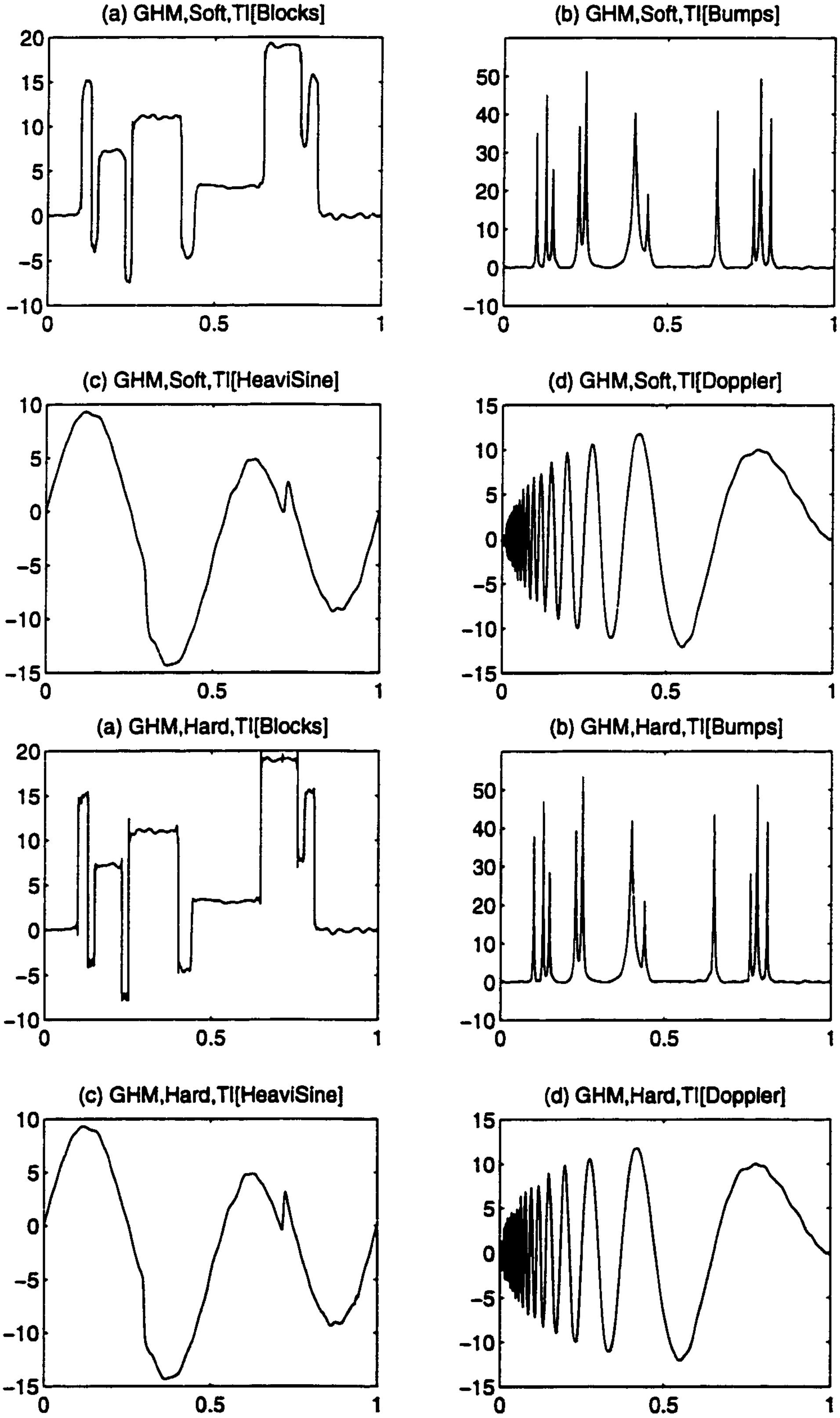

Figure 22: TI GHM MultiWavelet Threshold: Bivariate

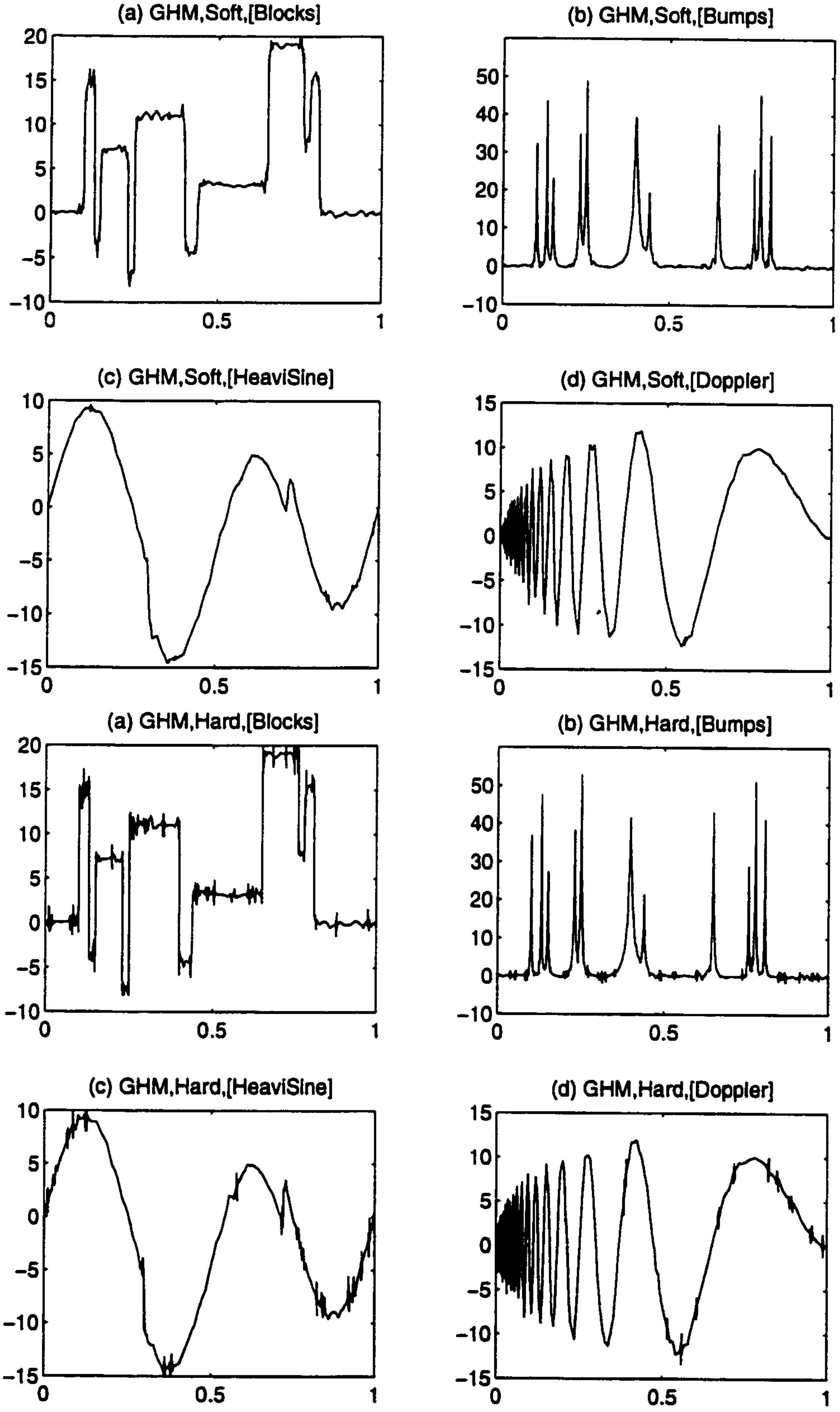

Figure 23: GHM MultiWavelet Threshold: Univariate

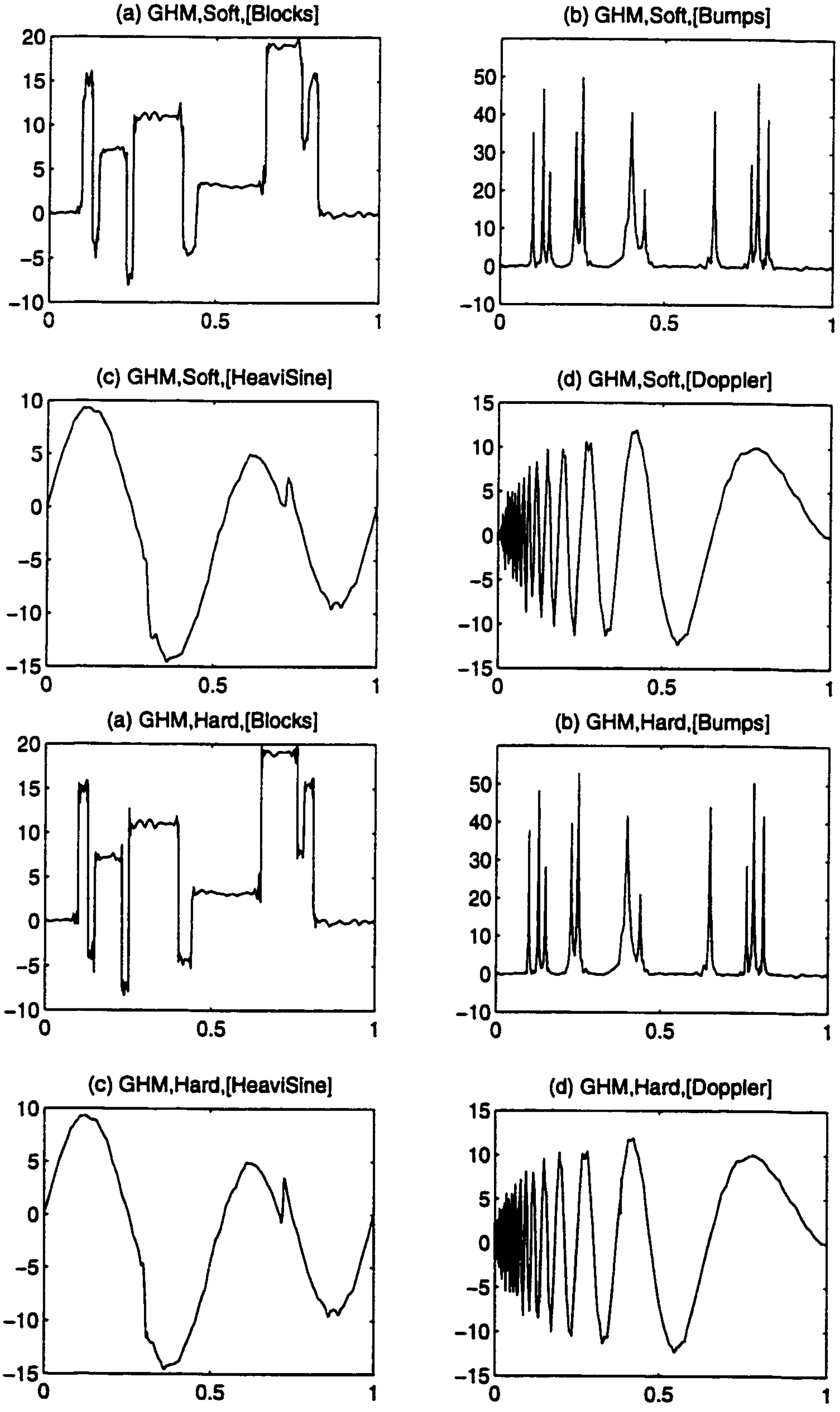

Figure 24: GHM MultiWavelet Threshold: Bivariate

| | | Blocks | Bumps | Heavisine | Doppler |
|---|---|---|---|---|---|
| Soft | TI D4 | 34.175 | 35.173 | 11.838 | 25.383 |
| | TI GHM (Univariate) | 27.108 | 24.965 | **11.157** | 14.869 |
| | TI GHM (Bivariate) | **23.869** | **20.428** | 11.387 | **13.586** |
| | GHM (Univariate) | 29.763 | 32.647 | 12.380 | 19.937 |
| | GHM (Bivariate) | 28.246 | 30.582 | 13.001 | 21.297 |
| Hard | TI D4 | 14.461 | **13.895** | **8.307** | 12.686 |
| | TI GHM (Univariate) | 16.637 | 17.312 | 11.975 | 12.571 |
| | TI GHM (Bivariate) | **14.365** | 14.961 | 10.183 | **10.951** |
| | GHM (Univariate) | 22.012 | 24.057 | 15.697 | 18.596 |
| | GHM (Bivariate) | 18.029 | 23.000 | 11.352 | 17.168 |

Table 15: Mean Square Errors (MSE) for single wavelet and multiwavelet de-nosing

| Prefilter | Blocks | Bumps | Heavisine | Doppler |
|---|---|---|---|---|
| Identity | 73.107 | 35.581 | 56.742 | 55.752 |
| Xia | 161.85 | 410.12 | 68.966 | 413.47 |
| Minimal | 57.483 | 49.695 | 22.984 | 32.113 |
| Repeated Row | **23.869** | **20.428** | **11.387** | **13.586** |

Table 16: MSE for the TI multiwavelet bivariate thresholding with different prefilters

| SNR | Blocks | Bumps | Heavisine | Doppler |
|---|---|---|---|---|
| 3 | 25.232 | 22.849 | **8.680** | **12.935** |
| 5 | 25.623 | 21.147 | 10.266 | 13.264 |
| 7 | 23.869 | 20.427 | 11.387 | 13.586 |
| 9 | **22.006** | **20.127** | 11.989 | 13.737 |

Table 17: MSE for the TI multiwavelet bivariate thresholding with different signal-to-noise ratio (SNR)

| Wavelets | Blocks | Bumps | Heavisine | Doppler |
|---|---|---|---|---|
| D4 | 34.175 | 35.173 | 11.838 | 25.383 |
| Symmlet8 | 34.173 | 35.814 | 12.095 | 19.473 |
| Haar | 24.299 | 39.524 | 11.447 | 32.754 |
| Coiflet4 | 39.712 | 39.896 | 12.671 | 20.632 |
| GHM | **23.869** | **20.428** | **11.387** | **13.586** |

Table 18: MSE for different TI single wavelets thresholding and TI multiwavelet bivariate thresholding

# Chapter 5

# Conclusion and Future Work

In this thesis. we study the application of wavelet transforms in two important areas: pattern recognition and de-noising. In the area of pattern recognition, we introduce an invariant descriptor for recognising 2-D patterns which can be represented by periodic 1-D signals such as the contour of an object, the ring-projection, the line-moment, etc. The method first performs orthonormal shell decomposition on the periodic 1-D signal, then applies Fourier transform on each scale of the shell coefficients. The descriptor is invariant to the rotation of the pattern, and we can use a coarse-to-fine matching strategy in the classification process. The translation- and scale-invariance can be achieved while extracting the periodic 1-D signal from the 2-D pattern. we get 100% recognition rate for nearly all combinations of translation, rotation, and scaling. It is shown that the orthonormal shell-Fourier descriptor gets higher recognition rates than the Fourier descriptor for recognising printed Chinese characters. The orthonormal shell-Fourier descriptor can be applied to any 1-D periodic signal, say the contour of an numeral, aircraft, etc.

The multi-resolution Orthonormal Shell-Fourier Descriptor can be applied to any 1-D periodic signal, say, the contour of an object. In hand-printed numeral recognition, we can first trace the contour of a numeral and then represent the contour by two periodical coordinate signals $Contour\_x$ and $Contour\_y$. We can decompose the two signals into multi-resolution orthonormal shell by means of a specific wavelet filter. After that, we get the Fourier spectra $Feature\_x$ and $Feature\_y$ by applying Fourier transform on each shell scale. The query process can be performed on complex numbers $Feature\_x + i\ \ Feature\_y$ by using a coarse-to-fine matching strategy. It is also possible that we combine neural network with the Orthonormal Shell-Fourier

Descriptor.

The PFW and PWF algorithms proposed in Chapter 3 are computational reliable tools for pattern recognition. Both algorithms are invariant to translation, rotation, and scaling. We achieve very high recognition rate for all different rotation angles and scaling factors by using different wavelets. We only use a few low scale wavelet coefficients in the classification phase. It should be noted that although our experiments are done on a set of printed Chinese characters, our method is equally applicable to other pattern recognition problems such as air planes, key sets, or road signs. Future work can also be done for recognising more deformed and noisy patterns by incorporating neural network into the PFW and PWF algorithms.

In Chapter 4, we discuss and implement signal de-noising by using TI multiwavelets. Instead of applying univariate thresholding, we experiment with bivariate thresholding as pioneered by Downie and Silverman. Experimental results show that TI multiwavelet de-noising gives better results than the conventional TI single wavelet de-noising. In future work we will apply this scheme to image de-noising.

# Bibliography

[1] V. V. Anh, Q. Tieng, T. D. Bui and G. Chen, "The Hellinger-Kakutani metric for pattern recognition," in *Proceedings of IEEE ICIP'97.*

[2] G. Beylkin, "On the representation of operators in bases of compactly supported wavelets," *SIAM J. Numer. Anal.*, vol. 29, pp. 1716-1740, 1992.

[3] T. D. Bui and G. Chen,"Translation invariant de-noising using multiwavelets," *IEEE Transactions on Signal Processing*, Vol.46, No.12, pp.3414-3420, 1998.

[4] T. D. Bui and G. Chen, "Invariant Fourier-wavelet descriptor for pattern recognition," *Pattern Recognition*, 1999, to appear.

[5] T. D. Bui and G. Chen, "Multiresolution moment-wavelet-Fourier descriptor for 2D pattern recognition," in *Proceedings of SPIE'97*, Orlando, USA.

[6] Gene C.-H. Chang and C.-C. Jay Kuo, "Wavelet descriptor of planar curves: theory and applications." *IEEE Transactions on Image Processing* **5**, pp. 56-70, 1996.

[7] C. K. Chui, *An Introduction to Wavelets*, Boston: Academic Press, 1992.

[8] R. R. Koifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, pp. 713-718, 1993.

[9] T. H. Cormen, *Introduction to Algorithms.* New York : McGraw-Hill, 1990.

[10] I. Daubechies, *Ten Lectures on Wavelets*, Philadelpha: SIAM, 1992.

[11] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. on Pure and Applied Mathematics* **41**, pp. 909-996, 1988.

[12] T. R. Downie and B. W. Silverman, "The discrete multiple wavelet transform and thresholding methods," *IEEE Transactions on Signal Processing*, 1998, to appear.

[13] R. R. Coifman and D. L. Donoho, "Translation invariant denoising," In *Wavelets and Statistics," Springer Lecture Notes in Statistics 103*, pp. 125-150, New York:Springer-Verlag.

[14] D. L. Donoho, "Denoising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, pp. 613-627, 1995.

[15] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shringkage," *Biometrika*, vol. 81, pp. 425-455, 1994.

[16] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet shrinkage: Asymptopia?" *Journal of the Royal Statistics Society, Sieries B*, vol. 57, pp. 301-369, 1995.

[17] D. V. Ellis, *Well Logging for Earth Scientists.* New York : Elsevier.

[18] H. Freeman, "Computer processing of line drawing images," *Comput. Serv.*, 6, pp. 57-97, 1974.

[19] J. S. Geronimo, D. P. Hardin, and P. R. Massopust, "Fractal functions and wavelet expansions based on several scaling functions," *Journal of Approximation Theory*, vol. 78, pp. 373-401, 1994.

[20] A. E. Grace and M. Spann, "A comparison between Fourier-Mellin descriptors and moment based features for invariant recognition using neural networks," *Pattern Recognition Letters* **12**, pp. 635-643, 1991.

[21] G. H. Granlund, "Fourier processing for handwritten character recognition," *IEEE Trans. Comput. 21*, pp. 195-201, 1992.

[22] U. Grenander, *Abstract Inference*, Wiley, 1981.

[23] C. Herley, J. Kovacevic, K. Ramachandran, and M. Vetterli, "Tilings of the time-frequency plane : Construction of orthogonal bases and fast tiling algorithms," *IEEE Trans. Signal Processing*, vol. 41, pp. 3342-3359, 1993.

[24] P. J. Huber, *Robust Statistics*, Wiley & Sons, New York, USA, 1981.

[25] Y. Hui, C. W. Kok, and T. Q. Nguyen, "Shift-invariant filter banks: Theory, design, and applications," submitted to *IEEE Transaction on Signal Processing.*

[26] Y. Hui, C. W. Kok, and T. Q. Nguyen, "Image compression using shift-invariant dydiac wavelet transform," submitted to *IEEE Transaction on Signal Processing.*

[27] N. S. Jayant and Peter Noll, *Digital coding of waveforms : Principles and applications to speech and video.* Englewood Cliffs, NJ:Prentice Hall, 1984.

[28] A. Khotanzad and Y. H. Hang, "Invariant image recognition by Zernike moments," *IEEE Transactions on PARMI*, vol. 12, no.5, pp. 489-497, 1990.

[29] Seong-Whan Lee, Chang-Hun Kim, Hong Ma and Yuan Y. Tang, "Multiresolution recognition of unconstrained handwritten numerals with wavelet transform and multilayer cluster neural network," *Pattern Recognition*, vol. 29, 1996.

[30] Jin Li and C.-C. Jay Kuo, "Automatic target shape recognition via deformable wavelet templates," In *SPIE: Wavelet Application III*, Orlando, FL, Apr. 9 1996.

[31] C. C. Lin and R. Chellappa, "Classification of partial 2-D shapes using Fourier descriprtors, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 686-690, 1987.

[32] Jie Liang and Thomas W.Parks, "A translation-invariant wavelet representation algorithm with application," *IEEE Transactions on Signal Processing*, Vol. 44, No. 2, pp. 225-232, 1996.

[33] Jie Liang and Thomas W.Parks, "A two-dimensional translation invariant wavelet representation and its applications," in *Proceedings of IEEE ICIP*, vol. 1, pp. 66-69, 1994.

[34] S. Mallat, "Zero-crossing of a wavelet transform," *IEEE Transactions on Information Theory*, Vol.37, No.4, pp. 1019-1033, 1991.

[35] S. Mallat, "Multifrequency channel decomposition of images and wavelet models," *IEEE Trans. Acoust. Speech Signal Processing*, Vol.37, No.12, pp. 2091-2110, 1989.

[36] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.11, No.7, pp. 674-693, 1989.

[37] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L_2(R)$," *Trans. Amer. Math. Soc.* Vol.315, pp. 69-87, 1989.

[38] S. Mallat and S. Zhong, "Characterisation of signals from multiscale edges," *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Vol.14, No.7, pp. 710-732, 1992.

[39] S. D. Marco, P. N. Heller, and J. Weiss, "An M-band, 2-dimensional translation invariant wavelet transform and its applications," in *Proceedings of IEEE ICASSP*, pp. 1077-1080, 1995.

[40] Y. Meyer, *Ondelettes et Operateurs*. Hermann, Paris, 1990.

[41] Eric Persoon and King-Sun Fu, "Shape discrimination using Fourier descriptors," *IEEE Transactions on Systems, Man and Cybernetics* **SMC-7**, pp. 170-179, 1977.

[42] Naoki Saito and Gregory Beylkin, "Multiresolution representation using the auto-correlation functions of compactly supported wavelets," in *Proceedings of ICASSP*, vol. 4, pp. 381-384, 1992.

[43] M. Shrihar and A. Badreldin, "High accuracy character recognition algorithm using Fourier and topological descriptors," *Pattern Recognition* **17**, pp. 515-524, 1984.

[44] M. J. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Transactions on ASSP*, vol. 34, pp. 434-441, 1986.

[45] G. Strang and V. Strela, "Orthogonal multiwavelets with vanishing moments," *Optical Engineering*, vol. 33, pp. 2104-2107, 1994.

[46] G. Strang and V. Strela, "Short wavelets and matrix dilation equations," *IEEE Trans. on Signal Processing*, vol. 43, pp. 108-115, 1995.

[47] V. Strela, P. N. Heller, G. Strang, P. Topiwala, and C. Heil, "The application of multiwavelet filter banks to image processing," Technical report, MIT, USA, 1995.

[48] Yuan Y. Tang, Bing F. Li, Hong Ma, Jiming Liu, C.H.Leung and Ching Y. Suen, "A novel approach to optical character recognition based on ring-projection-wavelet-fractal signatures," *ICPR'96*, Vol II, pp. 325-329, 1996.

[49] O. D. Trier, A. K. Jain and T. Taxt, "Feature extraction methods for character recognition - A survey," *Pattern Recognition* **29**, pp. 641-662, 1996.

[50] Shuen-Shyang Wang, Po-Cheng Chen and Wen-Gou Lin, "Invariant pattern recognition by moment Fourier descriptor," *Pattern Recognition* **27**, pp. 1735-1742, 1994.

[51] Jeffrey Wood, "Invariant pattern recognition : A review," *Pattern Recognition*, vol. 29, pp. 1-17, 1996.

[52] Patrick Wunsch and Andrew F. Laine, "Wavelet descriptors for multiresolution recognition of handprinted characters," *Pattern Recognition* **28**, pp. 1237-1249, 1995.

[53] X. G. Xia, J. Geronimo, D. Hardin, and B. Suter, "On discrete multiwavelet transform," *IEEE Trans. on Signal Processing*, vol. 44, pp. 25-35, 1996.

[54] Charles T. Zahn and Ralph Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Transactions on Computers* **C-21**, pp. 269-281, 1972.

[55] Ming Zhang, Ching Y. Suen and Tien D. Bui, "Feature extraction in character recognition with associative memory classifier," *International Journal of Pattern Recognition and Artificial Intelligence* **10**, pp. 325-348, 1996.