WAVELETS FOR TEXTURE ANALYSIS

M. Sc Thesis

in

Electrical and Electronics Engineering
University of Gaziantep

By
Bayram KARA
September 2003

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Bülent GÖNÜL

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Tuncay EGE

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Nurdal WATSUJI

Supervisor

Examining Committee Members

Prof. Dr. Arif NACAROĞLU .....

Asst. Prof. Dr. Nurdal WATSUJİ .....

Asst. Prof. Dr. İlyas EKER .....

To my son Fırat,

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst. Prof. Dr. Nurdal Watsuji for her guidance and insight throughout the research. Thanks go to the other department members, Prof. Dr. Arif Nacaroğlu and Assoc. Prof. Dr. Gülay Tohumoğlu for their suggestions and comments. I would like to thank also assistances Nuran Doğru and Kemal Delihacıoğlu for their helps and friendships.

To my wife, Kezban, I offer sincere thanks for her unshakable faith in me and her willingness to endure with me the vicissitudes of my endeavors.

To my children, Fırat and Aslı, I thank them for understanding my frequent absences. I apologize especially to my son Fırat for leaving him alone for many years. He was only one year old when I started this work and I could not be interested in him any more. I would like to say that I love him very much and I hope he will understand me in the future.

ABSTRACT

WAVELETS FOR TEXTURE ANALYSIS

KARA, Bayram
M.Sc. in Electrical and Electronics Eng.
Supervisor: Asst. Prof. Dr. Nurdal WATSUJI
September 2003, 73 pages

In this thesis, mainly, the classification of textured images which has a great importance in texture analysis has been studied. Texture analysis plays an important role in many image processing tasks such as remote sensing, medical imaging, robot vision and query by content in large image databases. An efficient classification of textured images relies on feature extraction stage. In our analysis, wavelets which can be accepted as a new analysis tool have been used. Our work is focused on derivation of appropriate wavelet coefficient properties that will lead to a successful texture analysis. It was shown that wavelet which is a multiscale representation of an image, was an appropriate tool for texture feature extraction and the statistical properties of wavelet coefficients could be used as efficient texture signatures.

In this work, eight real world textured images from different natural scenes from the VisTex database were selected. Then each image was subdivided into 64 non-overlapping subimages and a database of 512 image regions of 8 texture classes was obtained. These images were analyzed with Biorthogonal Wavelets and classified with using statistical values of wavelet detail coefficients. The resulting error rates of classification experiments were compared and it was shown that, the standart, median and mean deviations of wavelet detail coefficients were appropriate features for a successful texture classification.

Key words: Wavelets, texture analysis, pattern recognition, image processing.

ÖZ

DOKU ANALİZİNDE WAVELETLER

KARA, Bayram
Yüksek Lisans Tezi, Elek. ve Elektronik Müh. Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. Nurdal WATSUJİ
Eylül 2003, 73 sayfa

Bu tezde esas olarak, doku analizinde büyük bir öneme sahip olan, dokularına ayrılmış resimlerin sınıflandırılması üzerinde çalışılmıştır. Doku analizi uzaktan algılama, tıbbi resimleme, robot görüntülemesi ve büyük resim veritabanlarında içerik sorgulama gibi birçok görüntü işleme görevlerinde önemli bir rol oynamaktadır. Dokularına ayrılmış görüntülerin etkili bir sınıflandırması, belirleyici özelliklerin seçilip çıkarılmasına bağlıdır. Analizlerimizde, yeni bir analiz metodu olarak kabul edilen waveletler kullanılmıştır. Çalışmamız, bizi başarılı bir doku analizine götürecek wavelet katsayısı özelliklerinin türetilmesi üzerinde odaklanmıştır. Bir görüntünün çoklu ölçekli gösterimi olan waveletin doku özelliği çıkarmada uygun bir araç olduğu, ve wavelet katsayılarının istatistiksel özelliklerinin etkili doku imzaları oldukları gösterilmiştir.

Bu çalışmada, VisTex veritabanından farklı doğal manzaralarda sekiz tane resim seçildi. Daha sonra, her bir resim üst üste gelmeyecek şekilde 64 tane alt resme bölündü ve 8 doku sınıfından toplam 512 resimden oluşan bir veritabanı elde edildi. Bu resimler Biorthogonal Wavelet ile analiz edildi ve wavelet detay katsayılarının istatistiksel değerleri kullanılarak sınıflandırmaya tabi tutuldu. Sınıflandırma deneylerinin sonucunda elde edilen hata oranları karşılaştırılarak, wavelet detay katsayılarının standart, medyan ve ortalama sapmalarının başarılı bir doku sınıflandırması için uygun özellikler oldukları gösterilmiştir.

Anahtar kelimeler: Waveletler, doku analizi, model tanıma, görüntü işleme.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

A	Amperes
C	Covariance matrix
CWT	Continuous Wavelet Transform
d	degree
DWT	Discrete Wavelet Transform
E	Energy
ECG	Electrocardiograph
EEG	Electroencephalograph
EMG	Electromyogram
f	frequency in hertz
FIR	Finite Impulse Response
FT	Fourier Transform
Hz	Hertz
I	Image
Knn	K nearest neighbor
LVQ	Learning Vector Quantization
MD	Mean Deviation
MRA	Multiresolution Analysis
ms	millisecond
QMF	Quadrature Mirror Filters
STFT	Short Time Fourier Transform
t	time
v	volts
Vs	Voltage source
WFT	Windowed Fourier Transform
WT	Wavelet Transform
$\langle f, \psi_{a,b} \rangle$	Inner product
$\downarrow$	downsampling
ω	frequency in radians per second
$\int$	Integration over the whole space
$\uparrow$	upsampling
*	convolution operator

CHAPTER 1

INTRODUCTION

One of the essential aims of the analysis of signals is the distinguishing them from each other, that is recognizing or classifying them and that is the main subject of this thesis. And this is the subject of the pattern recognition. Here, a pattern can be any object like an image, a face, a sound, a trend in curve etc. The most important step in pattern recognition is the feature extraction stage. If the extracted features contain the required information, then we can classify the patterns truly. Otherwise, we can not make a true classification. That is why we have to obtain the suitable features of patterns for a powerful classification.

Texture analysis plays an important role in many pattern recognition and image processing tasks such as remote sensing, medical imaging, robot vision and query by content in large image databases. Texture that can be defined as the set of local neighbourhood properties of the grey levels of an image region is an important cue for the analysis and segmentation of many types of images.

Texture analysis is usually dealt with in two major phases. The first phase is feature extraction in which the image information is reduced to a small set of descriptive features. The second step deals with classifying the features obtained from individual pixels (texture segmentation) or a collection of pixels (image classification) [1]. Various methods for texture analysis have been proposed during the last decades but this problem remains difficult and subject to intensive research. Three well known methods are local linear transforms [2,3], Gabor filtering [2,4] and the co-occurrence approach [2,5]. As an alternative to these methods, in this thesis, wavelets are used in the sense of feature extraction as Wouwer did [1].

Wavelets are not very old and can be accepted as a new signal analysis tool. We can define a wavelet as a waveform of effectively limited duration that has an average value of zero [6]. Wavelet Transform provides the time-frequency representation of a signal. The Discrete Wavelet Transform maps an image on a series of wavelet detail coefficients which constitute a multiscale representation of the image. In this work, it was shown that a texture can be characterized by the statistical properties of the horizontal, vertical and diagonal wavelet detail coefficients. The standart, median absolute and mean absolute deviations of the coefficients are used as wavelet signatures.

In 1998, G. Van de Wouwer has studied texture analysis and classification in his thesis. He used wavelets in the analysis and energy and cooccurrence signatures (second order statistical properties) for texture classification. He obtained quite low error rates in his work [1].

In 2001, A. Drimbarean has done some experiments in colour texture analysis and classification. In addition to grey levels of an image, he used RGB colour information of images for texture classification. In his experiments, he used Local linear transforms, Gabor filters and Co-occurrence approach and obtained acceptable low error rates [2].

The organization of this thesis is as follows:

Chapter 2 deals with the descriptions of the basic mathematical transformations. The Fourier and the Wavelet Transforms are explained basically in this chapter. In chapter 3, wavelet analysis is explained in more detail. The Discrete and Continuous Wavelet Transforms and multiresolution analysis are explained in this chapter. In chapter 4, wavelet applications and various wavelet families are introduced. In chapter 5, basic principles of texture analysis and pattern recognition are introduced. And the classification experiments we made are also presented in this chapter. And finally, chapter 6 contains the conclusions and discussions of this work.

CHAPTER 2

MATHEMATICAL TRANSFORMATIONS

2.1 Introduction

In our environment, there are many kinds of signals which have to be analyzed in a way. For example, there are seismic tremors, human speech, engine vibrations, medical images, financial data, music and many other types of signals. We try to analyze all these signals because we want to learn more about them. We want to find out their behaviours, characteristics, contents, shapes etc. Shortly, we want to know them with all their properties. They are in all parts of our life. We use them for human life being easier and more comfortable. For this reason, we have to know more features about them as possible.

Sometimes it is possible to get enough information about a signal in its original form. In these cases, there is no need to apply further mathematical transformations. For example, if you only want to learn the amplitudes of a signal at different time points, simply you plot the amplitude versus time characteristic of the signal and you can learn how much the amplitude is at which time point.

But in many cases, we can not derive enough information of a signal from its time-amplitude representation. Because, the most distinguished information is hidden in the frequency content of the signal. The frequency spectrum of a signal is basically the frequency components (spectral components) of that signal. The frequency spectrum of a signal shows what frequencies exist in the signal. In these cases we need to apply some mathematical transformations to signal. Basically, mathematical transformations are applied to a signal to obtain further information from that signal

that is not readily available in its original form. Here we assume a time-domain signal as a **raw signal** (original signal), and a signal that has been transformed by any of the available mathematical transformations as a **processed signal.** There are number of transformations that can be applied, among which the Fourier transforms are probably by far the most popular [7,8].

2.2 Fourier Transform

In practice, most of the signals are time-domain signals in their original format. That is, whatever that signal is measuring, is a function of time. But this representation is not always the best representation of the signal for most signal processing related applications. In these cases, we need to know the frequency content of the signal.

We can define the frequency as the change in rate of some quantity. If this quantity (a mathematical or physical variable) changes rapidly, we say that it is of high frequency and if it changes slowly, we say that it is of low frequency. If this variable does not change at all, then we say it has zero frequency, or no frequency. The frequency is measured in cycles/second, or with a more common name, in 'Hertz'. For example let's take two alternating electric current equations with 5 and 50 hertz frequency and plot them:

$$I_1 = \sin(2 * pi * 5 * t) \quad (A) \tag{2.1}$$

$$I_2 = \sin(2 * pi * 50 * t) \quad (A) \tag{2.2}$$

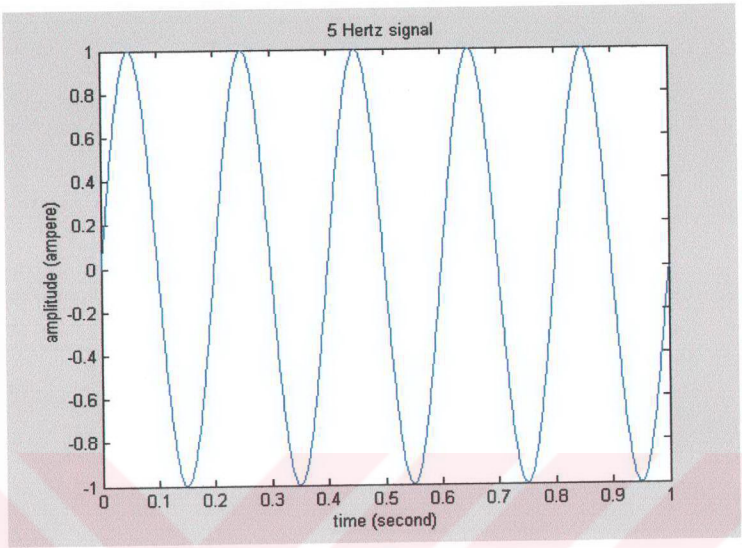


Figure 2.1: 5 Hertz signal

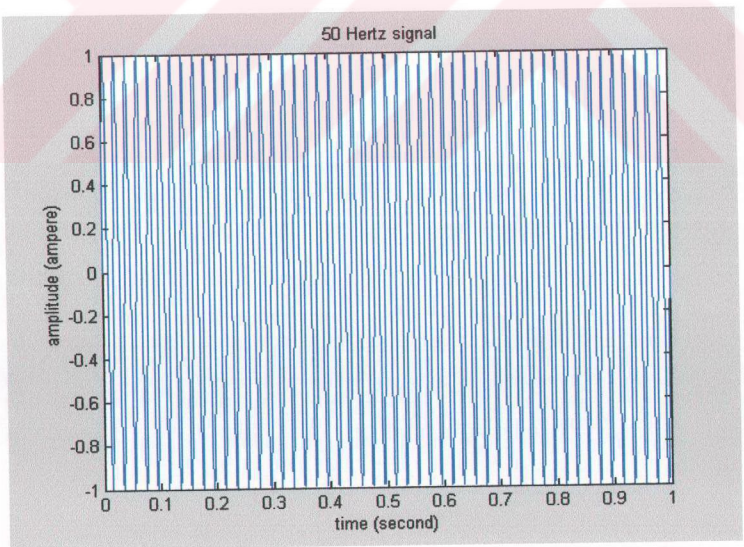


Figure 2.2: 50 Hertz signal

In order to find the frequency content of a signal, we use Fourier Transform. When the Fourier Transform of a signal in time domain is taken, the frequency-amplitude representation of that signal is obtained. This plot tells us how much of each frequency exists in the signal. The Fourier Transform of a signal x(t) is taken as follows:

$$X(f) = \int_{-\infty}^{\infty} x(t).e^{-2j\pi ft}dt \tag{2.3}$$

Here x(t) denotes any signal in time domain and X(f) denotes the Fourier Transform of the signal in frequency domain, t stands for time and f stands for frequency. The following equation shows that the inverse Fourier Transform:

$$x(t) = \int_{-\infty}^{\infty} X(f).e^{2j\pi ft}df \tag{2.4}$$

Fourier Transform decomposes a signal to complex exponential functions of different frequencies. Now let's take the Fourier Transform of 50 Hertz signal in Figure 2.2 and plot it in Figure 2.3:

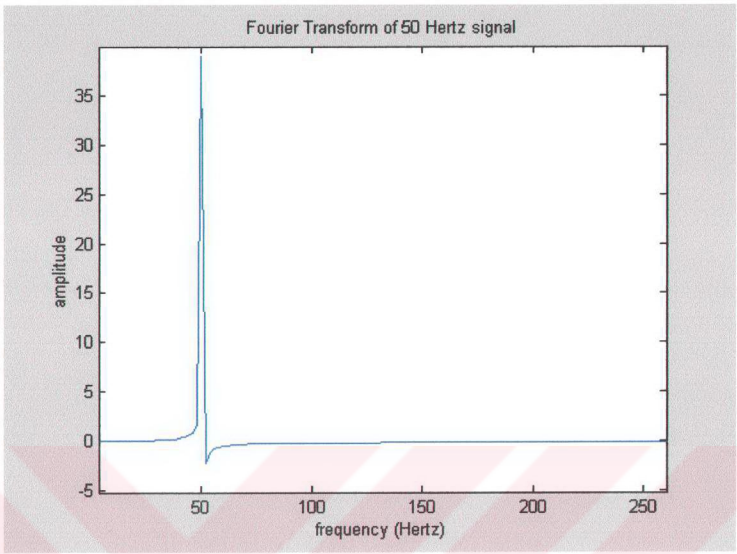


Figure 2.3: Fourier Transform of 50 hertz signal

In Figure 2.3, there is only one spike at 50 Hertz, and nothing elsewhere, since this signal has only 50 Hertz frequency component. We can derive two conclusions from Figures 2.2 and 2.3. The first one is that, there is time information but no frequency information in a time-domain signal. And the second is that, there is frequency information but no time information in Fourier transformed (frequency-domain) signal.

Fourier Transform tells us how much of each frequency exists in the signal, but it does not tell us when in time these frequency components exist. This information is not required when the signal is so-called stationary. As a definition, signals whose frequency content does not change in time are called **stationary signals** [7]. For example let us take a signal such as:

$$f(t) = \cos(2\pi 50t) + \cos(2\pi 100t) + \cos(2\pi 200t) + \cos(2\pi 400t) \qquad (2.5)$$

This signal has four different frequency components as 50, 100, 200 and 400 Hertz at any given time instant therefore it is a stationary signal. This signal is plotted in time-domain in the Figure 2.4, and its Fourier Transform is plotted in Figure 2.5.

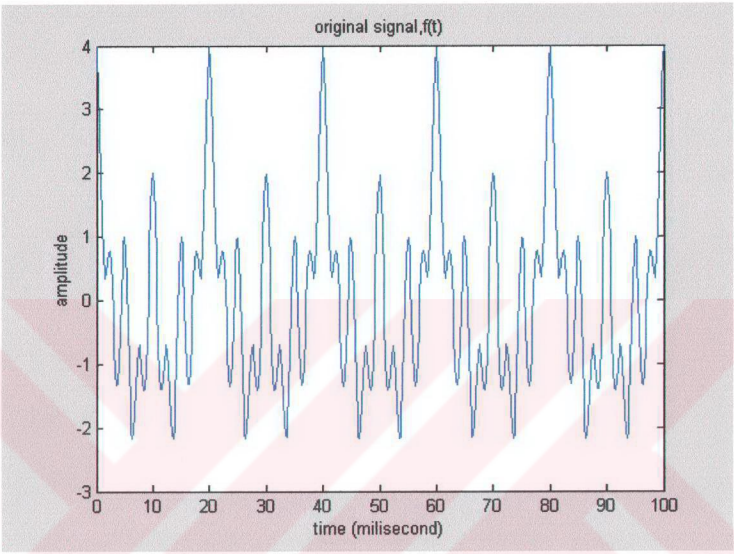


Figure 2.4: Amplitude-time representation of signal f(t) in (2.5)

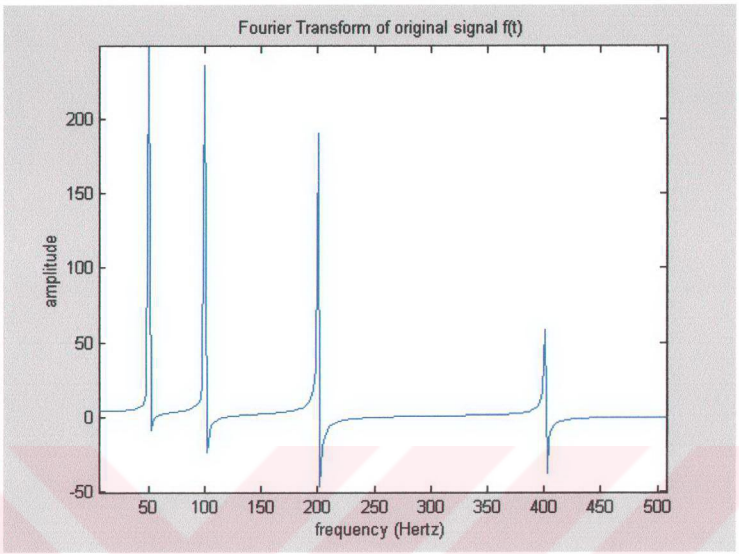


Figure 2.5: Fourier Transform of original signal f(t) in (2.5)

In Figure 2.5, we can see the four 50, 100, 200 and 400 Hertz frequency components clearly.

But in some signals, the frequency content changes in time. These signals are called **non-stationary signals** [7]. In these signals, there exist different frequency components at different time intervals. For example let's take a signal g(t) such as:

$$g(t) = \begin{cases} \sin(2\pi 50t); 0 \le t \le 250\text{ms}, \\ \sin(2\pi 100t); 251 \le t \le 500\text{ms}, \\ \sin(2\pi 200t); 501 \le t \le 750\text{ms}, \\ \sin(2\pi 400t); 751 \le t \le 1000\text{ms}. \end{cases} \tag{2.6}$$

This signal is a non-stationary signal. Because it has four different frequency components at four different time intervals. We plotted this signal in the following figure:

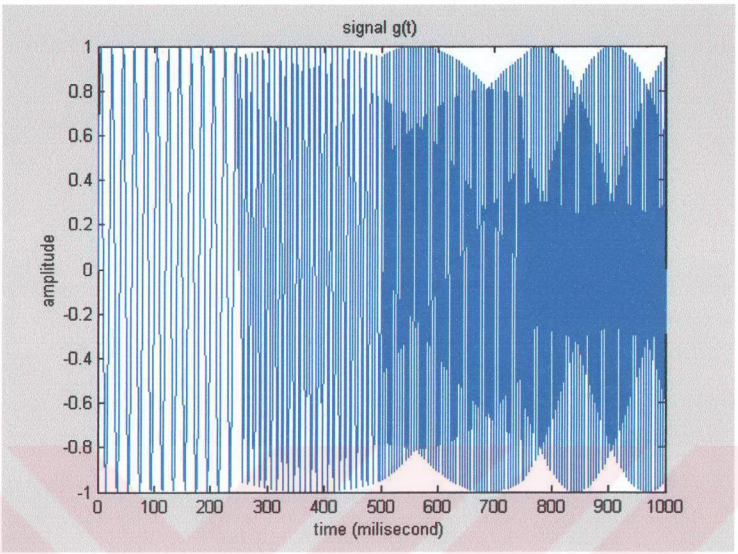


Figure 2.6: The plot of non-stationary signal g(t) in (2.6)

And now, if we take the Fourier Transform of signal g(t) in (2.6) and plot it, we obtain the Figure 2.7.

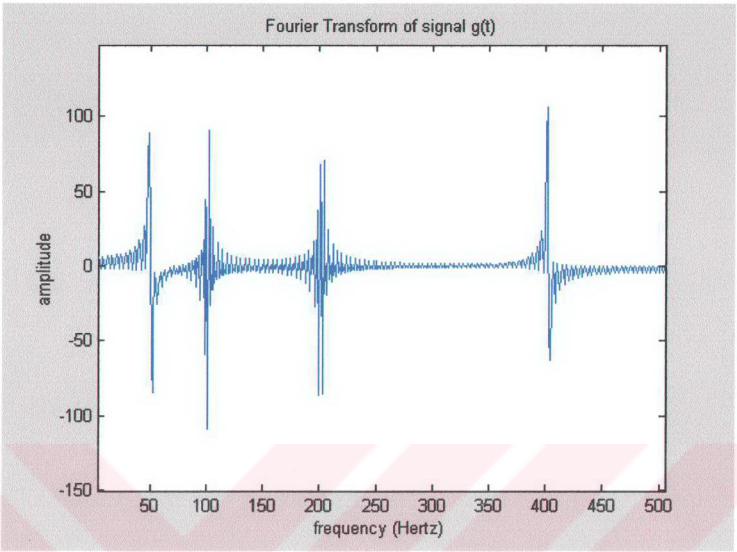


Figure 2.7: Fourier Transform of non-stationary signal g(t) in (2.6)

Now let's compare Figure 2.5 and Figure 2.7. The frequency spectrums in these figures are almost identical except little differences in amplitudes. We see the same 50,100,200 and 400 Hz frequency components in both them. But indeed, the graphics in these figures are frequency spectrums of exactly different time-domain signals such as a stationary signal f(t) in (2.5) and a non-stationary signal g(t) in (2.6) respectively. But Fourier Transforms of these different time-domain signals are almost the same. It means, Fourier Transform can not separate these two signals. Therefore we can conclude that Fourier Transform is not a suitable technique for non-stationary signals.

There are a lot of practical stationary signals, as well as non-stationary ones. Almost all biological signals are non-stationary. Some of the most famous ones are ECG (electrical activity of the heart, electrocardiograph), EEG (electrical activity of the brain, electroencephalograph) and EMG (electrical activity of the muscles, electromyogram) [7].

When the time localization of the spectral components is needed, a transform giving the time-frequency representation of the signal is needed. There are some transforms which are suitable for these purposes, such as Short Time Fourier Transform, Wigner distributions, etc. The Wavelet Transform is a transform of this type and we think that it is the ultimate solution.

2.3 Wavelet Transform

Wavelet Transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal, and it was developed as an alternative to the STFT (Short Time Fourier Transform). Often times, some information can not be obtained in the time-domain but it can be seen in the frequency domain. A particular spectral component occurring at any instant can be of particular interest. For example, in EEGs, the latency of an event-related potential is of particular interest (event-related potential is the response of the brain to a specific stimulus like flash-light, the latency of this response is the amount of time elapsed between the onset of the stimulus and the response) [7].

With WT (Wavelet Transform), we pass the time-domain signal from various high pass and low pass filters. Every time this procedure is repeated, some portion of the signal corresponding to some frequencies being removed from the signal. Here, let's try to explain shortly how Wavelet Transform works. Suppose we have a signal which has frequencies up to 1000 Hz. In the first stage we split up the signal in to two parts by passing the signal from a high pass and a low pass filter which results in two different versions of the same signal: portion of the signal corresponding to 0-500 Hz and 500-1000 Hz. Then, we take either portion (usually low pass portion) or both, and do the same thing again. This operation is called **decomposition**. Assuming that we have taken the low pass portion, we now have 3 sets of data, each corresponding to the same signal at frequencies 0-250 Hz, 250-500 Hz, 500-1000 Hz. Then we take the low pass portion again and pass it through low and high pass filters; we now have 4 sets of signals corresponding to 0-125 Hz, 125-250 Hz, 250-500 Hz, and 500-1000 Hz. We continue like this until we have decomposed the signal to a pre-defined certain level. Then we have a bunch of signals, which actually

represent the same signal, but all corresponding to different frequency bands. We know which signal corresponds to which frequency band, and if we put all of them together and plot them on a three-dimensional graph, we will have time in one axis, frequency in the second and amplitude in the third axis. This will show us which frequencies exist at which time (there is an issue, called "**uncertainty principle**", which states that, we cannot exactly know what frequency exists at what time instance , but we can only know what frequency bands exist at what time intervals).

The uncertainty principle, originally found and formulated by Heisenberg, states that, the momentum and the position of a moving particle cannot be known simultaneously [7]. This applies to our subject as follows: The frequency and time information of a signal at some certain point in the time-frequency plane, cannot be known. In other words: We cannot know what spectral component exists at any given time instant. The best we can do is to investigate what spectral components exist at any given interval of time. This is a problem of resolution, and it is the main reason why researchers have switched to WT from STFT. STFT gives a fixed resolution at all times, whereas WT gives a variable resolution as follows:

Higher frequencies are better resolved in time, and lower frequencies are better resolved in frequency. This means that, a certain high frequency component can be located better in time (with less relative error) than a low frequency component. On the contrary, a low frequency component can be located better in frequency compared to high frequency component.

In the following figures there are, examples of wavelet transforms of some signals. The detailed explanation about Wavelet Transform is given in chapters 3 and 4.

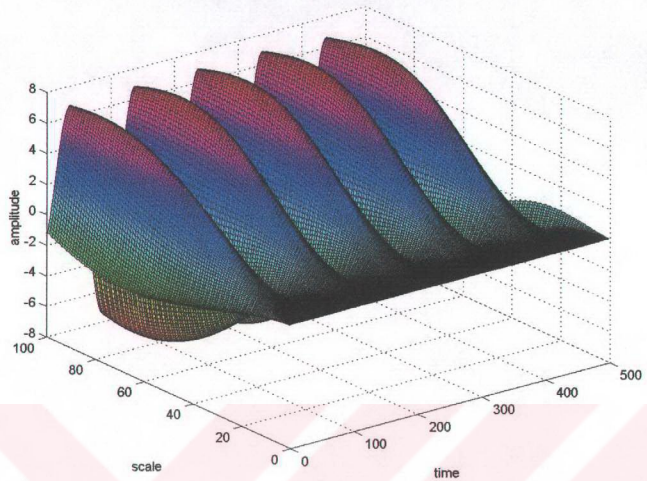


Figure 2.8: Wavelet transform of 50 Hz signal I_2 in (2.2)

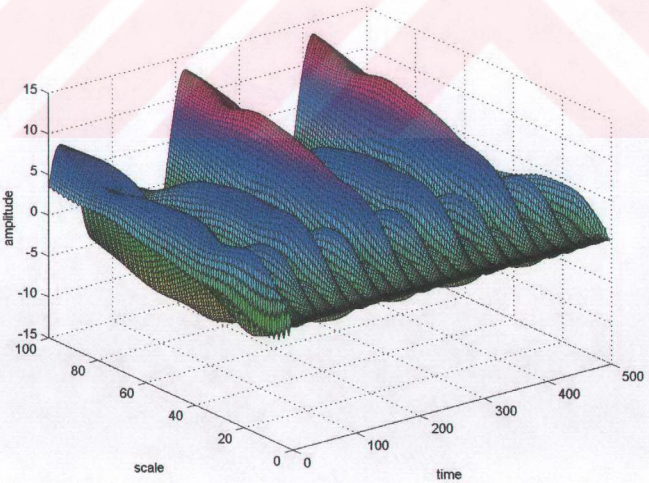


Figure 2.9: Wavelet transform of stationary signal f(t) in (2.5)

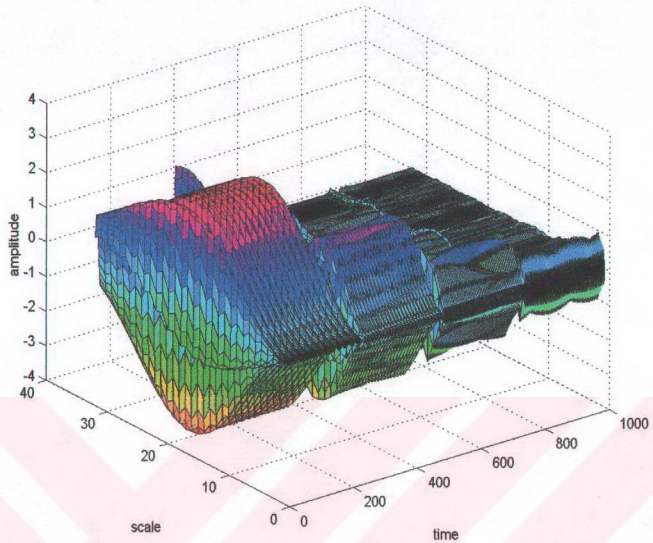


Figure 2.10: Wavelet transform of non-stationary signal g(t) in (2.6)

CHAPTER 3

WAVELET ANALYSIS

3.1 Introduction

The name 'wavelets' was introduced in the early eighties by Morlet, a french geophysicists. The seismic data he was studying typically exhibited rapidly changing frequency contents for which Fourier analysis did not success as an analyzing tool. This is why he investigated the existence of functions which had sufficient compact support in both time as frequency domain, and called them wavelets. Grossman has provided the mathematical basis of Morlet's ideas which has triggered the start of the construction of a complete mathematical framework, nowadays known as wavelet theory, and a still rapidly growing number of applications of this frame-work [1,9].

Wavelet theory is not really a whole new theory. Rather it is the result of a cross fertilization and generalizes concepts known in several fields like geophysics, signal (and image) analysis and compression, physics and mathematics (statistics). All the researches in wavelets resulted in a comprehensive mathematical framework for signal analysis with applications in physics, speech processing, image coding, image recognition and segmentation, denoising, density estimation, etc.

3.2 Continuous Wavelet Transform

The continuous wavelet transform (CWT) was developed as an alternative approach to the short time Fourier transform (STFT), to overcome the resolution problem. The wavelet analysis is done in a similar way to the STFT analysis, in the sense that the

signal is multiplied with a function, (the wavelet), similar to the window function in the STFT, and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and the CWT:

1. The Fourier transforms of the windowed signals are not taken, and therefore single peak will be seen corresponding to a sinusoid, i.e., negative frequencies are not computed.

2. The width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform [7].

An important issue in the analysis of objects (whether they are images, sound signals, operators, etc) is their representation. One desires to represent the object in such a way that the required information is easily accessible. For instance, to build a classifier which recognizes several classes of images, one desires a representation that allows good feature extraction in the sense of obtaining features which minimize the overlap between class conditional densities.

One of the classical tools to achieve such a representation is Fourier analysis which maps a signal f(t) from the time domain to the frequency domain f (ω). The Fourier transform assumes that the signal to be analyzed is of infinite duration or is at least periodical [7,8]. This condition is often not met in real-world signals. Consider for example a musical sound wave, which consists of several notes with a particular frequency which are played during a limited time (see Figure 3.1).

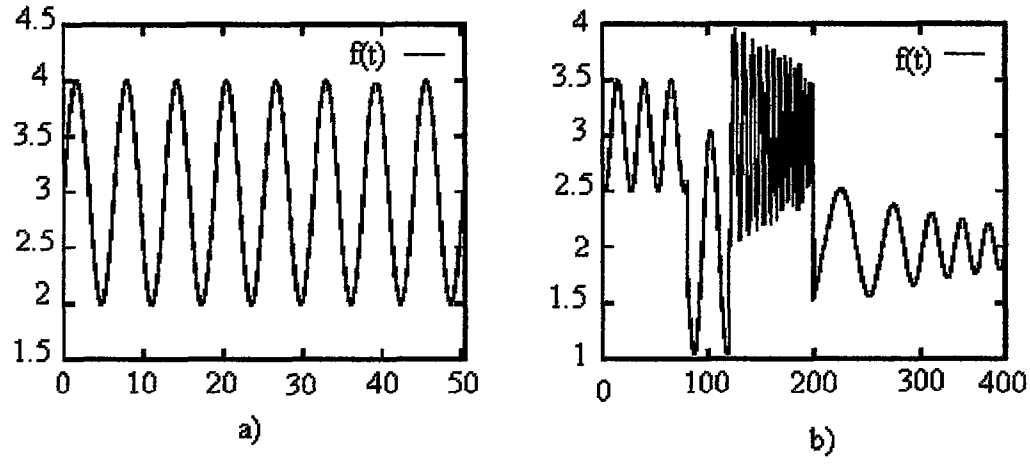


Figure 3.1: Example of two signals with different time-frequency characteristics

In Figure 3.1, a) the signal is clearly periodic and the Fourier transform will be able to analyze its frequency contents. In Figure 3.1, b) Fourier analysis is not appropriate since the signal exhibits different frequency behavior in different time windows (one says that particular frequencies live during a particular window in time). The last signal requires a time-frequency representation to be properly analyzed.

An obvious way to obtain such a representation is to place a localized window on the signal which is shifted along the time axis and to perform a Fourier transform at each point in time. This yields the Windowed Fourier Transform (WFT):

$$Sf(u,T) = \int f(t)W(t,T)e^{2\pi iut}dt \qquad (3.1)$$

Figure 3.2, a) schematically depicts where those functions live in the plane (the tiling of the time-frequency plane). Note that it is not possible to obtain simultaneously arbitrary fine localization in time and frequency due to the uncertainty principle. This principle states that the blocks tiling the time-frequency plane all have the same area and thus, once the width of the window W(t,T) is chosen, this tiling is fixed. However, it would be desirable to study the slowly varying properties of the signal (low frequencies) over a longer time span and vice versa for the high frequencies. Such a tiling is represented in Figure 3.2, b) and is provided by employing wavelets as analyzing functions.

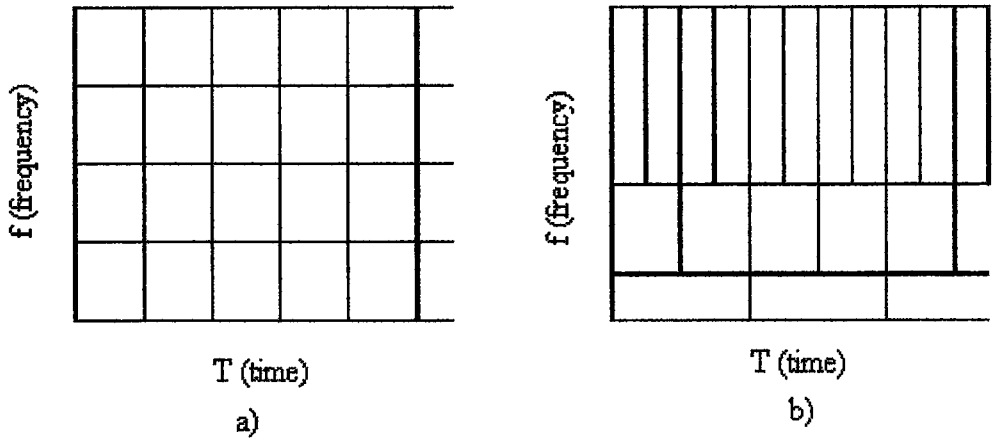


Figure 3.2: Tiling of the time frequency plane: a) as obtained by the WFT, b) as obtained by the wavelet transform

To obtain such a representation, one employs a wavelet family as analyzing functions. This is a family of functions derived from one single function (the mother wavelet) which is indexed by two parameters **a** and **b**:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{3.2}$$

The parameter **a** gives the dilation (which is inversely proportional with frequency) and **b** gives the displacement (time localization) [1,7,8]. A classical example of such a family is the Mexican hat given by

$$\psi(t) = (1-t^2).e^{-t^2/2} \tag{3.3}$$

which is the second derivative of a Gaussian [6].

The one dimensional continuous wavelet transform (CWT) is defined by:

$$Wf_a(b) = \int f(t)\psi^*_{a,b}(t)dt = \langle f, \psi_{a,b}\rangle, a \neq 0 \tag{3.4}$$

And the inverse wavelet transform is given by:

$$f(t) = C_{\psi}^{-1} \iint \langle f, \psi_{a,b} \rangle \psi_{a,b}(t) \frac{da.db}{a^2} \tag{3.5}$$

Where we require that the wavelet satisfies (admissibility condition):

$$C_{\psi} = 2\pi \int \frac{\left| \hat{\psi}(u) \right|^2}{|u|} du < \infty \tag{3.6}$$

If $\hat{\psi}$ is continuous, condition (3.6) is reduced to:

$$\hat{\psi}(0) = 0 \Leftrightarrow \int \psi(t) dt = 0 \tag{3.7}$$

This shows that a wavelet is a zero mean function and hence exhibits some oscillatory behavior. It can be proven that the admissibility condition is necessary to ensure that the wavelet transform is invertible (i.e. that (3.5) makes sense) [1,7-11].

Note that the CWT is essentially redundant which means that there is not a one-to-one mapping between the function space and the space of all wavelet transforms. However, this redundancy can be reduced or eliminated by constraining the choice of wavelet ψ and by discretizing the parameters **a** and **b**. This leads to the Discrete Wavelet Transform (DWT). Due to this discretization, the inverse discrete transform is not directly available and the questions of how f can be recovered from its wavelet transform and whether there exists numerically stable algorithms to perform the transform.

The different DWT schemes can be subdivided into two types:

1. Redundant. If a digitized signal consists of N samples, then a redundant DWT maps it on M > N samples. The theory of frames provides an answer to the questions mentioned above.

2. Non-redundant. In this case, use is made of wavelets which form an orthogonal basis. It is a particular interesting case since a fast algorithm has been constructed to perform the DWT [1,7-11].

3.3 Multiresolution Analysis

Although the time and frequency resolution problems are results of a physical phenomenon (the Heisenberg uncertainty principle) and exist regardless of the transform used, it is possible to analyze any signal by using an alternative approach called the **multiresolution analysis (MRA)**. MRA, as implied by its name, analyzes the signal at different frequencies with different resolutions. Every spectral component is not resolved equally as was the case in the STFT.

MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. This approach makes sense especially when the signal at hand has high frequency components for short durations and low frequency components for long durations. Fortunately, the signals that are encountered in practical applications are often of this type.

By now it is clear that a wavelet decomposition is a multiscale representation. The signal is studied at several scales for different values of **a**. For a larger **a,** the wavelet becomes broader in the time domain and thus smaller in the frequency domain and hence it will capture more low frequency characteristics of the signal. If one defines resolution as the number of basis functions per unit length used to represent the signal, one can view the wavelet representation as a multiresolution representation (where lower resolution means lower frequency and larger **a**).

An important step in wavelet theory is the connection of the decomposition with sub band filtering schemes, which were frequently used for signal compression. It turns out that the wavelet coefficients can be computed by iterative filtering of the signal. To this end, a set of quadrature mirror filters are employed. They consist of a lowpass filter h and a highpass filter g (strictly speaking, g is a bandpass filter, but

since it captures all the remaining high frequencies of the signal, it is commonly called a highpass filter). Further they employ the mirror filters $\tilde{h}$ and $\tilde{g}$ defined by $\tilde{h}(n) = h(-n)$ and $\tilde{g}(n) = g(-n)$. The decomposition and reconstruction schemes are illustrated in Figure 3.3.

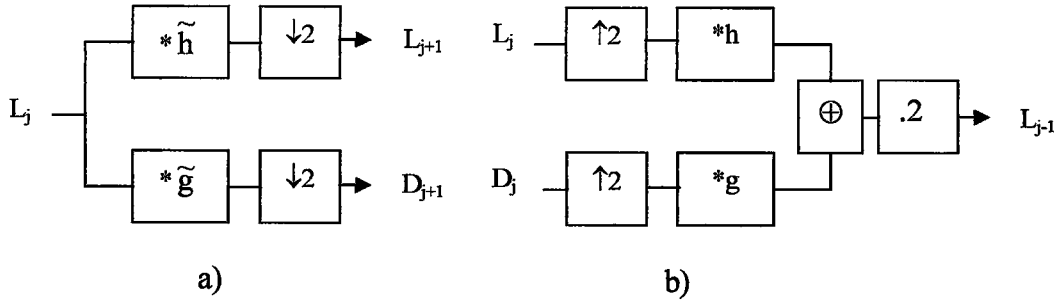


Figure 3.3: Decomposition (a) and reconstruction (b) schemes for computation of the wavelet coefficients using quadrature mirror filters, * denotes convolution and ↓2 (↑2) down sampling (up sampling) by a factor of 2

This scheme takes the original samples, filters and sub samples them, which produces a low resolution and detail signal on a lower scale. Mathematically speaking this sub sampling stems from the factor 2^{-j} in the basis functions $\phi_{j,m}$, but it is intuitively clear that when a signal is transformed to a lower resolution scale, less samples are required to represent it. Note that, due to this sub sampling, there are exactly as many wavelet coefficients as there are samples in the original signal f(n). This representation thus indeed is non-redundant, which at the first place was the point of introducing orthogonal wavelets [1].

3.4 Discrete Wavelet Transform

Although the discretized continuous wavelet transform enables the computation of the continuous wavelet transform by computers, it is not a true discrete transform. As a matter of fact, the wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal

is concerned. This redundancy, on the other hand, requires a significant amount of computation time and resources. The discrete wavelet transform (DWT), on the other hand, provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time. The DWT is considerably easier to implement when compared to the CWT.

The foundations of the DWT go back to 1976 when Croiser, Esteban, and Galand devised a technique to decompose discrete time signals. Crochiere, Weber, and Flanagan did a similar work on coding of speech signals in the same year. They named their analysis scheme as subband coding. In 1983, Burt defined a technique very similar to subband coding and named it pyramidal coding which is also known as multiresolution analysis. Later in 1989, Vetterli and Le Gall made some improvements to the subband coding scheme, removing the existing redundancy in the pyramidal coding scheme [7]. Subband coding is explained below.

The main idea is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. Recall that the CWT is a correlation between a wavelet at different scales and the signal with the scale (or the frequency) being used as a measure of similarity. The continuous wavelet transform was computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor n reduces the number of samples in the signal n times.

Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal. For example, upsampling by two refers to adding a new sample, usually a zero or an interpolated value, between every two samples of the signal. Upsampling a signal by a factor of n increases the number of samples in the signal by a factor of n [7,8].

Since the signal is a discrete time function, the terms function and sequence will be used interchangeably in the following discussion. This sequence will be denoted by x[n], where n is an integer.

The procedure starts with passing this signal (sequence) through a half band digital lowpass filter with impulse response h[n]. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k].h[n-k] \qquad (3.8)$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz. The unit of frequency is of particular importance at this time. In discrete signals, frequency is expressed in terms of radians. Accordingly, the sampling frequency of the signal is equal to 2p radians in terms of radial frequency. Therefore, the highest frequency component that exists in a signal will be p radians, if the signal is sampled at Nyquist's rate (which is twice the maximum frequency that exists in the signal); that is, the Nyquist's rate corresponds to p rad/s in the discrete frequency domain. Therefore using Hz is not appropriate for discrete signals. However, Hz is used whenever it is needed to clarify a discussion, since it is very common to think of frequency in terms of Hz. It should always be remembered that the unit of frequency for discrete time signals is radians.

After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest

frequency of p/2 radians instead of p radians. Simply discarding every other sample will subsample the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled. Note that the lowpass filtering removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies, which can be interpreted as losing half of the information. Therefore, the resolution is halved after the filtering operation. Note, however, the subsampling operation after filtering does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information. In summary, the lowpass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples are redundant. This doubles the scale [7,8].

This procedure can mathematically be expressed as:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k].x[2n-k] \tag{3.9}$$

Having said that, we now look how the DWT is actually computed:

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with lowpass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal x[n] is first passed through a halfband highpass filter g[n] and a lowpass filter h[n]. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p /2 radians instead of p . The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_{n} x[n].g[2k - n] \tag{3.10}$$

$$y_{low}[k] = \sum_{n} x[n].h[2k - n] \tag{3.11}$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure, which is also known as the subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence double the frequency resolution). Figure 3.4 illustrates this procedure, where x[n] is the original signal to be decomposed, and h[n] and g[n] are lowpass and highpass filters, respectively. The bandwidth of the signal at every level is marked on the figure as "f".

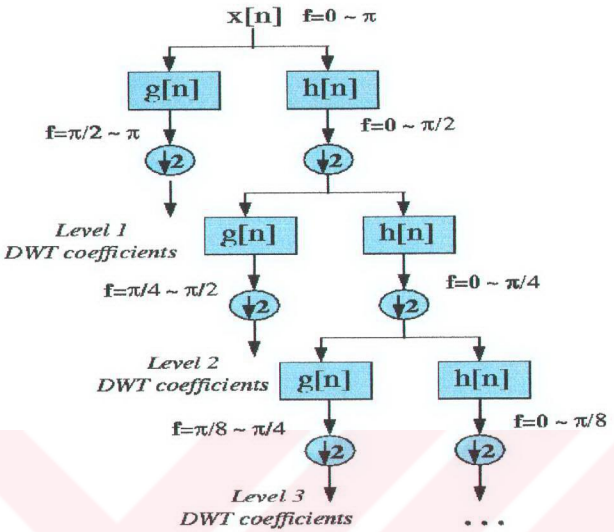


Figure 3.4: The subband coding algorithm

As an example, suppose that the original signal x[n] has 512 sample points, spanning a frequency band of zero to p rad/s. At the first decomposition level, the signal is passed through the highpass and lowpass filters, followed by subsampling by 2. The output of the highpass filter has 256 points (hence half the time resolution), but it only spans the frequencies p/2 to p rad/s (hence double the frequency resolution). These 256 samples constitute the first level of DWT coefficients. The output of the lowpass filter also has 256 samples, but it spans the other half of the frequency band, frequencies from 0 to p/2 rad/s. This signal is then passed through the same lowpass and highpass filters for further decomposition. The output of the second lowpass filter followed by subsampling has 128 samples spanning a frequency band of 0 to p/4 rad/s, and the output of the second highpass filter followed by subsampling has 128 samples spanning a frequency band of p/4 to p/2 rad/s. The second highpass filtered signal constitutes the second level of DWT coefficients. This signal has half the time resolution, but twice the frequency resolution of the first level signal. In other words, time resolution has decreased by a factor of 4, and frequency resolution

has increased by a factor of 4 compared to the original signal. The lowpass filter output is then filtered once again for further decomposition. This process continues until two samples are left. For this specific example there would be 8 levels of decomposition, each having half the number of samples of the previous level. The DWT of the original signal is then obtained by concatenating all coefficients starting from the last level of decomposition (remaining two samples, in this case). The DWT will then have the same number of coefficients as the original signal.

The frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies. The difference of this transform from the Fourier transform is that the time localization of these frequencies will not be lost. However, the time localization will have a resolution that depends on which level they appear. If the main information of the signal lies in the high frequencies, as happens most often, the time localization of these frequencies will be more precise, since they are characterized by more number of samples. If the main information lies only at very low frequencies, the time localization will not be very precise, since few samples are used to express signal at these frequencies. This procedure in effect offers a good time resolution at high frequencies, and good frequency resolution at low frequencies. Most practical signals encountered are of this type.

The frequency bands that are not very prominent in the original signal will have very low amplitudes, and that part of the DWT signal can be discarded without any major loss of information, allowing data reduction. Figure 3.5 illustrates an example of how DWT signals look like and how data reduction is provided. Figure 3.5, a) shows a typical 512-sample signal that is normalized to unit amplitude. The horizontal axis is the number of samples, whereas the vertical axis is the normalized amplitude. Figure 3.5, b) shows the 8 level DWT of the signal in Figure 3.5, a). The last 256 samples in this signal correspond to the highest frequency band, the previous 128 samples correspond to the second highest frequency band and so on. It should be noted that only the first 64 samples, which correspond to lower frequencies of the analysis, carry relevant information and the rest of this signal has virtually no information. Therefore, all but the first 64 samples can be discarded without any loss of information. This is how DWT provides a very effective data reduction scheme.

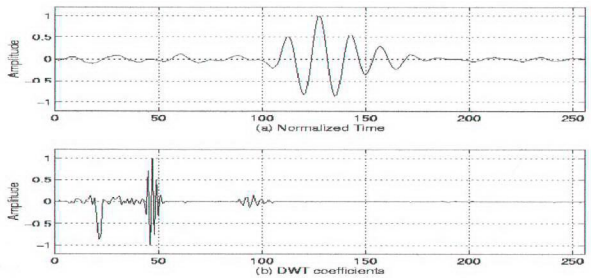


Figure 3.5: Example of a Discrete Wavelet Transform

One important property of the discrete wavelet transform is the relationship between the impulse responses of the highpass and lowpass filters. The highpass and lowpass filters are not independent of each other, and they are related by

$$g[L-1-n] = (-1)^n . h[n] \tag{3.12}$$

where g[n] is the highpass, h[n] is the lowpass filter, and L is the filter length (in number of points). Note that the two filters are odd index alternated reversed versions of each other. Lowpass to highpass conversion is provided by the $(-1)^n$ term. Filters satisfying this condition are commonly used in signal processing, and they are known as the Quadrature Mirror Filters (QMF) [7].

The two filtering and subsampling operations can be expressed by

$$y_{high}[k] = \sum_n x[n].g[-n+2k] \tag{3.13}$$

$$y_{low}[k] = \sum_{n} x[n].h[-n+2k] \tag{3.14}$$

The reconstruction in this case is very easy since halfband filters form orthonormal bases. The above procedure is followed in reverse order for the reconstruction. The signals at every level are upsampled by two, passed through the synthesis filters g'[n], and h'[n] (highpass and lowpass, respectively), and then added. The interesting point here is that the analysis and synthesis filters are identical to each other, except for a time reversal. Therefore, the reconstruction formula becomes (for each layer) as follows:

$$x[n] = \sum_{k=-\infty}^{\infty} \left((y_{high}[k].g[-n+2k]) + (y_{low}[k].h[-n+2k]) \right) \tag{3.15}$$

However, if the filters are not ideal halfband, then perfect reconstruction cannot be achieved. Although it is not possible to realize ideal filters, under certain conditions it is possible to find filters that provide perfect reconstruction. The most famous ones are the ones developed by Ingrid Daubechies, and they are known as Daubechies wavelets [10].

Note that due to successive subsampling by 2, the signal length must be a power of 2, or at least a multiple of power of 2, in order this scheme to be efficient. The length of the signal determines the number of levels that the signal can be decomposed to. For example, if the signal length is 1024, ten levels of decomposition are possible.

Interpreting the DWT coefficients can sometimes be rather difficult because the way DWT coefficients are presented is rather peculiar. To make a real long story real short, DWT coefficients of each level are concatenated, starting with the last level.

Suppose we have a 256-sample long signal sampled at 10 MHZ and we wish to obtain its DWT coefficients. Since the signal is sampled at 10 MHz, the highest frequency component that exists in the signal is 5 MHz. At the first level, the signal is passed through the lowpass filter h[n], and the highpass filter g[n], the outputs of which are subsampled by two. The highpass filter output is the first level DWT

coefficients. There are 128 of them, and they represent the signal in the [2.5 5] MHz range. These 128 samples are the last 128 samples plotted. The lowpass filter output, which also has 128 samples, but spanning the frequency band of [0 2.5] MHz, are further decomposed by passing them through the same h[n] and g[n]. The output of the second highpass filter is the level 2 DWT coefficients and these 64 samples precede the 128 level 1 coefficients in the plot. The output of the second lowpass filter is further decomposed, once again by passing it through the filters h[n] and g[n]. The output of the third highpass filter is the level 3 DWT coefficients. These 32 samples precede the level 2 DWT coefficients in the plot (Figure 3.4).

The procedure continues until only 1 DWT coefficient can be computed at level 9. This one coefficient is the first to be plotted in the DWT plot. This is followed by 2 level 8 coefficients, 4 level 7 coefficients, 8 level 6 coefficients, 16 level 5 coefficients, 32 level 4 coefficients, 64 level 3 coefficients, 128 level 2 coefficients and finally 256 level 1 coefficients. Note that less and less number of samples is used at lower frequencies, therefore, the time resolution decreases as frequency decreases, but since the frequency interval also decreases at low frequencies, the frequency resolution increases. Obviously, the first few coefficients would not carry a whole lot of information, simply due to greatly reduced time resolution.

One area that has benefited the most from this particular property of the wavelet transforms is image processing. As we may well know, images, particularly high-resolution images, claim a lot of disk space. DWT can be used to reduce the image size without losing much of the resolution. Here is how:

For a given image, we can compute the DWT of, say each row, and discard all values in the DWT that are less then a certain threshold. We then save only those DWT coefficients that are above the threshold for each row, and when we need to reconstruct the original image, we simply pad each row with as many zeros as the number of discarded coefficients, and use the inverse DWT to reconstruct each row of the original image. We can also analyze the image at different frequency bands, and reconstruct the original image by using only the coefficients that are of a particular band.

3.5 Discrete Wavelet Transform in Two Dimensions

The DWT in two dimensions can be implemented by filtering with the (1-dimensional) quadrature mirror filters along the rows and columns of an image. Figure 3.6 represents this scheme. In each filtering step we now have a low resolution image and three detail images D_j^1, D_j^2 and D_j^3. Since the detail images are obtained by applying the lowpass and/or highpass filters along rows and columns, they contain the vertical (D_j^1), horizontal (D_j^2) and diagonal (D_j^3) details of the original images at a certain scale j. Figures 3.7 and 3.8 depict the typical organization of these images and examples of wavelet transformed images.

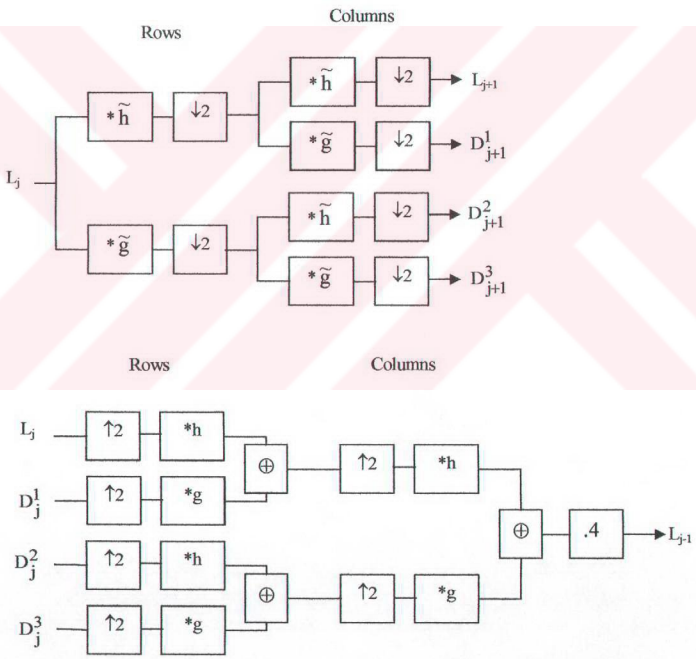


Figure 3.6: Decomposition (top) and reconstruction (bottom) schemes for computation of the wavelet coefficients in 2 dimensions using quadrature mirror filters, * denotes convolution and ↓2 (↑2) downsampling (upsampling) of the rows or columns by a factor of 2

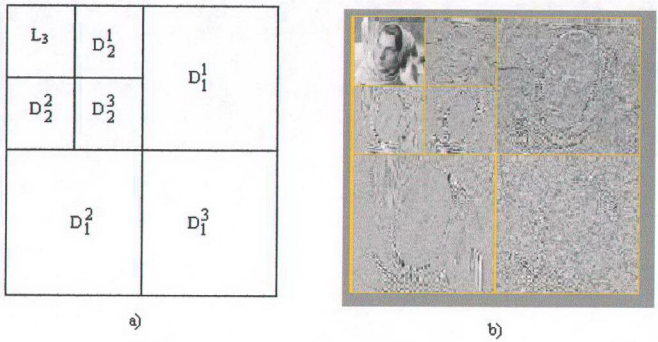


Figure 3.7: a) Typical organization of the detail images within the wavelet transform, b) Example of a wavelet transform of the Woman image (depth 2) [6]

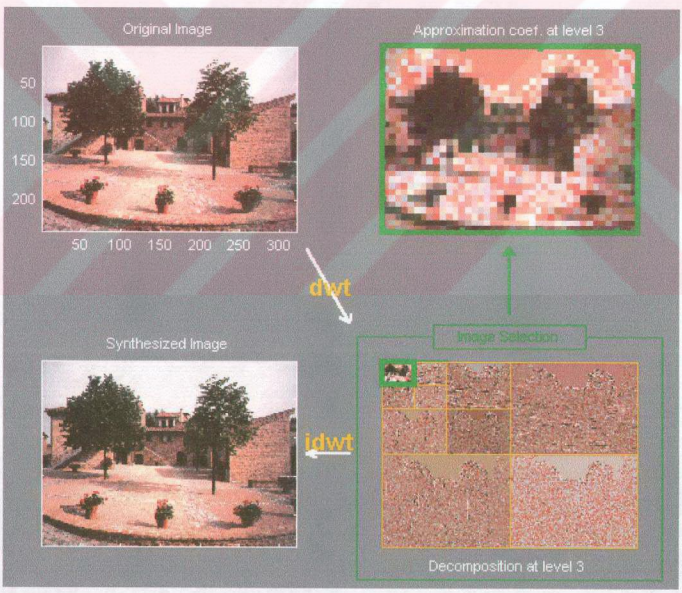


Figure 3.8: Example of a wavelet transform of the Belmont 1 image (depth 3) [6]

CHAPTER 4

WAVELET APPLICATIONS AND FAMİLİES

4.1 Introduction

Wavelets have scale and time aspects consequently every application has scale and time aspects. To clarify them we try to untangle the aspects somewhat arbitrarily.

As a complement to the spectral signal analysis, new signal forms appear. They are less regular signals than the usual ones. To illustrate this notion physically, imagine you take a piece of aluminum foil. The surface is very smooth, very regular. You first crush it into a ball, and then you spread it out so that it looks like a surface. The asperities are clearly visible. Each one represents a two-dimension cusp and analog of the one dimensional cusp. If you crush again the foil, more tightly, in a more compact ball, when you spread it out, the roughness increases and the regularity decreases.

Several domains use the wavelet techniques of regularity study [6]:

•Biology for cell membrane recognition, to distinguish the normal from the pathological membranes,
•Metallurgy for the characterization of rough surfaces,
•Finance (which is more surprising), for detecting the properties of quick variation of values,
•In Internet traffic description, for designing the services size.

Let's switch to time aspects. The main goals are:

•Rupture and edges detection,
•Study of short-time phenomena as transient processes.

As domain applications, we get:

•Industrial supervision of gear-wheel,
•Checking undue noises in craned or dented wheels, and more generally in non destructive control quality processes,
•Detection of short pathological events as epileptic crises or normal ones as evoked potentials in EEG (medicine),
•Automatic target recognition,
•Intermittence in physics.

Many applications use the wavelet decomposition taken as a whole. The common goals concern the signal or image clearance and simplification, which are parts of denoising or compression. We find many published papers in oceanography and earth studies. One of the most popular successes of the wavelets is the compression of the FBI fingerprints.

When trying to classify the applications by domain, it is almost impossible to sum up several thousand papers written within the last 15 years. Moreover, it is difficult to get information on real-world industrial applications from companies. They understandably protect their own information.

Some domains are very productive. Medicine is one of them. We can find studies on micro-potential extraction in EKGs, on time localization of his bundle electrical heart activity, in ECG noise removal. In EEGs, a quick transitory signal is drowned in the usual one. The wavelets are able to determine if a quick signal exists, and if so, can localize it. There are attempts to enhance mammograms to discriminate tumors from calcifications [1,6].

Another prototypical application is a classification of Magnetic Resonance Spectra. The study concerns the influence of the fat we eat on our body fat. The type of feeding is the basic information and the study is intended to avoid taking a sample of

the body fat. Each Fourier spectrum is encoded by some of its wavelet coefficients. A few of them are enough to code the most interesting features of the spectrum. The classification is performed on the coded vectors.

4.2 Fourier Analysis

Signal analysts already have at their disposal an impressive arsenal of tools. Perhaps the most well-known of these is Fourier analysis, which breaks down a signal into constituent sinusoids of different frequencies that an example is shown in Figure 4.1. Another way to think of Fourier analysis is as a mathematical technique for transforming our view of the signal from time-based to frequency-based.

Figure 4.1: Example of a fourier transform of a signal

For many signals, Fourier analysis is extremely useful because the signal's frequency content is of great importance. So why do we need other techniques, like wavelet analysis?

Fourier analysis has a serious drawback. In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event took place.

If the signal properties do not change much over time, that is, if it is what is called a stationary signal then this drawback isn't very important. However, most interesting signals contain numerous non stationary or transitory characteristics such as drift, trends, abrupt changes, and beginnings and ends of events. These characteristics are often the most important part of the signal, and Fourier analysis is not suited to detecting them.

4.3 Short Time Fourier Analysis

In an effort to correct this deficiency, Dennis Gabor (1946) adapted the Fourier transform to analyze only a small section of the signal at a time, which is a technique called windowing the signal. Gabor's adaptation, called the Short-Time Fourier Transform (STFT), maps a signal into a two-dimensional function of time and frequency (an example is shown in Figure 4.2) [6].

Figure 4.2: Example of a short time fourier transform of a signal

The STFT represents a sort of compromise between the time and frequency based views of a signal. It provides some information about both when and at what frequencies a signal event occurs. However, you can only obtain this information with limited precision, and that precision is determined by the size of the window. While the STFT compromise between time and frequency information can be useful, the drawback is that once you choose a particular size for the time window, that window is the same for all frequencies. Many signals require a more flexible approach one where we can vary the window size to determine more accurately either time or frequency.

4.4 Wavelet Analysis

Wavelet analysis represents the next logical step, a windowing technique with variable sized regions. Wavelet analysis allows the use of long time intervals where we want more precise low frequency information, and shorter regions where we want high frequency information (an example is shown in Figure 4.3).

Figure 4.3: Example of a wavelet transform of a signal

Here's what this looks like in contrast with the time-based, frequency-based, and STFT views of a signal (in Figure 4.4):

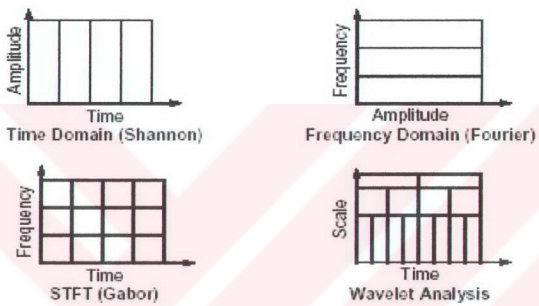


Figure 4.4: Different transforms of a signal

We may have noticed that wavelet analysis does not use a time-frequency region, but rather a time-scale region.

One major advantage afforded by wavelets is the ability to perform local analysis, that is, to analyze a localized area of a larger signal. Consider a sinusoidal signal with a small discontinuity one so tiny as to be barely visible. Such a signal easily could be generated in the real world, perhaps by a power fluctuation or a noisy switch (in Figure 4.5).

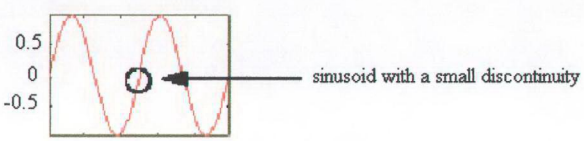


Figure 4.5: A sinusoidal signal with a small discontinuity

A plot of the Fourier coefficients of this signal shows nothing particularly interesting: a flat spectrum with two peaks representing a single frequency. However, a plot of wavelet coefficients clearly shows the exact location in time of the discontinuity, which is shown in Figure 4.6.

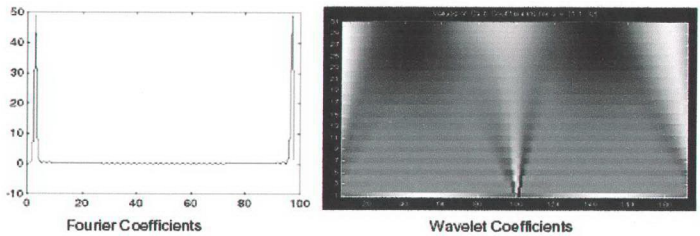


Figure 4.6: Fourier and Wavelet coefficients of a signal

Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. Furthermore, because it affords a different view of data than those presented by traditional techniques, wavelet analysis can often compress or denoise a signal without appreciable degradation. Indeed, in their brief history within the signal processing field, wavelets have already proven themselves to be an indispensable addition to the analyst's collection of tools and continue to enjoy a burgeoning popularity today.

4.5 History of Wavelets

From an historical point of view, wavelet analysis is a new method, though its mathematical underpinnings date back to the work of Joseph Fourier in the nineteenth century. Fourier laid the foundations with his theories of frequency analysis, which proved to be enormously important and influential.

The attention of researchers gradually turned from frequency-based analysis to scale-based analysis when it started to become clear that an approach measuring average fluctuations at different scales might prove less sensitive to noise.

The first recorded mention of what we now call a "wavelet" seems to be in 1909, in a thesis by Alfred Haar. The concept of wavelets in its present theoretical form was first proposed by Jean Morlet and the team at the Marseille Theoretical Physics Center working under Alex Grossmann in France.

The methods of wavelet analysis have been developed mainly by Y. Meyer and his colleagues, who have ensured the methods' dissemination. The main algorithm dates back to the work of Stephane Mallat in 1988. Since then, research on wavelets has become international. Such research is particularly active in the United States, where it is spearheaded by the work of scientists such as Ingrid Daubechies, Ronald Coifman, and Victor Wickerhauser. Barbara Burke Hubbard describes the birth, the history and the seminal concepts in a very clear text [1,6].

The wavelet domain is growing up very quickly. A lot of mathematical papers and practical trials are published every month.

4.6 Wavelet Families

There are different types of wavelet families whose qualities vary according to several criteria. The main criteria are:

•The support of ψ, $\hat{\psi}$ (and ϕ, $\hat{\phi}$): the speed of convergence to 0 of these functions ($\psi(t)$ or $\hat{\psi}(\omega)$) when the time t or the frequency ω goes to infinity, which quantifies both time and frequency localizations,

•The symmetry, which is useful in avoiding dephasing in image processing,

•The number of vanishing moments for ψ or for ϕ (if it exists), which is useful for compression purposes,

•The regularity, which is useful for getting nice features, like smoothness of the reconstructed signal or image, and for the estimated function in nonlinear regression analysis.

These are associated with two properties that allow fast algorithm and space-saving coding:

•The existence of a scaling function ϕ,
•The orthogonality or the biorthogonality of the resulting analysis.

They may also be associated with these less important properties:

•The existence of an explicit expression,
•The ease of tabulating,
•The familiarity with use.

The table below outlines the most common used wavelet families [6].

Table 4.1: The most common used wavelet families

Wavelet Family Short Name	**Wavelet Family Name**
'haar'	Haar wavelet
'db'	Daubechies wavelets
'sym'	Symlets
'coif'	Coiflets
'bior'	Biorthogonal wavelets
'rbio'	Reverse biorthogonal wavelets
'meyr'	Meyer wavelet
'dmey'	Discrete approximation of Meyer wavelet
'gaus'	Gaussian wavelets
'mexh'	Mexican hat wavelet
'morl'	Morlet wavelet
'cgau'	Complex Gaussian wavelets
'shan'	Shannon wavelets
'fbsp'	Frequency B-Spline wavelets
'cmor'	Complex Morlet wavelets

4.6.1 Haar wavelet

Any discussion of wavelets begins with Haar wavelet, the first and simplest. Haar wavelet is discontinuous, and resembles a step function. It represents the same wavelet as Daubechies db1 (Figure 4.7) [6,10].

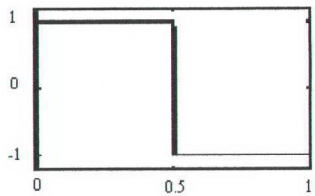


Figure 4.7: Haar wavelet

$$\psi(x) = 1, \quad \text{if} \quad x \in [\,0,\,0.5\,[\tag{4.1}$$

$$\psi(x) = -1, \quad \text{if} \quad x \in [\,0.5,\,1\,[\tag{4.2}$$

$$\psi(x) = 0, \quad \text{if} \quad x \notin [\,0,\,1\,[\tag{4.3}$$

$$\phi(x) = 1, \quad \text{if} \quad x \in [\,0,\,1\,] \tag{4.4}$$

$$\phi(x) = 0, \quad \text{if} \quad x \notin [\,0,\,1\,] \tag{4.5}$$

4.6.2 Daubechies wavelets (dbN)

Ingrid Daubechies, one of the brightest stars in the world of wavelet research, invented what are called compactly supported orthonormal wavelets, thus making discrete wavelet analysis practicable. The names of the Daubechies family wavelets are written dbN, where N is the order, and db the "surname" of the wavelet. Some authors use 2N instead of N. The db1 wavelet, as mentioned above, is the same as Haar wavelet. Here are the wavelet functions psi of the next nine members of the family (Figure 4.8):

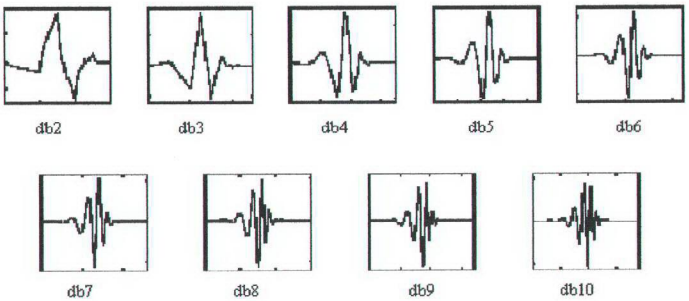


Figure 4.8: Wavelet functions psi of Daubechies wavelets

These wavelets have no explicit expression except for db1, which is the Haar wavelet. However, the square modulus of the transfer function of h is explicit and fairly simple. Let,

$$P(y) = \sum_{k=0}^{N-1} C_k^{N-1+k} y^k \tag{4.6}$$

where C_k^{N-1+k} denotes the binomial coefficients. Then,

$$|m_0(\omega)|^2 = \left(\cos^2\left(\frac{\omega}{2}\right)\right)^N P\left(\sin^2\left(\frac{\omega}{2}\right)\right) \tag{4.7}$$

where

$$m_0(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{2N-1} h_k e^{-j\omega k} \tag{4.8}$$

The support length of ψ and ϕ is 2N-1. The number of vanishing moments of ψ is N. Most dbN are not symmetrical. For some, the asymmetry is very pronounced.

The regularity increases with the order. When N becomes very large, ψ and ϕ belong to $C^{\mu N}$ where μ is approximately equal to 0.2. Certainly, this asymptotic value is too pessimistic foe small order N. Note that the functions are more regular at certain points than at others. The analysis is orthogonal [6,10].

4.6.3 Biorthogonal wavelet pairs (biorNr.Nd)

This family of wavelets exhibits the property of linear phase, which is needed for signal and image reconstruction. By using two wavelets, one for decomposition (on the left side) and the other for reconstruction (on the right side) instead of the same single one, interesting properties are derived (Figure 4.9).

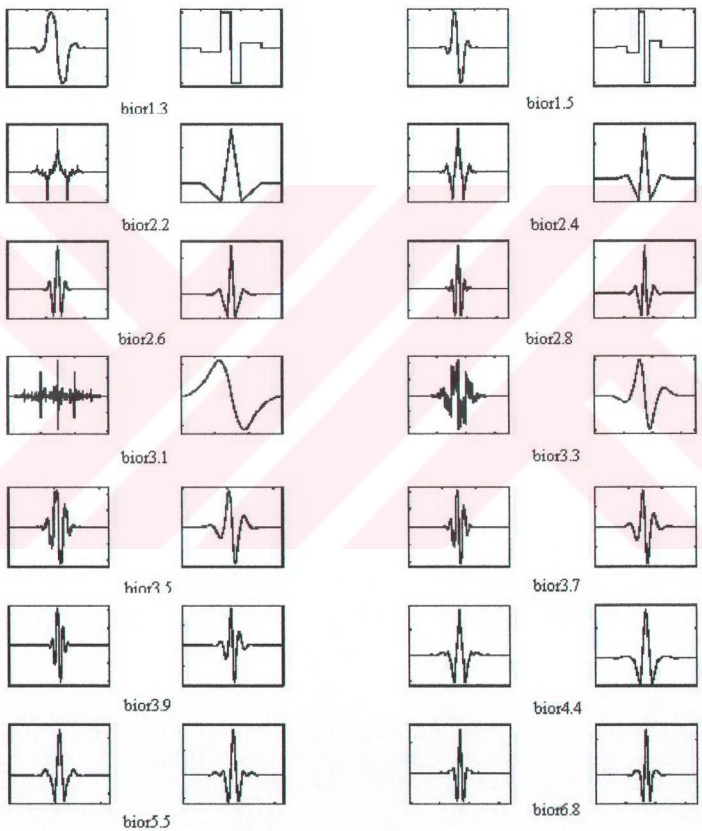


Figure 4.9: Biorthogonal wavelets

The new family extends the wavelet family. It is well known in the subband filtering community that symmetry and exact reconstruction are incompatible (except for the

Haar wavelet) if the same FIR filters are used for reconstruction and decomposition. Two wavelets, instead of just one, are introduced:

One, $\tilde{\psi}$, is used in the analysis, and the coefficients of a signal s are:

$$\tau_{j,k} = \int s(x)\tilde{\psi}_{j,k}(x)dx \tag{4.9}$$

The other, ψ, is used in the synthesis:

$$s = \sum_{j,k} \tau_{j,k}\psi_{j,k} \tag{4.10}$$

In addition, the wavelets ψ and $\tilde{\psi}$ are related by duality in the following sense:

$$\int \tilde{\psi}_{j,k}(x)\psi_{j',k'}(x)dx = 0 \quad \text{as soon as} \quad j \neq j' \quad \text{or} \quad k \neq k' \quad \text{and even} \tag{4.11}$$

$$\int \tilde{\phi}_{0,k}(x)\phi_{0,k'}(x)dx = 0 \quad \text{as soon as} \quad k \neq k' \tag{4.12}$$

It becomes apparent, as Cohen pointed out in his thesis, that "the useful properties for analysis (e.g. oscillations, zero moments) can be concentrated on the $\tilde{\psi}$ function whereas the interesting properties for synthesis (regularity) are assigned to the ψ function. The separation of these two tasks proves very useful".

$\tilde{\psi}$ and ψ can have very different regularity properties. The $\tilde{\psi}$, ψ, $\tilde{\phi}$ and ϕ functions are zero outside of a segment. The calculation algorithms are maintained, and thus very simple. The filters associated with m_0 and $\tilde{m}_0$ can be symmetrical. The functions used in the calculations are easier to build numerically than those used in the usual wavelets [6,10,12].

4.6.4 Coiflet wavelets (coifN)

In coifN, N is the order. Some authors use 2N instead of N. Built by I. Daubechies at the request of R. Coifman. The wavelet function ψ has 2N moments equal to 0 and, what is more unusual, the function ϕ has 2N-1 moments equal to 0. The two functions have a support of length 6N-1 (Figure 4.10).

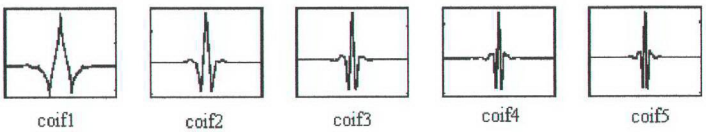


Figure 4.10: Coiflet wavelets

The coifN ψ and ϕ are much more symmetrical than the dbNs. With respect to the support length, coifN has to be compared to db3N or sym3N. With respect to the number of vanishing moments of ψ, coifN has to be compared to db2N or sym2N.

If s is a sufficiently regular continuous time signal, for large j the coefficient $\langle s, \phi_{-j,k} \rangle$ is approximately by $2^{-j/2} s(2^{-j}k)$. If s is a polynomial of degree d, $d \leq N-1$, then the approximation becomes an equality. This property is used, connected with sampling problems, when calculating the difference between an expansion over the $\phi_{j,k}$ of a given signal and its sampled version [6,10].

4.6.5 Symlet wavelets (symN)

In symN, N is the order. Some authors use 2N instead of N. The symlets are nearly symmetrical wavelets proposed by Daubechies as modifications to the db family. The properties of the two wavelet families are similar. Here are the wavelet functions psi (Figure 4.11).

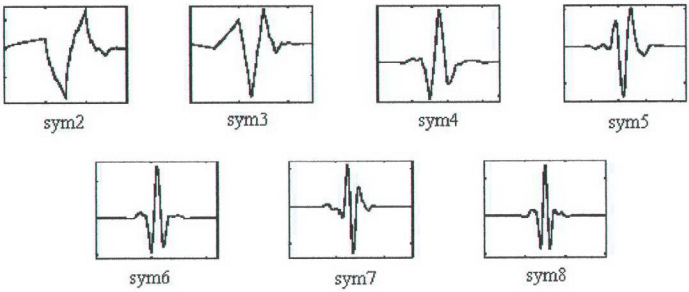


Figure 4.11: Symlet wavelets

Daubechies proposes modifications of her wavelets that increase their symmetry can be increased while retaining great simplicity. The idea consists of reusing the function m_0 introduced in the dbN, considering the $|m_0(\omega)|^2$ as a function W of $z = e^{j\omega}$. Then we can factor W in several different ways in the form of $W(z) = U(z)\overline{U\left(\frac{1}{z}\right)}$ because the roots of W with modulus not equal to 1 go in pairs. If one of the root is z_1, then $\frac{1}{z_1}$ is also a root. By selecting U such that the modulus of all its roots is strictly less than 1, we build Daubechies wavelets dbN. The U filter is a "minimum phase filter". By making another choice, we obtain more symmetrical filters; these are symlets. The symlets have other properties similar to those of the dbNs [6,10].

4.6.6 Morlet wavelet (morl)

This wavelet has no scaling function, but is explicit (Figure 4.12).

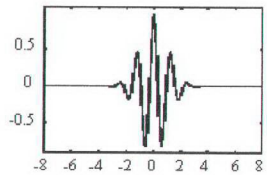


Figure 4.12: Morlet wavelet

$$\psi(x) = Ce^{-x^2/2}\cos(5x) \tag{4.13}$$

The constant C is used for normalization in view of reconstruction. The Morlet wavelet does not satisfy exactly the admissibility condition [6,10].

4.6.7 Mexican hat wavelet (mexh)

This wavelet has no scaling function and is derived from a function that is proportional to the second derivative function of the Gaussian probability density function (Figure 4.13).

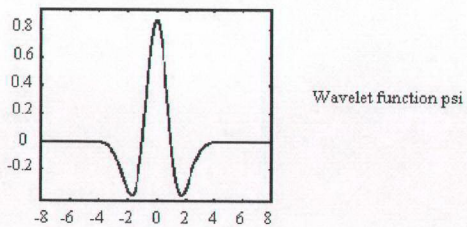


Figure 4.13: Mexican hat wavelet

$$\psi(x) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)\left(1 - x^2\right)e^{-x^2/2} \tag{4.14}$$

This function is proportional to the second derivative function of the Gaussian probability density function. As the ϕ function does not exist, the analysis is not orthogonal [6,10].

4.6.8 Meyer wavelet (meyr)

Both ψ and ϕ are defined in the frequency domain, starting with an auxiliary function v (Figure 4.14).

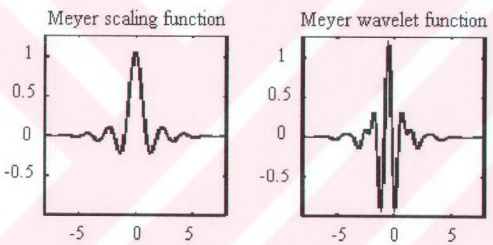


Figure 4.14: The Meyer wavelet

The Meyer wavelet and scaling function are defined in the frequency domain. Wavelet function is:

$$\hat{\psi}(\omega) = (2\pi)^{-1/2} e^{j\omega/2} \sin\left(\frac{\pi}{2} v\left(\frac{3}{2\pi}|\omega| - 1\right)\right) \quad \text{if} \quad \frac{2\pi}{3} \le |\omega| \le \frac{4\pi}{3} \tag{4.15}$$

$$\hat{\psi}(\omega) = (2\pi)^{-1/2} e^{j\omega/2} \cos\left(\frac{\pi}{2} v\left(\frac{3}{4\pi}|\omega| - 1\right)\right) \quad \text{if} \quad \frac{4\pi}{3} \le |\omega| \le \frac{8\pi}{3} \tag{4.16}$$

and

$$\hat{\psi}(\omega) = 0 \quad \text{if} \quad |\omega| \notin \left[\frac{2\pi}{3}, \frac{8\pi}{3}\right] \tag{4.17}$$

where

$$v(a) = a^4\left(35 - 84a + 70a^2 - 20a^3\right), a \in [0,1] \qquad (4.18)$$

Scaling function is:

$$\hat{\phi}(\omega) = (2\pi)^{-1/2} \quad \text{if} \quad |\omega| \le \frac{2\pi}{3} \qquad (4.19)$$

$$\hat{\phi}(\omega) = (2\pi)^{-1/2} \cos\left(\frac{\pi}{2} v\left(\frac{3}{2\pi}|\omega| - 1\right)\right) \quad \text{if} \quad \frac{2\pi}{3} \le |\omega| \le \frac{4\pi}{3} \qquad (4.20)$$

$$\hat{\phi}(\omega) = 0 \quad \text{if} \quad |\omega| \succ \frac{4\pi}{3} \qquad (4.21)$$

By changing the auxiliary function, we get a family of different wavelets. This wavelet ensures orthogonal analysis.

The function ψ does not have finite support, but ψ decreases to 0 when $x \to \infty$, faster than any inverse polynomial such as:

$$\forall n \in N, \exists C_n \quad \text{such that} \quad |\psi(x)| \le C_n\left(1 + |x|^2\right)^{-n} \qquad (4.22)$$

This property holds also for the derivatives:

$$\forall k \in N, \forall n \in N, \exists C_{k,n} \quad \text{such that} \quad \left|\psi^{(k)}x\right| \le C_{k,n}\left(1 + |x|^2\right)^{-n} \qquad (4.23)$$

The wavelet is infinitely differentiable. Although the Meyer wavelet is not compactly supported, there exists a good approximation leading to FIR filters, and then allowing DWT [6,10].

4.6.9 Battle Lemarie wavelets

There are two forms of the wavelet: one does not ensure the analysis to be orthogonal, while the other does. For N = 1, the scaling functions are linear splines. For N = 2, the scaling functions are quadratic B-spline with finite support. More generally, for an N-degree B-splines:

$$\hat{\phi}(\omega) = (2\pi)^{-1/2} e^{-j\omega K/2} \left[\frac{\sin(\omega/2)}{\omega/2} \right]^{N+1} \tag{4.24}$$

with K = 0 if N is odd, K = 1 if N is even.

This formula can be used to build the filters. The twin scale relation is:

$$\phi(x) = 2^{-2M} \sum_{j=0}^{2M+1} C_j^{2M+1} \phi(2x - M - 1 + j) \quad \text{if} \quad N = 2M \tag{4.25}$$

$$\phi(x) = 2^{-2M-1} \sum_{j=0}^{2M+2} C_j^{2M+2} \phi(2x - M - 1 + j) \quad \text{if} \quad N = 2M+1 \tag{4.26}$$

For an even N, ϕ is symmetrical around, x=1/2; ψ is anti-symmetrical around x=1/2. For an odd N, ϕ is symmetrical around x = 0; ψ is symmetrical around x=1/2.

The analysis becomes orthogonal if we transform the functions ψ and ϕ somewhat. For N = 1, for instance, let:

$$\overset{\Lambda}{\phi}^{\perp}(\omega) = 3^{1/2} (2\pi)^{-1/2} \frac{4\sin^2(\omega/2)}{\omega^2 \left[1 + 2\cos^2(\omega/2)\right]^{1/2}} \tag{4.27}$$

The supports of ψ and $\phi^{\perp}$ are not finite, but the decrease of the functions ψ and $\phi^{\perp}$ to 0 is exponential. The support of ϕ is compact. The ψ functions have derivatives up to order N-1 [6,10].

CHAPTER 5

TEXTURE ANALYSIS AND CLASSIFICATION EXPERIMENTS

5.1 Introduction

This chapter discusses the extraction of texture features from images. In the recent past, some authors have conjectured that multiscale decomposition is a natural basis for texture characterization [1]. This conjecture is formalized by defining texture as the set of local statistical properties of the coefficients which constitute the image's multiscale representation. Features to reflect these statistics are described and evaluated.

Texture analysis plays an important role in many image processing tasks, ranging from remote sensing to medical imaging, robot vision, query by content in large image databases. Texture analysis is usually dealt with in two major phases. The first phase is feature extraction in which the image information is reduced to a small set of descriptive features. The second step deals with classifying the features obtained from individual pixels (e.g. texture segmentation) or a collection of pixels (e.g. image classification). Various methods for texture feature extraction have been proposed during the last decades, but the texture analysis problem remains difficult and subject to intensive research [1].

A major class of feature extractors relies on the assumption that texture can be defined by the local statistical properties of pixel gray levels. These properties are considered to be translation invariant and (approximate) periodic. From the image histogram, which satisfies these conditions, first order statistics can be derived and used as texture features.

A weakness shared by all these texture analysis schemes is that the image is analyzed at one single scale, a limitation which can be lifted by employing multiscale representations. Studies in the human visual system support this approach since researchers have found that the visual cortex can be modeled as a set of independent channels, each with a particular orientation and spatial frequency tuning [1,13].

Several multichannel texture analysis systems have been developed [14]. In particular, Gabor filters were employed to perform texture segmentation [2,4]. In the last decade, wavelet theory has emerged and became a mathematical framework which provides a more formal, solid and unified framework for multiscale image analysis. Typically, the wavelet transform maps an image on a low resolution image and a series of detail images. The low resolution image is obtained by iteratively blurring the image; the detail images contain the information lost during this operation. The mean or energy deviation of the detail images are the most commonly used features for texture classification and segmentation problems [1].

First order statistical information is derived from the detail image histogram. As observed by Mallat, the detail histograms of natural textured images can be modeled by a family of exponential functions [1,15]. Introducing the parameters of this model as texture features completely describes the wavelet coefficients first order statistics. In the following section, a very brief review of the wavelet representation will be given.

5.2 The Wavelet Representation

The (continuous) wavelet transform of a 1-dimensional signal f(x) is defined as

$$(W_a f)(b) = \int f(x)\psi^*_{a,b}(x)dx \qquad (5.1)$$

where the wavelet $\psi_{a,b}$ is computed from the mother wavelet ψ by translation and dilation:

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \tag{5.2}$$

The mother wavelet ψ has to satisfy the admissibility criterion to ensure that it is a localized zero-mean function [1,10].

(5.1) can be discretized by restraining **a** and **b** to a discrete lattice ($\mathbf{a} = 2^n$, $\mathbf{b} \in Z$). Typically some more constraints are imposed on ψ to ensure that the transform is complete (non-redundant) and constitutes a multiresolution representation of the original signal. This has led to an efficient real-space implementation of the transform using quadrature mirror filters.

The extension to the 2-dimensional case is usually performed by using a product of 1-dimensional filters. In practice the transform is computed by applying a separable filter bank to the image:

$$L_n(b_i,b_j) = [H_x * [H_y * L_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j) \tag{5.3}$$

$$D_{n1}(b_i,b_j) = [H_x * [G_y * L_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j) \tag{5.4}$$

$$D_{n2}(b_i,b_j) = [G_x * [H_y * L_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j) \tag{5.5}$$

$$D_{n3}(b_i,b_j) = [G_x * [G_y * L_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j) \tag{5.6}$$

* denotes the convolution operator, $\downarrow 2,1$ ($\downarrow 1,2$) subsampling along the rows (columns) and $L_0 = I(\vec{x})$ is the original image. H and G are a low and bandpass filter respectively. L_n is obtained by lowpass filtering and is therefore referred to as the low resolution image at scale n. The D_{ni} are obtained by bandpass filtering in a specific direction and thus contain directional detail information at scale n; they are referred to as the detail images. The original image I is thus represented by a set of subimages at several scales: $\{L_d, D_{ni}\}_{i=1,2,3;n=1...d}$ which is a multiscale representation of depth d of the image I [1,10].

5.3 Wavelet Mean Deviation Signatures

The mean deviation (MD signature) of a subimage D_{ni} is defined as

$$MD_{ni} = \frac{1}{N}\sum_{j,k}\left|D_{ni}(b_j, b_k)\right| \tag{5.7}$$

where N is the total number of wavelet coefficients in D_{ni}. The mean deviation signatures $\{MD_{ni}\}_{n=1...d,i=1,2,3}$ reflect the distribution of energy along the frequency axis over scale and orientation and have proven to be very powerful for texture characterization [1].

An alternative measure which is sometimes used as a texture feature is the energy signature:

$$E_{ni} = \frac{1}{N}\sum_{j,k}\left(D_{ni}(b_j, b_k)\right)^2 \tag{5.8}$$

Since most relevant texture information has been removed by iteratively lowpass filtering, the energy of the low resolution image L_d is generally not considered a texture feature. That is why, the energies of the detail images are used as texture features.

Although these features have been successfully used for classification and segmentation of textured images, some authors have attempted to increase classification performance by adding other measures, such as the variance of the mean deviation signatures, histogram signatures, cooccurrence signatures etc [1].

In this thesis, the mean deviation signatures are used as texture features. And in order to increase the classification performance, the standart and median absolute deviations are used additionally. We used these signatures of horizontal, vertical and diagonal wavelet detail coefficients together in order to increase the total number of features used and of course classification performance.

5.4 Pattern Recognition

Pattern recognition constitutes a very important part of this thesis. Once obtained the data sets from the analysis, we have to recognize these data sets, that is, classify them such as to which class they belong correctly. A workable definition for pattern recognition is given by Vapnik: "A person (the instructor) observes occurring situations and determines to which of c classes each one of them belongs. It is required to construct a device (the classifier) which, after observing the instructors procedure, will carry out the classification approximately in the same manner as the instructor"[16].

Pattern recognition problems can be subdivided into 2 types. The first one assumes that the class label of each sample is known (i.e. an "instructor" is available) and that use of this information is made during classifier design (i.e. the construction of the classifier). This type is called supervised learning. An example is the classification of textured images into one of c predefined classes. The second type consist of unsupervised learning algorithms which must be employed if for a certain reason the class labels are not available and the pattern classes must be inferred from the available data. An example of the latter is image segmentation in which each pixel is classified to a certain class for the purpose of subdividing the image into c homogeneous regions [1].

The "occurring situations" in the above definition are the patterns one wishes to recognize. A pattern can be any abstract object like an image, a face, a sound, a trend in curve, etc. Usually, there is some preprocessing to be done on this object (the feature extraction) so that one object is represented by a discrete number of values: the features, usually seen as elements of a feature vector. If from each pattern d features are extracted, the corresponding feature vector represents a point in a d-dimensional space which is called the feature space.

Classifying patterns is then equivalent to assigning a class to a particular feature vector. Designing a classifier means finding the decision rule: i.e. that rule which, given feature vector $\vec{x}$, assigns it to class ω_i. It can be seen that the decision rule

partitions the feature space into separate regions; if a feature vector is located inside a particular region, it will be assigned to the class associated with that region.

We will use the term data sample to denote either a pattern or the feature vector derived from it. The class label denotes to which class the sample belongs. The collection of data samples available to design a classifier is called the design set.

It is obvious that a good classifier which is capable of accurately recognizing the patterns relies on good features. By "good" features we mean that patterns of the same class yield similar feature vectors. In this way, feature vectors from one class form a cluster in feature space which should be well separated from the clusters in other classes [1,2,5].

There are several classifiers which may be used for classification problems. Some of them are explained briefly in the following parts.

5.4.1 Knn-classifier

In Knn-classifier (K nearest neighbor classifier), one assumes that the feature space is a metric space; i.e. there exists some function $d(\vec{x},\vec{y})$ which expresses the distance between two points x and y in feature space. A frequently used one is the Mahalanobis distance:

$$d(\vec{x},\vec{y}) = \sum_{i=1}^{d} (\vec{x}-\vec{y})^T C^{-1} (\vec{x}-\vec{y}) \qquad (5.9)$$

where C is the covariance matrix estimated from the design set. If C is the identity matrix, we obtain the Euclidian distance:

$$d(\vec{x},\vec{y}) = \sum_{i=1}^{d} (x_i - y_i)^2 \qquad (5.10)$$

If the Mahalanobis distance is used, the region S is a hyperellipsoid in feature space; in the Euclidean case it is a hypersphere [1].

Note that, when the Euclidean distance measure is used, some features (the ones with the largest variance across the design set) tend to dominate this measure. It is therefore useful to normalize the features. If we denote the i^{th} feature from the j^{th} pattern by x_i^j then normalization is performed by

$$x_i'^j = \frac{x_i^j - \overline{x_i}}{\sigma_i} \tag{5.11}$$

where the mean $\overline{x_i}$ and variance σ_i^2 of the i^{th} feature are defined by

$$\overline{x_i} = \frac{1}{N}\sum_{j=1}^{N} x_i, \quad \sigma_i^2 = \frac{1}{N}, \quad \sum_{j=1}^{N}(x_i^j - \overline{x_i})^2 \tag{5.12}$$

A problem is that this normalization could be affected by outliers. These are points in feature space which, due to uncontrollable causes, deviate from the underlying probability density and deteriorate the estimates of $\overline{x_i}$ and σ_j. A solution is to replace $\overline{x_i}$ by the median $m(x_i)$ of the x_i^j and σ_i by $(1/N)\sum_{j=1}^{N}\left|x_i^j - m(x_i)\right|$ in (5.11)-(5.12), which are known to be less affected by outliers. Whenever we will state that the features are normalized, we refer to the last procedure. This leads to a very simple classification procedure: the sample $\vec{x}$ should be assigned to that class to which most of its K nearest neighbors belong. It is optimal in the sense that it minimizes the conditional risk [1].

5.4.2 Neural networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a

neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in Figure 5.1. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network [6].

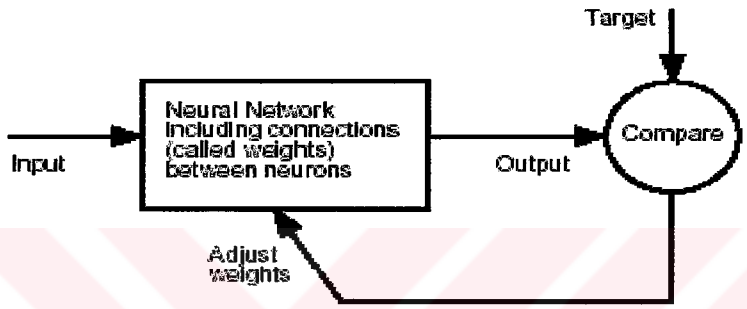


Figure 5.1: Basic principle of a neural network

Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as "on line" or "adaptive" training.

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems. Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings.

The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make.

The field of neural networks has a history of some five decades but has found solid application only in the past fifteen years, and the field is still developing rapidly. Thus, it is distinctly different from the fields of control systems or optimization where the terminology, basic mathematics, and design procedures have been firmly established and applied for many years.

5.4.3 The LVQ network

The learning vector quantization (LVQ) network architecture is shown in Figure 5.2 [6]:

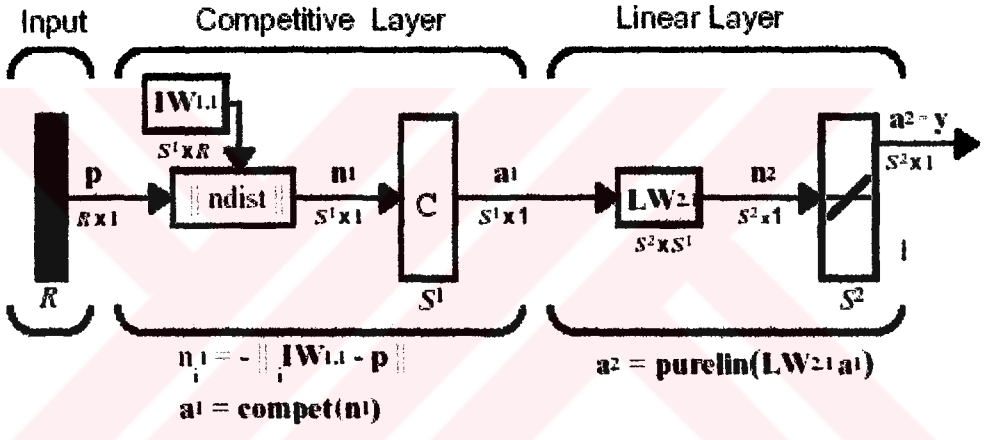


Figure 5.2: The learning vector quantization network architecture

where R denotes the number of elements in input vector, S^1 denotes the number of competitive neurons and S^2 denotes the number of linear neurons.

An LVQ network has a first competitive layer and a second linear layer. The competitive layer learns to classify input vectors in much the same way as the competitive layers. The linear layer transforms the competitive layer's classes into target classifications defined by the user. We refer to the classes learned by the competitive layer as subclasses and the classes of the linear layer as target classes.

Both the competitive and linear layers have one neuron per (sub or target) class. Thus, the competitive layer can learn up to S^1 subclasses. These, in turn, are combined by the linear layer to form S^2 target classes (S^1 is always larger than S^2).

LVQ learning in the competitive layer is based on a set of input/target pairs.

$$\{p_1, t_1\}, \{p_2, t_2\}, \ldots, \{p_Q, t_Q\} \tag{5.13}$$

Each target vector has a single 1. The rest of its elements are 0. The 1 tells the proper classification of the associated input. For instance, consider the following training pair.

$$\left\{ p_1 = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, t_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right\} \tag{5.14}$$

Here we have input vectors of three elements, and each input vector is to be assigned to one of four classes. The network is to be trained so that it classifies the input vector shown above into the third of four classes.

To train the network, an input vector p is presented, and the distance from p to each row of the input weight matrix $IW^{1,1}$ is computed. The hidden neurons of layer 1 compete. Suppose that the i^{th} element of n^1 is most positive, and neuron i* wins the competition. Then the competitive transfer function produces a 1 as the i^{*th} element of a^1. All other elements of a^1 are 0.

When a^1 is multiplied by the layer 2 weights $LW^{2,1}$, the single 1 in a^1 selects the class, k* associated with the input. Thus, the network has assigned the input vector p to class k* and a^2_{k*} will be 1. Of course, this assignment may be a good one or a bad one, for t_{k*} may be 1 or 0, depending on whether the input belonged to class k* or not.

We adjust the i^{*th} row of $IW^{1,1}$ in such a way as to move this row closer to the input vector p if the assignment is correct, and to move the row away from p if the assignment is incorrect. So if p is classified correctly,

$$\left(a^2_{k^*} = t_{k^*} = 1\right) \tag{5.15}$$

we compute the new value of the i^{*th} row of $IW^{1,1}$ as:

$$i^*IW^{1,1}(q) = i^*IW^{1,1}(q-1) + \alpha(p(q) - i^*IW^{1,1}(q-1)) \tag{5.16}$$

On the other hand, if p is classified incorrectly,

$$\left(a^2_{k^*} = 1 \neq t_{k^*} = 0\right) \tag{5.17}$$

we compute the new value of the i^{*th} row of $IW^{1,1}$ as:

$$i^*IW^{1,1}(q) = i^*IW^{1,1}(q-1) - \alpha(p(q) - i^*IW^{1,1}(q-1)) \tag{5.18}$$

These corrections to the i^{*th} row of $IW^{1,1}$ can be made automatically without affecting other rows of $IW^{1,1}$ by back propagating the output errors back to layer 1. Such corrections move the hidden neuron towards vectors that fall into the class for which it forms a subclass, and away from vectors that fall into other classes [1].

5.4.4 Error calculation

Until now, we have been concerned with designing a classifier from a given design set. However, after classifier design, it is also important to evaluate the classifier performance, that is, how good is the classifier. A "good" classifier should accurately classify any possible pattern for which it is designed. To validate this, samples should be used which were not used for design. In this way, the generalization ability of the classifier (the ability to recognize "unseen samples") is tested [1,2,5].

Let us define the true error rate e_C as the probability of misclassification associated with the classifier C. Calculation of this quantity is usually performed by error counting. In this, a collection of data samples not used for training (the test set) is

presented to the classifier and the percentage of samples classified falsely is determined. Suppose that there are N_t test samples, let τ_i be a variable which takes the value 0 if the i^{th} test sample is classified correctly and 1 otherwise. The classification performance of the classifier is then calculated by the (observed) error rate:

$$\hat{e}_C = \frac{1}{N_t}\sum_{i=1}^{N_t}\tau_i \tag{5.19}$$

Of course, $\hat{e}_C$ depends on the test set and is therefore itself a random variable. Therefore, this calculation should be accompanied by a confidence interval which reflects its accuracy. For a large enough test set ($N_t > 30$), a 95% confidence interval is given by:

$$\hat{e}_C - \hat{\sigma}(\hat{e}_C) \le e_C \le \hat{e}_C + \hat{\sigma}(\hat{e}_C) \tag{5.20}$$

with

$$\hat{\sigma}(\hat{e}_C) = 1.96\sqrt{\frac{\hat{e}_C(1-\hat{e}_C)}{N_t}} \tag{5.21}$$

In practice, only a finite number N of data samples is available. Using all samples for designing and afterwards for testing the classifier (the re substitution method) gives a negative bias to the error rate since test and design set are not independent.

Subdividing the available data into a design set (containing N_d samples) and a test set (N_t samples, $N_d + N_t = N$) and calculating the error rate as above is called the holdout method.

The holdout method is not very economical since part of the data is not used for classifier design. This is solved by the leave-k-out method, where k samples are used as a test set, while the remaining (N-k) samples are used for design. Next, the k test samples are added to the design set from which k other samples are now used for testing. This procedure is repeated until each available sample is exactly used once for testing.

5.5 Classification Experiments

We made several classification experiments with various textured images. Specifically we got eight real world 512x512 images from different natural scenes from the VisTex database which are presented in Figure 5.3 [17].

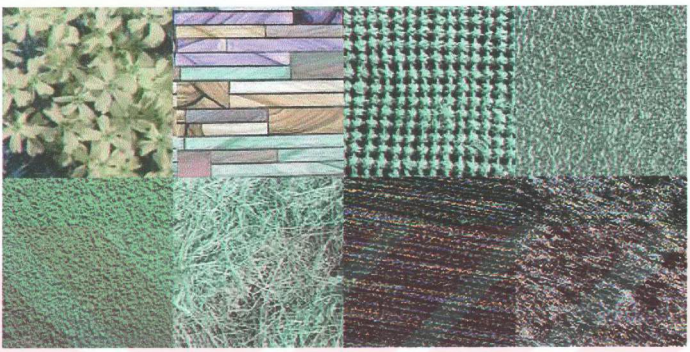


Figure 5.3: Selected images, from left to right and top to bottom: Flowers3, Brick2, Fabric8, Metal3, Food5, Grass1, Sand5, Leaves3

Each 512x512 image is subdivided into 64 non-overlapping 64x64 subimages. That's way a database of 512 image regions of 8 texture classes is constructed. These 512 images are analyzed with Biorthogonal Wavelets of order 2.2 and level 1 and statistical values of wavelet detail coefficients of each subimage are obtained.

Here we obtained a total of 9 features for each subimage which are standart deviation, median absolute deviation and mean absolute deviation of horizontal wavelet detail coefficients, vertical wavelet detail coefficients and diagonal wavelet detail coefficients. Since every 64 subimages belong to one separate class, we computed the means of their statistical values and assigned them to their class. An example analysis is shown in Figure 5.4.

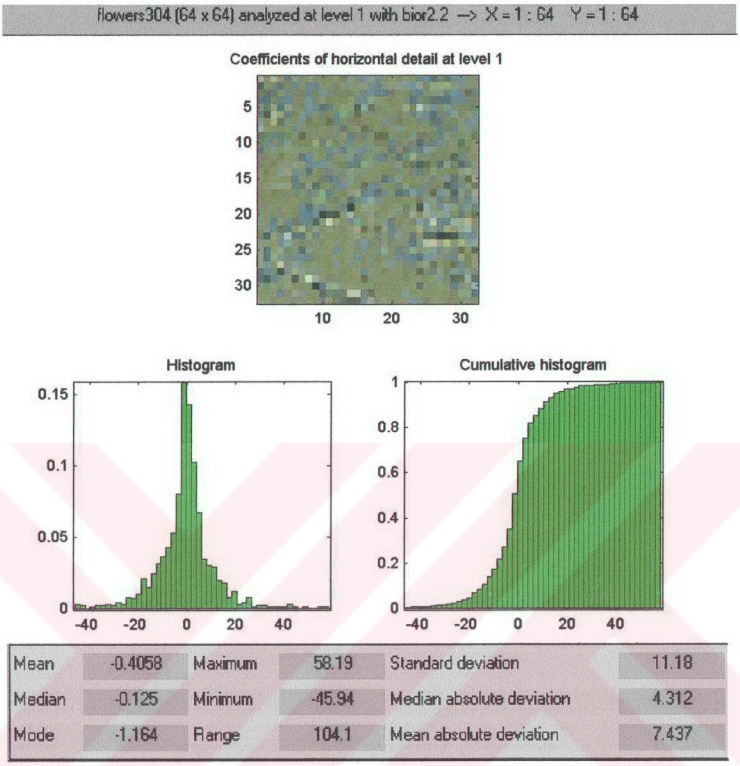


Figure 5.4: Example of a wavelet analysis of a subimage

Then these 512 subimages are classified using Knn-classifier algorithm and the error rates in all classification experiments are computed. To obtain the possible minimum error rate, a total of 17 classification experiments are made. In each experiment, different possible combinations of features are tried. For example, first the standart deviation is used alone, then median absolute deviation and after that mean absolute deviation is used. And finally these three features are used together and the results obtained from each experiment are compared.

CHAPTER 6

CONCLUSIONS AND DISCUSSIONS

The features used in the classification experiments are listed in Table 6.1, and the resulting error rates of 17 classification experiments are presented in tables 6.2, 6.3 and 6.4.

We can derive some conclusions from the resulting error rates of the 17 classification experiments:

1. There is a slight difference between the performances of vertical and diagonal coefficients such as 24,22% in CE8 and 26,96% in CE12. These are quite high error rates and therefore vertical or diagonal wavelet detail coefficients, when they are used alone, are not suitable features for texture classification.

2. Horizontal wavelet detail coefficients are very powerful with respect to vertical and diagonal coefficients. The error rate of horizontal coefficients (12,31% in CE4) is about half of the other ones (24,22% in CE8 and 26,96% in CE12).

3. When the number of features used increases, then the classification performance also increases and the mean error rate decreases. Inversely, when the number of features used decreases, then the classification performance also decreases and the mean error rate increases. For example, the mean error rate in CE 1 is 19,53% where only one feature is used. The mean error rate in CE 4 is 12,31% where three features are used. But the lowest mean error rate for 8 texture classes, is achieved in CE16 as 7,228% where all of the 9 features are used together. This is an acceptable low error rate therefore we can say that, when used together, the standard, median and mean

absolute deviations of horizontal, vertical and diagonal wavelet detail coefficients are suitable features for texture classification.

4. The lowest mean error rate of all the experiments is achieved in CE 17 as 3,125% where 6 texture classes were used. This is a very low error rate therefore, we can conclude that, when the number of texture classes increases then the error rate also increases and of course the classification performance decreases. But when the number of texture classes decreases, then the error rate also decreases and the classification performance increases.

5. These wavelet features are more suitable for some texture classes while for the others not. For example, for class Metal3 we got zero error rate means all the classifications for this class were correct. But for class Grass1, the error rate is quite high such as 15,63%.

As a result we have found that the selection and the number of features of a given data set, were very important for a classification. If selected features are suitable and contain enough information about given samples, then any classifier will classify successfully. And the number of features also plays an important role in classifying tasks.

G. Van De Wouwer has obtained an error rate of 2.5% in his experiments which is a little smaller than our result (3.125% in CE17) [1] and A. Drimbarean has obtained an error rate of 9% in his experiments which is quite greater than our result [2].

In this thesis we have seen that, when used together, Standard Deviation, Median Absolute Deviation and Mean Absolute Deviation of horizontal, vertical and diagonal wavelet detail coefficients yield to an acceptable low error rate in texture classification.

For future work, the number of features used may be increased or different wavelet families may be used in the analysis for achieving lower error rates in classification.

Table 6.1: The features used in the classification experiments

Classification Experiments	Features Used
CE 1	Standart deviations of horizontal wavelet detail coefficients
CE 2	Median absolute deviations of horizontal wavelet detail coefficients
CE 3	Mean absolute deviations of horizontal wavelet detail coefficients
CE 4	Standart, Median and Mean absolute deviations of horizontal wavelet detail coefficients
CE 5	Standart deviations of vertical wavelet detail coefficients
CE 6	Median absolute deviations of vertical wavelet detail coefficients
CE 7	Mean absolute deviations of vertical wavelet detail coefficients
CE 8	Standart, Median and Mean absolute deviations of vertical wavelet detail coefficients
CE 9	Standart deviations of diagonal wavelet detail coefficients
CE 10	Median absolute deviations of diagonal wavelet detail coefficients
CE 11	Mean absolute deviations of diagonal wavelet detail coefficients
CE 12	Standart, Median and Mean absolute deviations of diagonal wavelet detail coefficients
CE 13	Standart, Median and Mean absolute deviations of horizontal and vertical wavelet detail coefficients
CE 14	Standart, Median and Mean absolute deviations of horizontal and diagonal wavelet detail coefficients
CE 15	Standart, Median and Mean absolute deviations of vertical and diagonal wavelet detail coefficients
CE 16	Standart, Median and Mean absolute deviations of horizontal, vertical and diagonal wavelet detail coefficients
CE 17	Standart, Median and Mean absolute deviations of horizontal, vertical and diagonal wavelet detail coefficients (6 texture classes are used)

Table 6.2: Error rates of classification experiments from 1 to 8

Texture Classes		% error rates in horizontal wavelet detail coefficients				% error rates in vertical Wavelet detail coefficients			
		CE 1	CE 2	CE 3	CE 4	CE 5	CE 6	CE 7	CE 8
1	Flowers 3	6,25	4,69	6,25	6,25	4,69	3,13	4,69	4,69
2	Brick 2	17,19	17,19	9,38	9,38	65,63	46,88	56,25	54,69
3	Fabric 8	48,44	26,56	48,44	9,38	7,81	40,63	20,31	10,94
4	Metal 3	15,63	7,81	46,88	0	20,31	28,13	20,31	18,75
5	Food 5	25	18,75	17,19	17,19	39,06	50	40,63	42,19
6	Grass 1	32,81	43,75	34,38	37,5	39,06	31,25	32,81	32,81
7	Sand 5	6,25	14,06	15,63	9,38	46,88	17,19	20,31	18,75
8	Leaves 3	4,69	10,94	7,81	9,38	9,38	28,13	10,94	10,94
Means		19,5325	17,9688	23,245	12,3075	29,1025	30,6675	25,7813	24,22

Table 6.3: Error rates of classification experiments from 9 to 16

Texture Classes		% error rates in diagonal wavelet detail coefficients				% error rates			
		CE 9	CE 10	CE 11	CE 12	CE 13	CE 14	CE 15	CE 16
1	Flowers 3	9,38	4,69	9,38	9,38	6,25	6,25	4,69	6,25
2	Brick 2	60,94	46,88	45,31	50	12,5	6,25	42,19	12,5
3	Fabric 8	35,94	57,81	34,38	23,44	1,56	7,81	9,38	1,56
4	Metal 3	21,88	46,88	25	23,44	0	0	17,19	0
5	Food 5	73,44	60,94	65,63	65,63	10,94	15,63	29,69	12,5
6	Grass 1	21,88	28,13	23,44	23,44	14,06	17,19	9,38	15,63
7	Sand 5	6,25	9,38	7,81	7,81	7,81	7,81	6,25	4,69
8	Leaves 3	15,63	42,19	12,5	12,5	6,25	9,38	9,38	4,69
Means		30,6675	37,1125	27,9313	26,955	7,42125	8,79	16,0188	7,2275

Table 6.4: Error rates of classification experiment 17

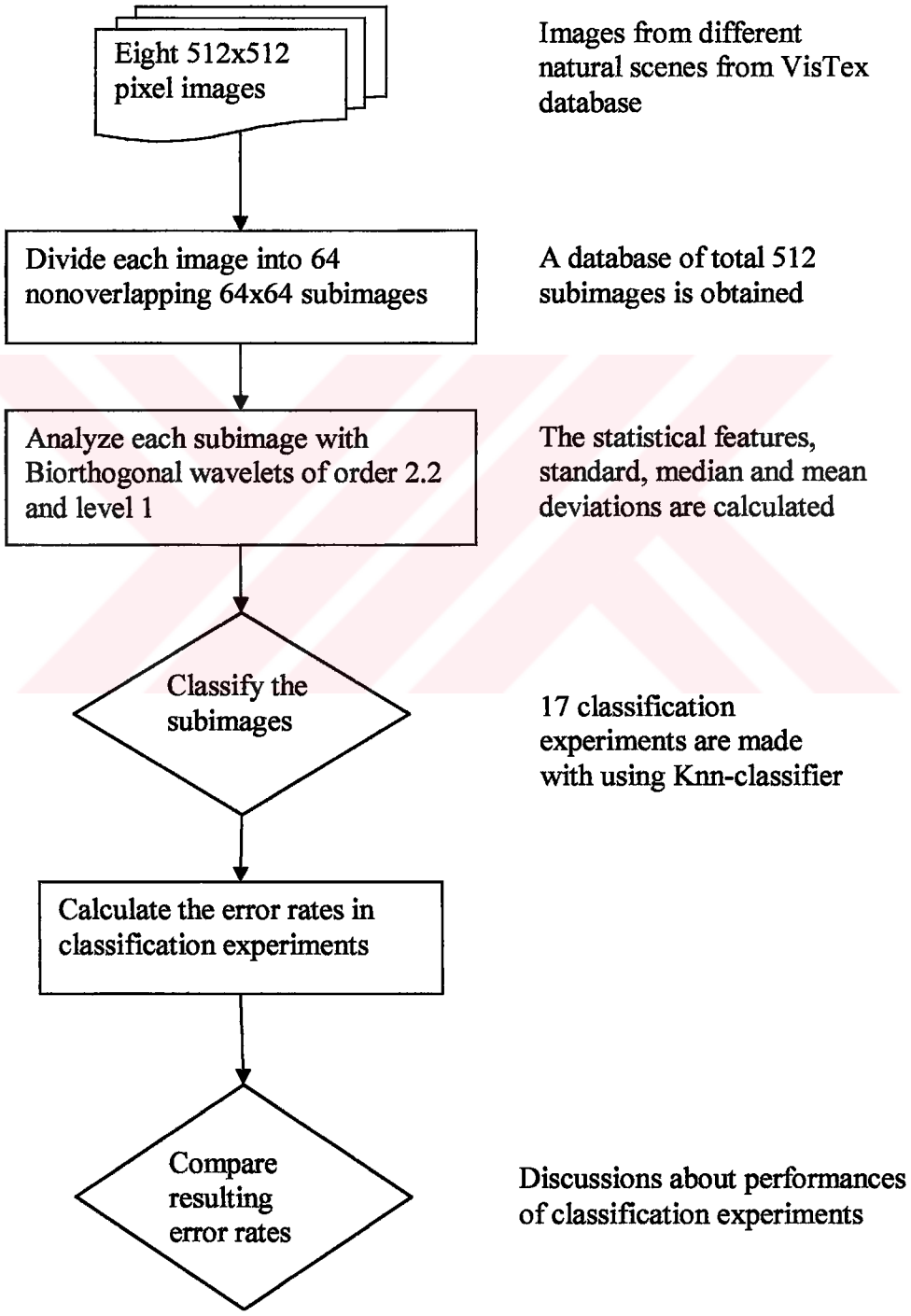
Texture Classes		% error rates
		CE 17
1	Flowers 3	1,56
2	Fabric 8	1,56
3	Metal 3	0
4	Grass 1	6,25
5	Sand 5	4,69
6	Leaves 3	4,69
Mean		3,125

REFERENCES

1. Wouwer, G.V. (1998). *Wavelets for multiscale texture analysis*. Ph. D. Thesis, Department of Physics, University of Antwerp, Groenenborgerlaan, Belgium.
2. Drimbarean, A., Whelan, P.F. (2001, February 9). Experiments in color texture analysis. *Elsevier, Pattern Recognition Letters*, **22**, 1161-1167.
3. Unser, M. (1986). Local linear transforms for texture measurements. *Elsevier, Signal Processing*, **11(1)**, 61-79.
4. Jain, A.K., Farrokhnia, F. (1991). Unsupervised texture segmentation using Gabor filters. *Elsevier, Pattern Recognition*, **24(12)**, 1167-1186.
5. Haralick, R. (1979, May). Statistical and structural approaches to textures. *Proceedings IEEE*, **67(5)**, pp. 786-804.
6. Misiti, M., Misiti, Y., Oppenheim, G., Poggi, J.M. (2001, June). *Wavelet toolbox user's guide for use with Matlab*. The MathWorks. Inc. http://www.mathworks.com
7. Polikar, R. (1996). *The Wavelet tutorial*. Iowa State University, Ames, U.S.A. http://www.public.iastate.edu/~rpolikar/homepage.html
8. Graps, A. (1995). An introduction to wavelets. *IEEE Computational Science and Engineering*, **2(2)**, 1-18.
9. Grossman, A., Morlet, J. (1984). Decompositions of hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.* **15**, 723-736.
10. Daubechies, I. (1994). *Ten lectures on wavelets*. SIAM, Capital City Press, Vermont.
11. Munoz, A., Ertle, R., Unser, M. (2002). Continuous wavelet transform with arbitrary scales and O(N) complexity. *Elsevier, Signal Processing*, **82**, 749-757.
12. Cohen, A. (1992). *Ondelettes, analyses multiresolution et traitement numerique du signal*. Ph. D. Thesis, University of Paris, IX Dauphine.
13. Beck, J., Sutter, A., Ivry, R. (1987). Spatial frequency channels and perceptual grouping in texture segmentation. *Comp. Vis. Graph. Im. Process*, **37**, 299-325.
14. Jain, A.K. (1996). Learning texture discrimination masks. *IEEE Trans. Patt. Anal. Mach. Intell.* **18(2)**, 195-205.
15. Mallat, S.G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Patt. Anal. Mach. Intell.* **11(7)**, 674-693.
16. Vapnik, V. (1982). *Estimation of dependences based on empirical data*. Springer-Verlag, New York.
17. VisTex, 2000. Color Image Database, http://www.white.media.mit.edu/vismod/imagery/VisionTexture
18. Kara, B. (2003). Using wavelets for texture classification. *International Conference on Signal Processing 2003*, Çanakkale.

APPENDIX A

FLOWCHART OF CLASSIFICATION EXPERIMENTS



APPENDIX B

COMPUTER PROGRAMS

B.1 A Matlab Program for Fourier and Wavelet Transforms

```
t=linspace(0,0.5,500);
u=cos(2*pi*10*t)+cos(2*pi*25*t)+cos(2*pi*50*t)+cos(2*pi*100*t);
plot((1000*t),u)
title('original signal')
grid
y=fft(u);
plot((2000*t),y)
grid
title('fourier transform of original signal')
m=abs(y);
p=unwrap(angle(y));
f=(0:length(y)-1)*999/length(y);
plot(f,m)
grid
title('Magnitude of fourier transform of original signal')
set(gca,'Xtick',[10 25 50 100])
plot(f,p*180/pi)
title('Phase of fourier transform of original signal')
set(gca,'XTick',[10 25 50 100])
grid
c = cwt(u,1:32,'db2','3dabsglb');
colorbar
grid
```

B.2 A Matlab Program for Converting a RGB Image to Indexed Image for Analysing with Matlab

```
RGB=imread('D:\Matlab\bark0\Bark.0000.jpg');
RGB1=im2uint16(RGB);
imshow(RGB1);
whos
[X,map]=rgb2ind(RGB,257);
X=rgb2ind(RGB,map);
imshow(X,map)
whos
save bark0 X map
```

B.3 A Matlab Program for Classification Experiments

```
% Horizontal, Vertical and Diagonal Wavelet Detail Coefficients
% Biorthogonal Wavelets of order 2.2 and level 1.
% CE 16:Standart,Median and Mean Deviation of
%          Horizontal,Vertical and Diagonal coefficients
 P=[11.5448 05.0498 07.8930 10.0635 04.3448 06.8047 05.7403 02.6748 03.9885;
     28.1108 14.2376 20.3328 16.7112 07.9861 11.6864 09.2173 04.3282 06.4087;
     45.8091 21.8300 32.3936 26.4753 13.2582 18.9125 19.5167 09.1193 13.5984;
     42.2747 25.5761 32.5008 46.4850 27.5641 35.4414 23.2411 13.7891 17.6969;
     51.3463 32.4970 40.1292 41.6155 25.2483 32.0145 21.3098 12.8008 16.3052;
     69.5694 42.8612 53.7166 38.0166 22.3220 28.8453 28.4967 16.1620 21.3553;
     85.7339 56.4481 68.3200 17.2114 09.8997 12.9867 12.3294 07.2803 09.3592;
     96.5916 66.9753 77.7769 20.0733 12.1519 15.4025 14.1000 08.6670 10.8785]';
%                                   Error rates
% 1: flowers3  : True_60 : False_04 : 06,25
% 2: brick2    : True_56 : False_08 : 12,50
% 3: fabric8   : True_63 : False_01 : 01,56
% 4: metal3    : True_64 : False_00 : 0
% 5: food5     : True_56 : False_08 : 12,50
% 6: grass1    : True_54 : False_10 : 15,63
% 7: sand5     : True_61 : False_03 : 04,69
% 8: leaves3   : True_61 : False_03 : 04,69
% total              475 :         37 : 07,23
PP1=[P1h;P1v;P1d];PP2=[P2h;P2v;P2d];PP3=[P3h;P3v;P3d];PP4=[P4h;P4v;P4d];
PP5=[P5h;P5v;P5d];PP6=[P6h;P6v;P6d];PP7=[P7h;P7v;P7d];PP8=[P8h;P8v;P8d];
True=0;
False=0;
for i=1:1:64
   for j=1:1:8
[a(1,j)]=sqrt((PP8(1,i)-P(1,j))^2+(PP8(2,i)-P(2,j))^2+(PP8(3,i)-P(3,j))^2+(PP8(4,i)-
P(4,j))^2+(PP8(5,i)-P(5,j))^2+(PP8(6,i)-P(6,j))^2+(PP8(7,i)-P(7,j))^2+(PP8(8,i)-
P(8,j))^2+(PP8(9,i)-P(9,j))^2);
end
m=min(a);
for k=1:1:8
if m==a(1,k);
  [b8(1,i)]= k;
  if k==8;
     True=True+1;
  else
     False=False+1;
  end
end
end
end
b8
True
False
ErrorRate=(False/64)*100
```