

1-1-2005

# Optimal filter bank representation in texture classification using support vector machines

Samsher Singh Sidhu  
*Ryerson University*

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Electrical and Computer Engineering Commons](#)

---

## Recommended Citation

Sidhu, Samsher Singh, "Optimal filter bank representation in texture classification using support vector machines" (2005). *Theses and dissertations*. Paper 393.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact [bcameron@ryerson.ca](mailto:bcameron@ryerson.ca).

# OPTIMAL FILTER BANK REPRESENTATION IN TEXTURE CLASSIFICATION USING SUPPORT VECTOR MACHINES

SAMSHER SINGH SIDHU  
B.Eng, Ryerson University, 2003.

A thesis submitted in partial fulfillment  
of the requirements for the degree of

MASTER OF APPLIED SCIENCE

Department of Electrical and Computer Engineering  
RYERSON UNIVERSITY  
Toronto, Ontario, Canada

May, 2005

©Samsheer Singh Sidhu 2005

UMI Number: EC53764

#### INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.



---

UMI Microform EC53764  
Copyright 2009 by ProQuest LLC  
All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.

---

ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

RYERSON UNIVERSITY  
DEPARTMENT OF ELECTRICAL  
AND COMPUTER ENGINEERING

The oral exam committee members certify that they have read and recommended to the School of Graduate Studies for acceptance a thesis entitled "**Optimal Filter Bank Representation in Texture Classification using Support Vector Machines**" by **Samsher Singh Sidhu** in partial fulfillment of the requirements for the degree of **Master of Applied Science**.

Dated: May 30, 2005

Committee Chair or External Examiner:

Dr. Fei Yuan

Research Supervisor:

Dr. Kaamran Raahemifar

Examining Committee:

Dr. Javad Alirezaie

Dr. Gul Khan

# RYERSON UNIVERSITY

Date: May 2005

Author: Samsheer Singh Sidhu

Title: Optimal Filter Bank Representation in Texture Classification using Support Vector Machines

Department: Department of Electrical and Computer Engineering


Degree: M.A.Sc.

Convocation: May

Year: 2005

I hereby declare that I am the sole author of this thesis.

Permission is herewith granted to Ryerson University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

 Signature of Author

THE AUTHOR AND THE SUPERVISOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

## Borrower's Page

RYERSON UNIVERSITY requires the signatures of all persons using or photocopying this thesis. Please fill in the required cells of the table: Name, Signature, Address and the Date.

[illegible]

*To Almighty Lord and Master of the Universe, who imparts  
understanding and inspires us to speak.*

## Acknowledgment

I would like to thank Dr. Kaamran Raahemifar, my supervisor, for his many suggestions and constant support during this research. I would also like to thank Dr. Sridhar Krishnan and Dr. Javad Alirezaie for introducing me to the field of signal and image processing and for helpful discussions during the course of my studies at RYERSON UNIVERSITY.

I would like to thank the committee members, the Chair, Dr. J. Alirezaie and Dr. G. Khan for taking time out of their busy schedules to review the thesis and for their valuable comments.

I am also thankful to Dr. Mike Kassam for his guidance throughout the years of chaos and confusion during my tenure at RYERSON. I would also express gratitude to the front office staff: Mary, Yvonne, Marta and Dawn; for supporting my undergraduate and graduate studies at RYERSON UNIVERSITY.

Of course, I am grateful to my parents, Swarn & Gurmeet, and my two sisters, Jaspreet & Jasmeet, for their *patience* and *love*.

Toronto, Ontario, Canada  
April 2, 2005

Samsher Singh Sidhu



# Abstract

## Optimal Filter Bank Representation in Texture Classification using Support Vector Machines

©Samsher Singh Sidhu 2005

Master of Applied Science  
Department of Electrical and Computer Engineering  
RYERSON UNIVERSITY

Texture analysis has been a field of study for over three decades in many fields including Electrical Engineering. Today, texture analysis plays a crucial role in many tasks ranging from remote sensing to medical imaging. Researchers in this field have dealt with many different approaches, all trying to achieve the goal of a high classification accuracy.

The main difficulty of texture analysis was the lack of ability of the tools to adequately characterize different scales of the textures effectively. The development in multi-resolution analysis such as Gabor and Wavelet Transform help to overcome this difficulty. This thesis describes the texture classification algorithm that uses the combination of Statistical features and Co-Occurrence features of the Discrete Wavelet Transformed images. The classification accuracy is increased by using translation-invariant features generated from the Discrete Wavelet Frame Transform.

The results are further improved by focussing on the transformed images used for feature extraction by using filters which essentially extract those areas of the image that discriminate themselves from other image classes. In effect, by reducing the spatial characteristics of images that contribute to the features, the texture classification method still has the ability to preserve the classification accuracy.

Support Vector Machines has provided excellent performance in the area of pattern recognition problems. We have applied SVMs with the texture classification method described above and when compared to traditional classifiers, SVM has produced more accurate classification results on the *Brodatz* texture album.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Wavelet Transform</b>	<b>5</b>
2.1	History on Wavelets . . . . .	5
2.2	Wavelet Applications . . . . .	5
2.3	Fourier Analysis . . . . .	6
2.4	Short-Time Fourier Analysis . . . . .	8
2.5	Wavelet Analysis . . . . .	9
2.5.1	Continuous Wavelet Transform . . . . .	10
2.5.2	Discrete Wavelet Transform . . . . .	12
2.5.3	Wavelet Packet Analysis . . . . .	18
2.5.4	Discrete Wavelet Frame Transform . . . . .	19
2.6	Summary . . . . .	20
<b>3</b>	<b>Support Vector Machines</b>	<b>22</b>
3.1	Linear Support Vector Machine Classifier . . . . .	23
3.2	Non-Linear Support Vector Machine Classifier . . . . .	26
3.3	SVM Applications . . . . .	31
3.4	Summary . . . . .	32
<b>4</b>	<b>Optimal Filter Bank Representation</b>	<b>33</b>
4.1	Introduction . . . . .	33
4.2	Non-Filtering . . . . .	34
4.2.1	Statistics . . . . .	34
4.2.2	Model-Based . . . . .	35
4.3	Fixed Filters . . . . .	35

4.3.1	Laws Filter Masks . . . . .	35
4.3.2	Discrete Cosine Transform . . . . .	36
4.3.3	Gabor Filter . . . . .	36
4.3.4	Wavelet Transforms, Packets, and Frames . . . . .	36
4.4	Applications . . . . .	37
4.5	Related Work . . . . .	38
4.6	Motivation . . . . .	40
4.7	Background . . . . .	42
4.8	Methodology . . . . .	43
4.9	Summary . . . . .	47
<b>5</b>	<b>Texture Classification: Comparative Experiments</b>	<b>48</b>
5.1	Experimental Setup . . . . .	48
5.1.1	Data . . . . .	49
5.1.2	Extracted Features . . . . .	50
5.1.3	Classification . . . . .	53
5.2	Results . . . . .	54
5.3	Multi-Texture Classification . . . . .	64
5.4	Summary . . . . .	66
<b>6</b>	<b>Conclusions</b>	<b>67</b>
6.1	Contributions of This Work . . . . .	67
6.2	Conclusions of This Work . . . . .	69
6.3	Future Research . . . . .	69
	<b>Bibliography</b>	<b>72</b>
	<b>A Co-Occurrence Matrix Equations</b>	<b>79</b>
	<b>B Brodatz Image Collection</b>	<b>81</b>
	<b>C Filter Shapes</b>	<b>89</b>

# List of Figures

2.1	An illustration of a signal in the time domain. . . . .	7
2.2	An illustration of a signal in the frequency domain. . . . .	7
2.3	A sampled window of the signal ready to be transformed. . . .	8
2.4	The resulting mapping function that occurs using STFT. . . .	8
2.5	Time domain mapping as prescribed by Shannon. . . . .	9
2.6	Frequency Domain mapping as prescribed by Fourier. . . . .	9
2.7	STFT mapping as prescribed by Gabor. . . . .	10
2.8	Wavelet Analysis mapping as prescribed by Daubechies. . . .	10
2.9	Top: Signal Under Study. Bottom: Low Scaled Wavelet. . . .	12
2.10	Top: Signal Under Study. Bottom: High Scaled Wavelet. . . .	12
2.11	Filtering process at the most basic level. . . . .	13
2.12	The pure Sinusoidal Wave. . . . .	14
2.13	The High frequency content random noise. . . . .	14
2.14	The sinusoidal wave added with high frequency random noise.	14
2.15	Level One Approximation Coefficients. . . . .	15
2.16	Level One Detail Coefficients. . . . .	15
2.17	Multi-Level DWT Decomposition. . . . .	15
2.18	Level Two Approximation Coefficients. . . . .	16
2.19	Level Two Detail Coefficients. . . . .	16
2.20	Level Three Approximation Coefficients. . . . .	16
2.21	Level Three Detail Coefficients. . . . .	17
2.22	Wavelet Packet Decomposition Tree. . . . .	18
2.23	Image Decomposition for DWT and DWFT. . . . .	20
2.24	One Stage 2-D DWT and IDWT . . . . .	21
2.25	One Stage 2-D DWFT and IDWFT . . . . .	21
3.1	Sample Linear Hyperplane. . . . .	23

3.2	Various Hyperplanes in existence. . . . .	24
3.3	Optimal Hyperplane. . . . .	24
3.4	Feature Mapping. . . . .	26
3.5	Separation Margin. . . . .	30
4.1	D1, Brodatz image collection. . . . .	44
4.2	The <i>Cross</i> Filter. . . . .	44
4.3	D1 intersected with <i>Cross</i> Filter. . . . .	45
5.1	Texture Training. . . . .	50
5.2	Texture Testing or Classification. . . . .	51
5.3	The complete texture classification implementation. . . . .	55
5.4	Triangle. Black represents pixels of interest. 50% Active. . . .	60
5.5	Triangle. Black represents pixels of interest. 50% Active. . . .	60
5.6	Inverted Diamond. 48% Active. . . . .	61
5.7	Diamond. 52% Active. . . . .	61
5.8	Circle. 71% Active. . . . .	61
5.9	Inverted Circle. 41% Active. . . . .	61
5.10	Hour Glass. 50% Active . . . . .	62
5.11	Inverted Hour Glass. 50% Active . . . . .	62
5.12	Cross Template. 55% Active. . . . .	62
5.13	Dual Triangle. 77% Active. . . . .	62
5.14	Checkerboard. 50% Active. . . . .	62
5.15	Vertical Lines. 50% Active. . . . .	62
5.16	Horizontal Lines. 50% Active. . . . .	63
5.17	2-class multi-texture image. . . . .	64
5.18	Misclassification Error Rate of 7.80%. . . . .	64
5.19	4-class multi-texture image. . . . .	65
5.20	Misclassification Error Rate of 12.44%. . . . .	65
5.21	8-class multi-texture image. . . . .	65
5.22	Misclassification Error Rate of 16.71%. . . . .	66
B.1	D1. . . . .	81
B.2	D3. . . . .	81
B.3	D6. . . . .	82

B.4 D11. . . . .	82
B.5 D16. . . . .	82
B.6 D17. . . . .	82
B.7 D20. . . . .	83
B.8 D21. . . . .	83
B.9 D24. . . . .	83
B.10 D28. . . . .	83
B.11 D29. . . . .	84
B.12 D32. . . . .	84
B.13 D34. . . . .	84
B.14 D35. . . . .	84
B.15 D46. . . . .	85
B.16 D47. . . . .	85
B.17 D49. . . . .	85
B.18 D51. . . . .	85
B.19 D52. . . . .	86
B.20 D53. . . . .	86
B.21 D55. . . . .	86
B.22 D56. . . . .	86
B.23 D57. . . . .	87
B.24 D65. . . . .	87
B.25 D78. . . . .	87
B.26 D82. . . . .	87
B.27 D84. . . . .	88
B.28 D85. . . . .	88
B.29 D101. . . . .	88
B.30 D104. . . . .	88
C.1 Cross Template. 55% Active. . . . .	89
C.2 Triangle. Black represents pixels of interest. 50% Active. . . .	90
C.3 Inverted Triangle. 50% Active. . . . .	90
C.4 Diamond. 52% Active. . . . .	91
C.5 Inverted Diamond. 48% Active. . . . .	91
C.6 Box Template. 65% Active. . . . .	92

C.7 Inverted Box Template. 80% Active. . . . .	92
C.8 Circle. 71% Active. . . . .	93
C.9 Inverted Circle. 41% Active. . . . .	93
C.10 Half Triangle. 50% Active. . . . .	94
C.11 Other-Half Triangle. 50% Active. . . . .	94
C.12 Hour Glass. 50% Active . . . . .	95
C.13 Inverted Hour Glass. 50% Active . . . . .	95
C.14 Horizontal Lines. 50% Active. . . . .	95
C.15 Vertical Lines. 50% Active. . . . .	96
C.16 Checkerboard. 50% Active. . . . .	96
C.17 Dual Hour Glass. 42% Active. . . . .	97
C.18 Inverted Dual Hour Glass. 58% Active. . . . .	97
C.19 Dual Triangle. 77% Active. . . . .	98

# List of Tables

3.1	Common Kernel Functions. . . . .	31
5.1	Classification of Three Decomposition Levels. . . . .	56
5.2	Classification of Four Decomposition Levels. . . . .	56
5.3	Classification of Five Decomposition Levels. . . . .	57
5.4	Different Classifier Comparison. . . . .	57
5.5	Different Classifier Comparison. . . . .	57
5.6	Different Classifier Comparison. . . . .	58
5.7	DWT versus extended DWFT. . . . .	59
5.8	Triangle Contrasted Filters. . . . .	60
5.9	Diamond Contrasted Filters. . . . .	61
5.10	Circle Contrasted Filters. . . . .	61
5.11	Hour-Glass Contrasted Filters. . . . .	62
5.12	Correct Classification Accuracy of the optimum filters. . . . .	63
C.1	Filter Cross Template. . . . .	89
C.2	Filter Triangle. . . . .	90
C.3	Filter Inverted Triangle. . . . .	90
C.4	Filter Diamond. . . . .	91
C.5	Filter Inverted Diamond. . . . .	91
C.6	Filter Box Template. . . . .	92
C.7	Filter Inverted Box Template. . . . .	92
C.8	Filter Circle. . . . .	93
C.9	Filter Inverted Circle. . . . .	93
C.10	Filter Dual Triangle. . . . .	94
C.11	Filter Other-Half Triangle. . . . .	94
C.12	Filter Hour Glass. . . . .	95



C.13 Filter Inverted Hour Glass. . . . .	95
C.14 Filter Horizontal Lines. . . . .	95
C.15 Filter Vertical Lines. . . . .	96
C.16 Filter Checkerboard. . . . .	96
C.17 Filter Inverted Dual Hour Glass. . . . .	97
C.18 Filter Dual Hour Glass. . . . .	97
C.19 Filter Dual Triangle. . . . .	98

# Chapter 1

## Introduction

TWO of the most problems studied in the context of signal and image processing of Texture Classification are Feature Extraction and Classification. Feature Extraction is the process in which signal coordinates are mapped onto another set of coordinates in a way that signal discrimination or signal storage is optimized through a transformation, linear or nonlinear. Essentially, Feature Extraction is based on disposal of less important (irrelevant) information in the signal by identifying the most relevant features in a signal. Classification has been a domain that has existed for many centuries in almost every field of science and arts. Due to this fact, it has led to a tremendous amount of knowledge about different machine recognition schemes, that requires either or both requirements of high accuracy or speed for applications.

This is very important when considering the increasing popularity of digital libraries and multimedia databases, where content-based image retrieval is becoming an important research topic [1]. A very useful characterization for a wide range of images is by their textures. Moreover, it is generally believed that even our human visual system uses textures for recognition and interpretation. Textures have been successfully used for object recognition, scene interpretation and segmentation, especially in the areas of industrial automation, remote sensing and medical diagnosis. As such, numerous techniques have been developed for texture classification, segmentation, synthesis and other related tasks.

This thesis discusses these different texture analysis methods with the use of different

feature extraction and classification protocols.

## Feature Extraction

Large number of texture features have been proposed [2, 3]. They can be categorized into four major methods, namely:

- Statistical - the co-occurrence method [4].
- Geometrical - Voronoi Tessellation Features [5].
- Model-Based - Markov Random Fields [6] and Fractals [7, 8].
- Signal Processing - Gabor Filters [9], Wavelet Transforms [10, 11] and QMFs [12].

Few empirical comparisons on these textures have been conducted with some mixed results [13, 14]. However, Randen et al. [12] performed a more systematic and comprehensive study, with particular emphasis on the signal processing approach. It was concluded that no texture feature is consistently superior for *all* images, though the wavelet frame with quadrature mirror filters are among the best for most images.

The evidence of good performance in texture classification is due to the fact that Wavelet Theory has a solid foundation based on formal mathematical theory for multi-resolution image analysis [11, 15, 16]. However, the main problem with the commonly used discrete wavelet transform (DWT) is that it is translation-variant. A simple integer shift of the signal/image would result in non-trivial modifications of the wavelet coefficients. This is problematic in texture analysis as textures are generally considered as translation-invariant. Unser [17] implemented the discrete wavelet frame transform (DWFT), which is an over-complete wavelet representation, to alleviate this problem. As it will be discussed in Section 2.5.4, DWFT avoids the down-sampling operations in DWT and, thus, yields a description that is translation-invariant.

## Classification

Another crucial component in texture classification is the classifier being used after feature extraction. A number of classifiers has been used from the most common Bayes classifier [12, 18, 19], to nearest neighbor classifiers [20], linear discriminants [21], neural networks [10], and learning vector quantization classifiers [12].

In the past decade, the Support Vector Machine (SVM) Classifier [22, 23] has proved to be a promising pattern recognition method and in achieving successful results. It has outperformed traditional techniques in various applications such as handwritten digit recognition [24], text categorization [25] and spam categorization [26]. SVM has also provided superior performance in regression [27] and time series prediction [28].

The reason for SVM's success is that it is based on *Structural Risk Minimization* [22], rather than empirical risk minimization used for many other methods. SVM attempts to minimize the generalization error, which is a combination of Empirical Risk and Structural Risk. Other methods simply attempt to minimize the training error, which carries many flaws such as *over-fitting* [2].

In this thesis, both feature extraction and classification are studied and a method has been proposed in terms of texture analysis and classification. Several researchers have proposed many different feature extraction methods and many have been applied in many different areas of study. This is also true with the study of classification. We have proposed a method by effectively applying SVMs and employing external shift-invariant texture features generated by DWFT to the problem of texture classification.

## Summary of Contributions

The contributions of this thesis to the original schemes are:

- Implementing a texture classification approach that operates on small windows instead of global ones from the texture classes and utilizing the complete separation of training and testing sets. These elements are used to provide an unbiased and un-boosted final classification accuracy.
- Extending the feature extraction algorithm by increasing the feature space of the feature vector by implementing the Discrete Wavelet Frame Transform, which incorporates the shift-invariant representation.
- Proposing an extension to the texture classification method by optimally representing the filter banks in the DWFT for better discriminatory features.

## Organization of the Thesis

The thesis consists of seven chapters, which are organized as follows:

In Chapter 2, the background information on the ground breaking Wavelet Transform, and how it is applied to the Texture Classification method is discussed.

In Chapter 3, SVM theory is reviewed and explained, and how it is a vital component in the texture classification methodology.

In Chapter 4, existing technology of texture analysis and features are reviewed. Our proposed method of optimal representation for feature extraction of the Wavelet Transform's decomposed filter banks is introduced and discussed.

In Chapter 5, comparative experiments are carried out. The different decompositions of wavelet transform and the extension method with the proposed filtering methods are compared.

In Chapter 6, conclusions and future work of this thesis are stated.

# Chapter 2

## Wavelet Transform

**E**VERYWHERE around us are signals that can be analyzed. For example there are: seismic tremors, human speech, engine vibrations, medical signals and images, financial data, music, and many other types of signals. Wavelet Analysis is a promising set of tools and techniques for analyzing these signals.

In this thesis, we have made use of the collection of built-in functions provided under the Wavelet Toolbox in the MATLAB® Technical Computing Environment. It provides tools for the analysis and synthesis of signals and images using wavelets and wavelet packets all within MATLAB. All further simulations are also completed using this platform.

### 2.1 History on Wavelets

The main algorithm dates back to the work of Stephane Mallat in 1988 [11, 15]. Since then, research on wavelets has become an international study. 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.

### 2.2 Wavelet Applications

The reason for which wavelets and its corresponding decompositions are used is its ability to combine scale and time aspects that are present in any application. Wavelets are mathematical functions that separate data into different frequency components. Each com-

ponent is further studied depending on the resolution of the scale. They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. This is discussed in Section 2.3. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering, and seismic geology. Interchanges between these fields over the past decade has led to many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction.

Many of the applications that apply the fundamentals of wavelets attempt to achieve the goal of signal or image clearance and simplification, which are essentially de-noising and compression. There are many published papers covering many different areas, it is almost impossible to sum up several thousand papers written within the last 15 years. However, the list below gives the reader an insight on the long reach of wavelets. Some of the domains are:

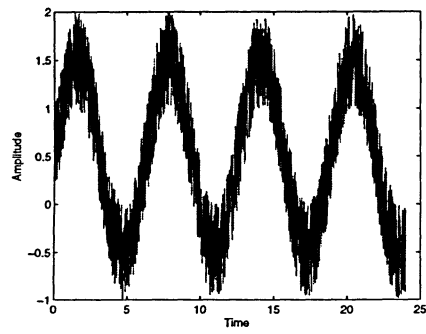
- Oceanography and Earth Studies [29].
- Compression of FBI Fingerprints [30, 31].
- ECG and EKG extraction for localization of heart activity and noise removal [32].
- Enhancing Mammograms to discriminate tumors from calcification [33].

If reader is interested in the area of wavelets and its applications, the reader is encouraged to read [34] to get a broad understanding of this field.

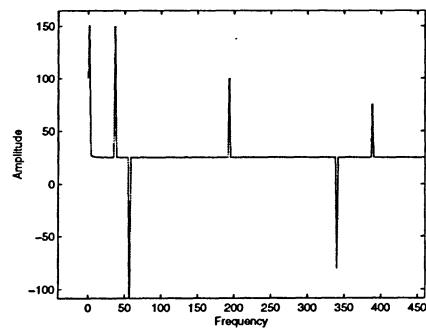
## 2.3 Fourier Analysis

The most well known of signal analysis is the Fourier analysis or *Fourier Transform*. This analysis tool breaks down the signal into constituent sinusoids of different frequencies. Essentially, it converts or *transforms* the view of the signal from a time-based to frequency-based.

Fourier Transform is very useful when the signal's frequency content is of importance. However, when the signal is transformed into the frequency domain, all time information is



**Figure 2.1:** An illustration of a signal in the time domain.



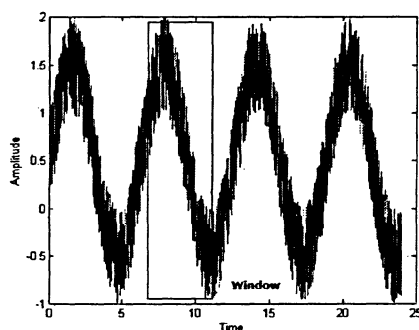
**Figure 2.2:** An illustration of a signal in the frequency domain.



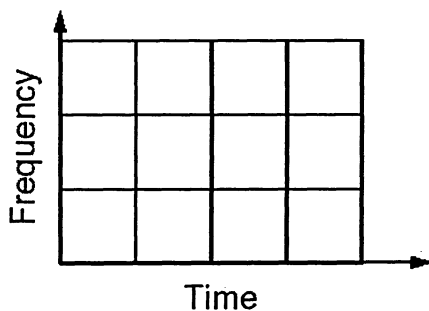
lost. When analyzing the transformed signal, it is impossible to dictate when a particular event took place.

## 2.4 Short-Time Fourier Analysis

Gabor adapted the Fourier Transform to analyze the signal in small sections at a time to overcome the deficiency of the original transform. In doing so, it essentially maps the signal into two functions of time and frequency. This is now known as *Short-Time Fourier Transform* or *STFT*.



**Figure 2.3:** A sampled window of the signal ready to be transformed.



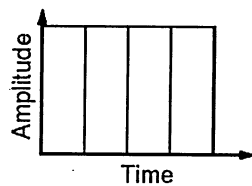
**Figure 2.4:** The resulting mapping function that occurs using STFT.

As can be seen by Figure 2.4, it merges the view points of analyzing a signal, time and frequency. It provides the ability to localize the information that has lacked the past efforts of signal analysis. However, this method suffers the effects of the size of the window

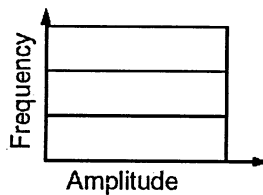
chosen for analysis. If the window is too small, low frequency signals fail to get analyzed. If the window is too big, high frequency signals fail to get analyzed. Many signals that are experienced need a flexible approach, where the window can be more adaptive to capture either the time or frequency information.

## 2.5 Wavelet Analysis

With the impressive set of tools already been developed, wavelet analysis represented the next step in signal analysis. It implements the adaptive approach to analyzing signals, more specifically a *time-scale* view of the signal. The *window* is expanded when we want more precise low-frequency information, and is shortened in those regions where we want high-frequency information. From Figure 2.5 to Figure 2.8 shows this progression.

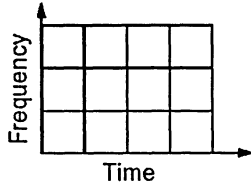


**Figure 2.5:** Time domain mapping as prescribed by Shannon.

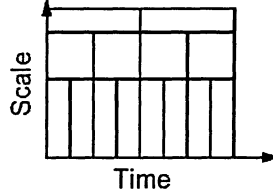


**Figure 2.6:** Frequency Domain mapping as prescribed by Fourier.

Because of wavelet's ability to be adaptive in analyzing signals, it does not use the time-frequency approach, as used by others, but a *time-scale* approach. The window is directly related to the scaling and shifting of wavelets. *Scaling* a wavelet simply means stretching (or compressing) it. It works in the same manner a scale factor would affect a sinusoid wave. The smaller the scale factor, the more *compressed* the wavelet. As with sinusoid wave,



**Figure 2.7:** STFT mapping as prescribed by Gabor.



**Figure 2.8:** Wavelet Analysis mapping as prescribed by Daubechies.

*Shifting* simply means delaying (or hastening) its onset. The mathematics are the same for both cases. Delaying a function  $f(t)$  by  $k$  is represented by  $f(t - k)$  in a traditional sinusoid case. In the wavelet family, the original function is  $\Psi(t)$  and the shifted function is  $\Psi(t - k)$ .

### 2.5.1 Continuous Wavelet Transform

This leads to the derivation of the *Continuous Wavelet Transform*. Recalling fourier analysis, Figure 2.6 and Section 2.3, the transform is the sum over all time of the signal  $f(t)$  multiplied by the complex exponential, which can be broken down into complex and imaginary sinusoidal components as shown in Equation 2.1.

$$F(w) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt. \quad (2.1)$$

The result of this transform are the *Fourier Coefficients*  $F(w)$ . Similarly, the *Continuous Wavelet Transform (CWT)* is defined as the sum over all time of the signal multiplied by the scaled and shifted versions of the wavelet function:

$$C(scale, position) = \int_{-\infty}^{+\infty} f(t)\Psi(scale, position, t)dt, \quad (2.2)$$

where the wavelet transform is represented by any arbitrary function as a superposition of wavelets, which are functions generated from a mother wavelet,  $\Psi$ , by dilations and translations of

$$\Psi^{a,b}(x) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right). \quad (2.3)$$

The result of this transform are many wavelet coefficients, which are a function of the mother wavelet,  $\Psi$ . If one were to multiply each coefficient with the appropriate scaled and shifted wavelet, it would yield the constituents signals of the original signal, just as Fourier.

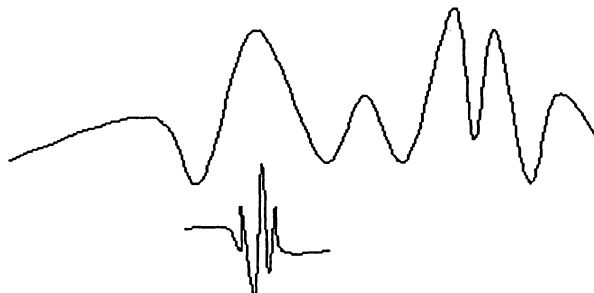
### Scale versus Frequency

As stated earlier, the windowing function adapts itself to signal under study. This has a direct effect on the wavelet used. Higher scales (window) correspond to the most stretched wavelets. The more stretched the wavelet, the longer the portion of the signal with which it is being compared, and thus, the coarser the signal features is being measured by the wavelet coefficients. The resultant coefficient  $C$  represents how closely correlated the wavelet is with the section of the signal. The coefficient  $C$  can be interpreted as a correlation coefficient, if the signal and wavelet are normalized to equal unity energy [34]. Therefore, through our analysis the relationship between wavelet scales and frequency becomes:

- *Low Scale*  $\Rightarrow$  Compressed Wavelet  $\Rightarrow$  Rapidly changing details  $\Rightarrow$  High Frequency of Signal under study.
- *High Scale*  $\Rightarrow$  Stretched Wavelet  $\Rightarrow$  Slowly changing features  $\Rightarrow$  Low frequency of Signal under study.

Figure 2.9 and 2.10 shows this relationship.

With the many strengths of this transform, almost all signal processing are performed on the computer using the real-world data that has been changed into discrete form. If one were to use CWT, it would operate at every scale, shifted over the full domain of the original signal to predefined maximum scale, which is the trade-off for the need for detailed analysis with computational power.



**Figure 2.9:** Top: Signal Under Study. Bottom: Low Scaled Wavelet.



**Figure 2.10:** Top: Signal Under Study. Bottom: High Scaled Wavelet.

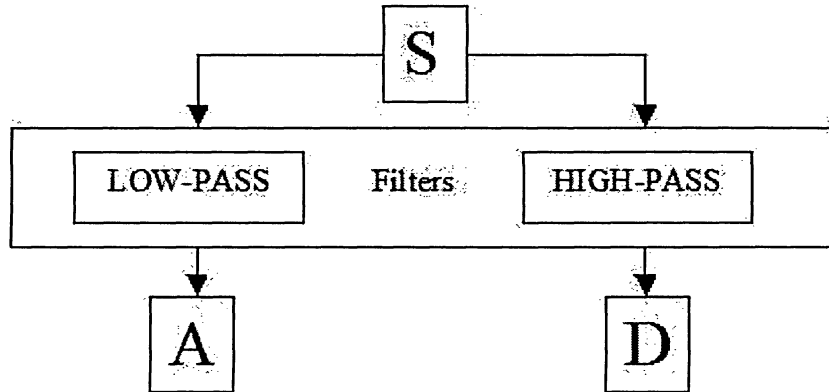
## 2.5.2 Discrete Wavelet Transform

The standard description of Discrete Wavelet Transform (DWT) has been a development over many years with the work of many researchers. It solves the problem of calculating wavelet coefficients at every scale, by using scales and positions based on *dyadic* (powers of two). This produces a more efficient algorithm, while still maintaining its accuracy. Mallat [11, 15] introduced this concepts as a two-channel subband coder. It essentially is a *fast wavelet transform*, when compared to the CWT.

DWT is built upon the simple fact that low-frequency content is the most important part for many signals. The remaining high frequency content simply adds flavor or noise [11]. The perfect example is human speech. By removing the high-frequency elements from the speech sample, the speech may change. However, the information or *what is being said*, can still be determined. If too much of the high-frequency content is removed, the speech may become distorted and sound irrelevant [34].

Wavelet Analysis uses *approximation* and *detailed* terms to describe frequency informa-

tion. Approximation (A) is the term to describe the low frequency information, while the Detail (D) describes the high frequency information. This is achieved by filtering the original signal,  $S$  through two complimentary high and low pass filters shown in Figure 2.11.



**Figure 2.11:** Filtering process at the most basic level.

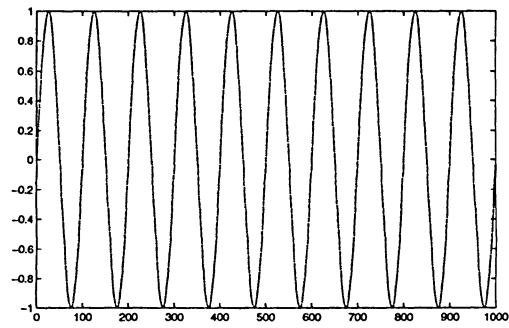
By *downsampling* the resultant A and D samples [11], it produces the DWT coefficients.

For a better understanding, a one stage discrete wavelet decomposition is shown. A pure sinusoid signal, Figure 2.12, is added with high frequency noise to generate the sample of study, as shown in Figure 2.13. One-level DWT is performed on the sample signal. It produced two sets of coefficients, Approximation (Figure 2.15) and Detail (Figure 2.16). Notice that the detail coefficients are small and consist mainly of a high-frequency noise, while the approximation coefficients contain much less noise than does the original signal of study.

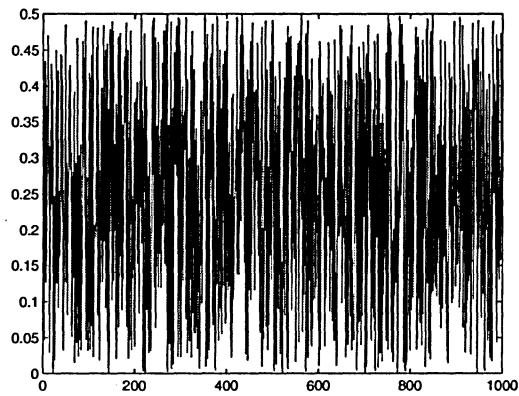
### Multi-Level Decomposition

This decomposition process can be iterated, with successive approximation coefficients being decomposed in turn, so that one signal is broken down into many lower resolution components. The resultant is called the wavelet decomposition tree.

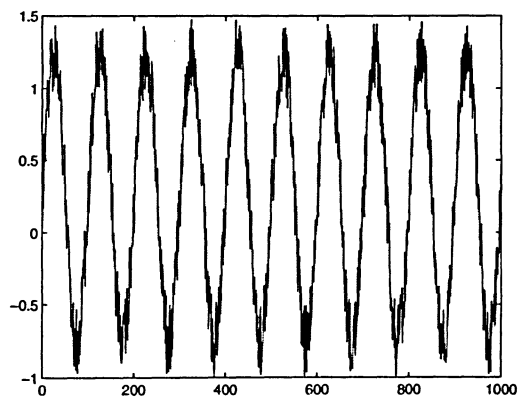
By continuing the example in Section 2.5.2, the decomposition tree can provide valuable information. By inspecting the Level Three Approximation Coefficients with the structure of the original signal, Figure 2.20, one can observe that it has removed almost all of the high



**Figure 2.12:** The pure Sinusoidal Wave.



**Figure 2.13:** The High frequency content random noise.



**Figure 2.14:** The sinusoidal wave added with high frequency random noise.

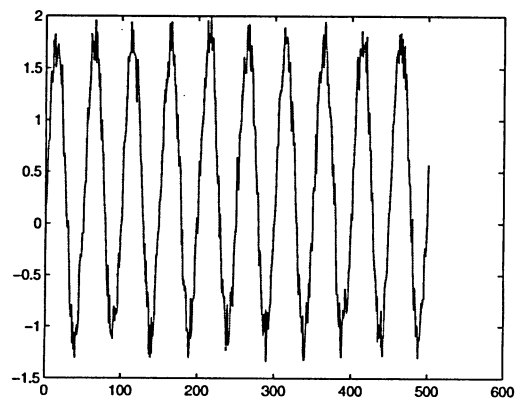


Figure 2.15: Level One Approximation Coefficients.

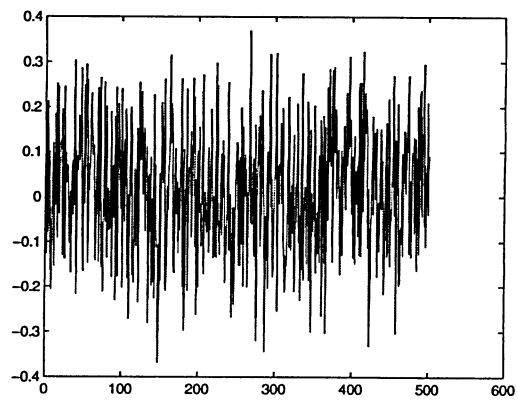


Figure 2.16: Level One Detail Coefficients.

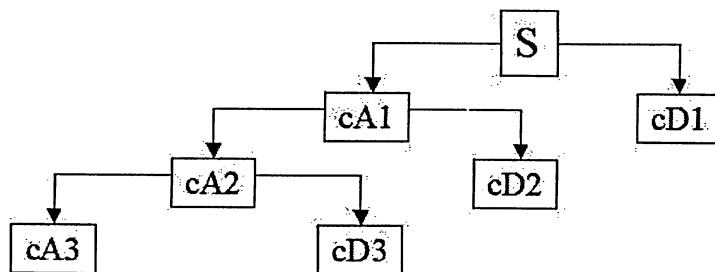
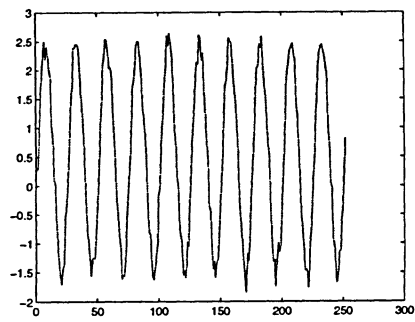


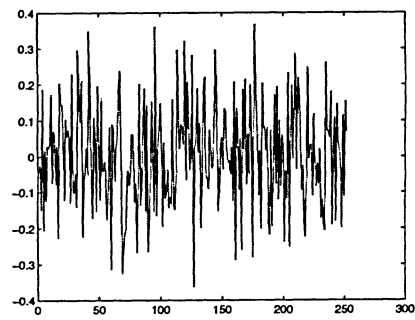
Figure 2.17: Multi-Level DWT Decomposition.



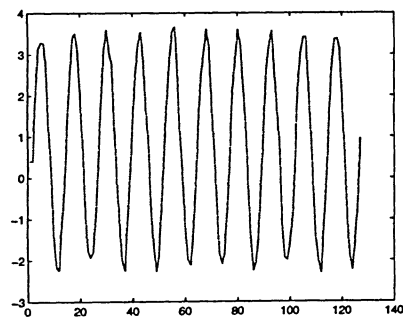
frequency random noise.



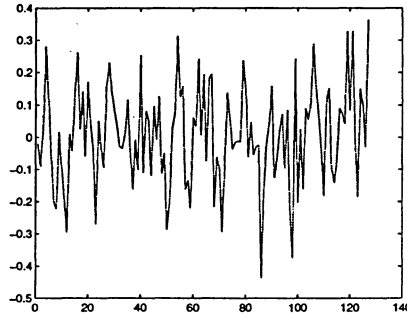
**Figure 2.18:** Level Two Approximation Coefficients.



**Figure 2.19:** Level Two Detail Coefficients.



**Figure 2.20:** Level Three Approximation Coefficients.



**Figure 2.21:** Level Three Detail Coefficients.

Since the analysis process is iterative, it can be continued indefinitely. However, in terms of processing, the decomposition can proceed only until the individual details consist of a single sample. In practice, the selection of a suitable number of levels is based on the nature of the signal, or on a suitable criterion such as entropy [11].

### Reconstruction or Synthesis

To regain the original signal, or combining the components, without loss of information is the other half of DWT. This process is called reconstruction, or synthesis or in mathematical terms, the inverse discrete wavelet transform (**IDWT**). Wavelet analysis involves filtering and downsampling, the wavelet synthesis process consists of upsampling and filtering. In digital signal processing, *downsampling* is the process of shortening a signal component by removing elements between samples, while *upsampling* is the process of lengthening a signal component by inserting zeros between samples [35].

The filtering part of the reconstruction process is an important issue because it is the choice of filters that is crucial in achieving perfect reconstruction of the original signal. The downsampling of the filtered signal components performed during the DWT decomposition phase introduces a distortion called *aliasing* [36]. It turns out that by carefully choosing filters that are closely related (but not identical) for the decomposition and reconstruction phases of DWT and IDWT, the effects of aliasing can be removed [37, 38]. By creating a complementary filter with the original filter, the pair of filters used in combination creates one

invertible lossless filter. When these filters ( $L$  &  $H$ ) are used in pairs to decompose an input, the same pair of complementary filters ( $L'$  &  $H'$ ) has to be used to put the decomposition back together. This *exact reconstruction filters* creates an operation that is reversible (no loss of information) and restores the original signal. The low and high-pass decomposition filters ( $L$  &  $H$ ), together with their associated reconstruction (complimentary) filters ( $L'$  &  $H'$ ), form a system of what is called **quadrature mirror filters**, QMF [37].

Choosing the quadrature mirror filters is an important detail that only determines whether perfect reconstruction is possible, it also determines the shape of the *wavelet* used to perform the analysis [36, 39, 40]. Daubechies, Haar, Biorthogonal, Coiflets, Symlets, Morlet, Mexican Hat, Meyer, and other complex wavelets are all apart of the QMF family each with its certain characteristics. Researchers have studied and attempted to find the optimal wavelet for use, however, the results have been mixed and it is mainly application dependent [12].

### 2.5.3 Wavelet Packet Analysis

DWT, as it can be deduced, only decomposes and iterates on the approximation coefficients, while the remaining detail coefficients remain idle for further operations. This results in  $n+1$  ways to decompose or encode the signal for an  $n$ -level decomposition. The wavelet packet method is a generalization of wavelet decomposition that offers a richer range of possibilities for signal analysis [10]. In wavelet packet analysis, the details as well as the approximations can be split. This yields more than  $2^{2^n-1}$  different ways to encode the signal as seen in Figure 2.22. Choosing one out of all these possible encodings presents an interesting problem.

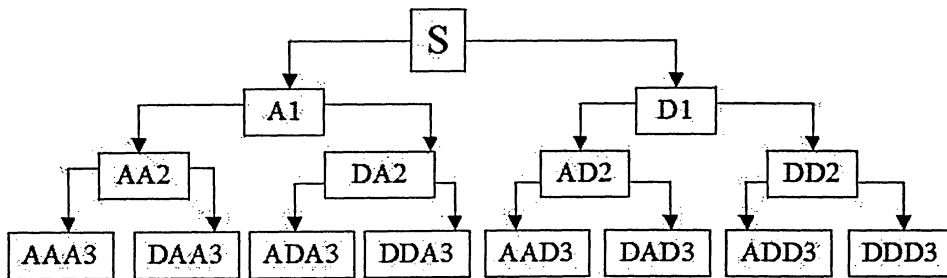


Figure 2.22: Wavelet Packet Decomposition Tree.

One of the possible methods is to use an entropy-based criterion to select the most suitable decomposition of a given signal [38]. This means we look at each node of the decomposition tree and quantify the information to be gained by performing each split. Then by using simple and efficient algorithms, the wavelet packet decomposition can be selected/pruned to represent the optimal decomposition, which has direct applications in optimal signal coding and data compression [16, 41, 42]. These adaptive filtering algorithms were developed by Coifman and Wickerhauser [37, 38].

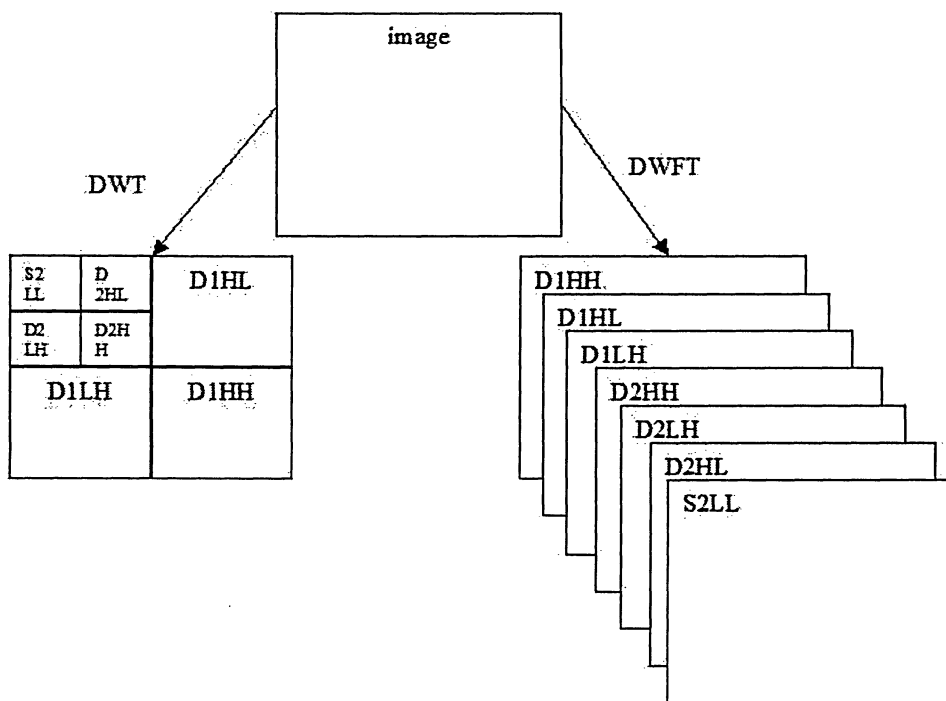
### 2.5.4 Discrete Wavelet Frame Transform

Discrete Wavelet Frame Transform resembles its DWT counterpart except that the dyadic down-sampling process is not imposed at each level. In doing so, unlike DWT, DWFT yields a shift-invariant signal representation [12, 17]. In DWT and DWFT, decomposing a 1-D signal to  $d$  decomposition levels results in  $d$  detail subbands and one approximation subband. However, the main difference is that the sizes of the subbands in DWT decreases as the decomposition is iterated. Each subband in DWFT has the same size as the original signal of decomposition (Figure 2.23). This redundant representation of DWFT is, thus, more demanding in terms of both memory and time.

### Image Processing

In image processing, we are mostly interested in the two-dimensional extension of the wavelet transform. Figure 2.24 shows one stage of the 2-D DWT and inverse DWT. As shown, the DWT is implemented using QMFs, a bank of 1-D low-pass ( $h$ ) and high-pass ( $g$ ) analysis filters, and the inverse DWT likewise uses its complement,  $(\hat{h})$  and  $(\hat{g})$ . After one stage of decomposition, an image at resolution level  $i$  will be decomposed into four subband images  $D_{LH}^{i+1}$ ,  $D_{HL}^{i+1}$ ,  $D_{HH}^{i+1}$  and  $S_{LL}^{i+1}$ . The three detail images,  $D_{LH}^{i+1}$ ,  $D_{HL}^{i+1}$  and  $D_{HH}^{i+1}$  correspond to the low-high (Horizontal), high-low (Vertical) and high-high (Diagonal) bands in the frequency domain respectively. The remaining low-low (Approximation) component  $S_{LL}^{i+1}$ , is a low-pass filtered version of  $S_{LL}^i$ , and can be used for further subband decomposition. A DWT with  $d$  decomposition levels results in a total of  $3d + 1$  end subbands. Performing DWFT to  $d$

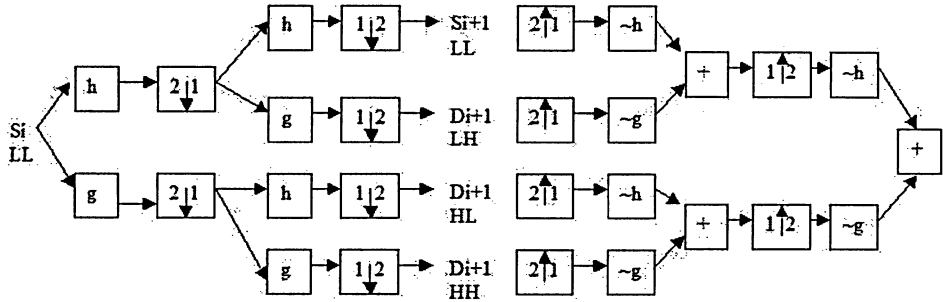
decomposition levels also results in a total of  $3d$  details subbands and one approximation subband.



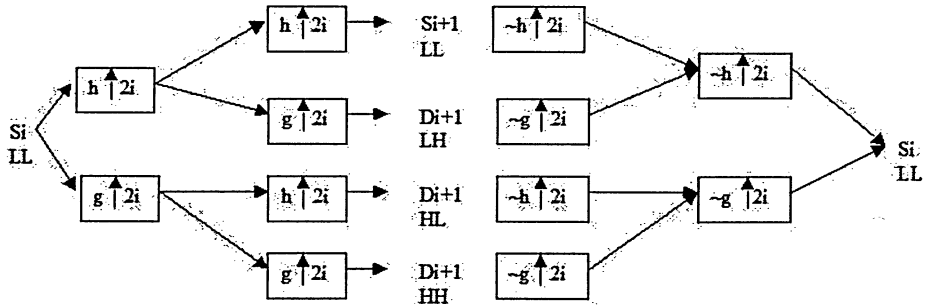
**Figure 2.23:** Image Decomposition for DWT and DWFT.

## 2.6 Summary

We have described the Wavelet Transform in terms of its history, its application and its evolution over time. Many different iterations of the wavelet transform has been shown as the general progression in the research community. The main focus was on the conceptual aspects of wavelet transform, and how it is incorporated in image processing and how it can be used as feature generation scheme for the texture classification method.



**Figure 2.24:** One stage 2-D DWT and inverse DWT. On the decomposition side, we have the down-samplers.  $2 \downarrow 1$  denotes keeping one column out of two.  $1 \downarrow 2$  denotes keeping one row out of two. On the synthesis side, the up-samplers put column and row of zeros between each column and row.



**Figure 2.25:** One stage 2-D DWFT and inverse DWFT. On the decomposition side, we have no the down-samplers.  $h \uparrow 2^i$  denotes inserting zeros between every samples in filter  $h$ , and similarly for  $g \uparrow 2^i$ ,  $\hat{h} \uparrow 2^i$  and  $\hat{g} \uparrow 2^i$ .

## Chapter 3

# Support Vector Machines

THE Support Vector Machine (SVM) Learning Classifier is essentially derived for cases of binary classification. The main focus of the SVM is to find and construct a hyperplane, otherwise known as the decision surface, in such a way that the margin of separation between the two classes, in this case positive and negative examples, are maximized. The goal is motivated by principles of statistical learning theory and the method of structural risk minimization [22], [23]. According to the statistical learning theory, the error rate or the *Generalization Error Rate* of a learning machine on the test data is bounded by the sum of two terms. The first term named *Training Error* and the second term that depends on Vapnick-Chervonenkis (VC) dimension [22] named *Structural Risk*. In the case of separable patterns, SVM produces a value of zero for the first term, while minimizing the second term. Accordingly, the optimal hyperplane which is sought by SVM is equivalent to minimizing the bound on the VC-dimension.

SVM is capable of generalizing well (predicting the unseen or unknown samples with a good degree of accuracy) as compared to many traditional classifiers (Neural Networks, etc). It offers several advantages which are typically not found in other classifiers:

- Computationally much less intensive (when compared to the iterative nature to Neural Networks).
- Performs well in higher dimensional spaces, a factor which limits many efficient classifiers.

- Lack of training data is often not a severe problem.
- Robust with noisy data, where noise can severely degrade the performance of other classifiers.
- Does not suffer as much from the curse of dimensionality and prevents over-fitting.

In the following sections, we will first review the mathematics of the derivation of the Support Vector Machine in the simple case where, patterns are linearly separable, and the difficult case of when the pattern are non-linearly separable case.

### 3.1 Linear Support Vector Machine Classifier

Let vector  $\mathbf{x} \in \mathbf{X}$  denote a pattern to be classified, and let scalar  $y$  denote its class label  $y \in [\pm 1]$ . In addition, let  $\{(x_i, y_i), i = 1, 2, \dots, l\}$  denote a given set of  $l$  training examples.

The main problem in any classifier, is how to generate a decision surface function or hyperplane,  $f(x)$ , that can correctly classify an input pattern that may or may not be from the training set. Given the linearly separable case, there exists a linear function  $f(x)$  of the form:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{b}, \quad (3.1)$$

such that for each training example  $x_i$ , the function yields  $f(x_i) \geq 0$  for  $y_i = +1$ , and  $f(x_i) < 0$  for  $y_i = -1$ . In other words, training examples from the two different classes are separated by the hyperplane  $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{b} = 0$ . For a given training set, while there may exist

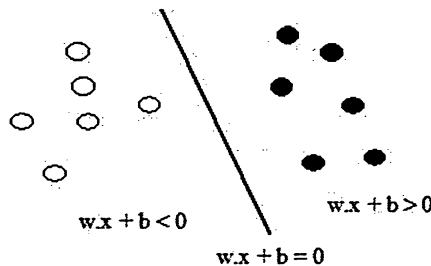
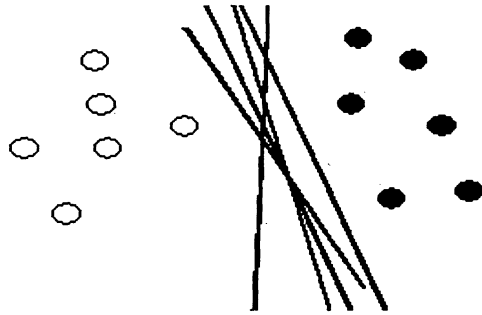
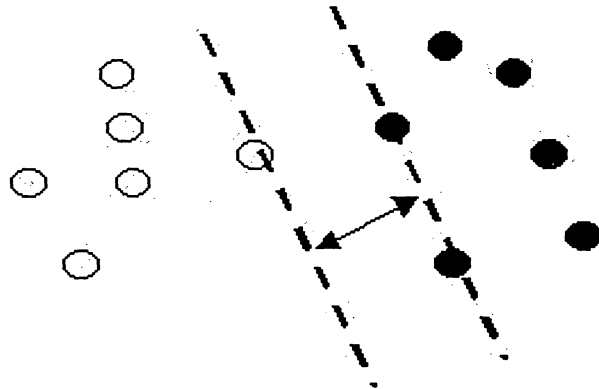


Figure 3.1: Sample Linear Hyperplane.





**Figure 3.2:** Various Hyperplanes in existence.



**Figure 3.3:** Optimal Hyperplane. Extracted support vectors are shown in bold.

many hyperplanes that separate the two classes (Figure 3.2), the SVM classifier is based on the hyperplane that maximizes the separating margin between the two classes. In other words, SVM finds the hyperplane that causes the largest separation between the decision function values for the borderline members of the two classes referred as support vectors, Figure 3.3. This hyperplane can be mathematically found by minimizing the following cost function:

$$\Psi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} = \frac{1}{2} \|\mathbf{w}\|^2. \quad (3.2)$$

Subject to separability constraints:

$$\mathbf{w}^T x_i + b \geq +1, \text{ for } y_i = +1, \quad (3.3)$$

and

$$\mathbf{w}^T x_i + b \leq -1, \text{ for } y_i = -1, \quad \text{where } i = 1, 2, \dots l. \quad (3.4)$$

Equivalently, these constraints can be written more compactly as

$$y_i(\mathbf{w}^T x_i + b) \geq +1, \text{ for } i = 1, 2, \dots l. \quad (3.5)$$

This specific problem formulation may not be useful in practice because the training data may not be completely separable by a hyperplane. In this case, a slack variables, denoted by  $\xi_i$  can be introduced to relax the separability constraints in Equation 3.5 as follows:

$$y_i(\mathbf{w}^T x_i + b) \geq 1 - \xi_i, \text{ where } \xi \geq 0; i = 1, 2, \dots l. \quad (3.6)$$

Accordingly, the cost function in Equation 3.2 can be modified as follows:

$$\Psi(\mathbf{w}, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i, \quad (3.7)$$

where,  $C$ , is a user specified, positive, regularization parameter. In Equation 3.7, the variable  $\xi$  is a vector containing all the slack variables  $\xi$ ,  $i = 1, 2, \dots l$ .

The modified cost function in Equation 3.7 represents the structural risk. This equation is a fine balance between the second term, empirical risk (i.e.- training error) with first term (model complexity). The regularization parameter  $C$  controls this trade-off. The purpose

of using model complexity to constrain the optimization of empirical risk is to avoid over-fitting, which is a usual occurrence in Neural Networks. This is a situation where the decision boundary is constructed too precisely with the training data and ultimately fails to perform well on data outside the training set. This is discussed more in Section 3.2.

## 3.2 Non-Linear Support Vector Machine Classifier

The set of equations are clean and simple with linearly separable data. However, the data set is unlikely to be linearly separable as seen in Figure 3.4. Support Vector Machines uses Cover's Theorem [23] to overcome the nature of non-linearly separable data. The theorem specifically states that such a non-linearly separable pattern space can be transformed into a new feature space where patterns are linearly separable with a high probability. However, the transform must be of non-linearity and the feature space dimension must be a high enough value. Let  $\mathbf{x}$ , with a dimension of  $m_0$  denote the vector drawn from the input space

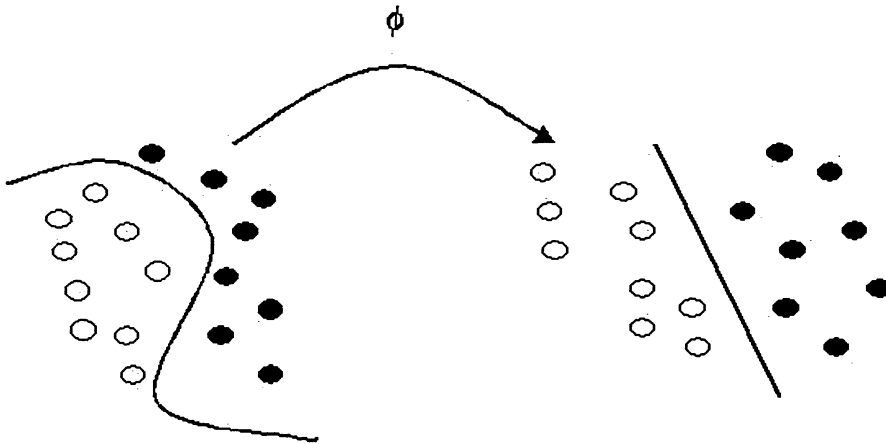


Figure 3.4: Feature Mapping.

**X.** Let  $\{\Phi_j(\mathbf{x})\}_{j=1}^{m_1}$  denote the set of non-linear transformations from the input space to the feature space dimension of  $m_1$ . It is assumed that  $\{\Phi_j(\mathbf{x})\}_{j=1}^{m_1}$  is defined *a priori* for all  $j$ . Then, a hyperplane acting as decision function is defined as:

$$\sum_{j=1}^{m_1} w_j \Phi_j(\mathbf{x}) + b = 0, \quad (3.8)$$

where  $\{w_j\}_{j=1}^{m_1}$  denotes the set of linear weights joining the feature space to output space and where  $b$  is simply the bias. The Equation 3.8 can be simplified by writing:

$$\sum_{j=1}^{m_1} w_j \Phi_j(\mathbf{x}) = 0, \quad (3.9)$$

where it is assumed that  $\Phi_0(\mathbf{x}) = 1$  for all  $\mathbf{x}$ . In doing so,  $w_0$  denotes the bias  $b$ . Equation 3.9 defines the decision function computed in the feature space in terms of the linear weights of the support vector machine. The quantity  $\Phi_j(\mathbf{x})$  represents the input supplied to the weight  $w_j$  via the feature space. In fact, the vector  $[\Phi_0(\mathbf{x}), \Phi_1(\mathbf{x}), \dots, \Phi_{m_1}(\mathbf{x})]^T$  can be considered the “image” induced in the feature space due to the input vector  $\mathbf{x}$ .

The problem of finding weight coefficients  $\mathbf{w}$  can be formulated as an optimization problem with a constraint. From [23], it can be shown that finding optimal hyperplane is equal to minimizing the cost function:

$$\Psi(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i. \quad (3.10)$$

A large value of the regularization parameter  $C$ , which is selected by the user, corresponds to a higher penalty being assigned to the training errors [67]. A lower value of  $C$  has the reverse effects.

Given the training samples,  $\{(x_i, d_i)\}_{i=1}^N$ , the constraints which *must* be satisfied are:

$$d_i(\mathbf{w}^T \Phi(x_i) + b) \geq 1 - \xi_i, \text{ for } i = 1, 2, \dots, N. \quad (3.11)$$

This constrained optimization problem can be solved using the *Lagrange* multiplier. The *Lagrangian* function is constructed as:

$$J(\mathbf{w}, \alpha, \xi, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [d_i(\mathbf{w}^T \Phi(x_i) + b) - 1 + \xi_i], \quad (3.12)$$

where the non-negative variables of  $\alpha_i$  are called Lagrange Multipliers. The solution to constrained optimization problem is determined by the saddle point of the Lagrangian function  $J(\mathbf{w}, \alpha, \xi, b)$ , which has to be minimized with respect to  $\mathbf{w}$  and  $b$  and maximized with respect

to  $\alpha$ . Applying the optimality conditions:

$$\frac{\partial J(\mathbf{w}, \alpha, \xi, b)}{\partial w} = 0, \quad (3.13)$$

$$\frac{\partial J(\mathbf{w}, \alpha, \xi, b)}{\partial b} = 0, \quad (3.14)$$

$$\frac{\partial J(\mathbf{w}, \alpha, \xi, b)}{\partial \xi} = 0, \quad (3.15)$$

to the Lagrangian function Equation 3.12 yields:

$$\mathbf{w} = \sum_{i=1}^N \alpha_i d_i \Phi(x_i). \quad (3.16)$$

Given the equations above, the final structure of the non-linear SVM classifier is obtained as:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i d_i \Phi^T(x_i) \Phi(x_i) + b = \sum_{i=1}^N \alpha_i d_i K(x_i, \mathbf{x}) + b, \quad (3.17)$$

where the kernel function  $K(\cdot, \cdot)$  is defined as:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T \mathbf{x}_2. \quad (3.18)$$

Lagrange Multipliers are solved from the dual form of Equation 3.10, which is expressed as:

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_j K(x_i, x_j) = 0, \quad (3.19)$$

which are subjected to the constraints:

$$0 \leq \alpha_i \leq C, \text{ where } i = 1, 2, \dots, N, \quad (3.20)$$

$$\sum_{i=1}^N \alpha_i d_i = 0. \quad (3.21)$$

This dual problem is solved numerically through quadratic programming (QP). The Karush-Kuhn-Tucker optimality conditions for Equation 3.19 leads to the following three cases for each  $\alpha_i$  as shown in Figure 3.5:

1.  $\alpha_i = 0$ .

- This corresponds to  $d_i(x_i) > 1$ . In this case, the data element  $x_i$ , is outside the decision margin of the function  $f(\mathbf{x})$  and is correctly classified. This is related to null Lagrange multipliers. These vectors are irrelevant in the future use of the SVMs.

2.  $0 < \alpha_i < C$ .

- This corresponds to  $d_i(x_i) = 1$ . In this case, the data element  $x_i$ , is strictly located on the decision margin of  $f(\mathbf{x})$ . Hence  $x_i$  is referred as a margin *support vector* (SV) for  $f(\mathbf{x})$ . These are correctly classified vectors placed on the margins.

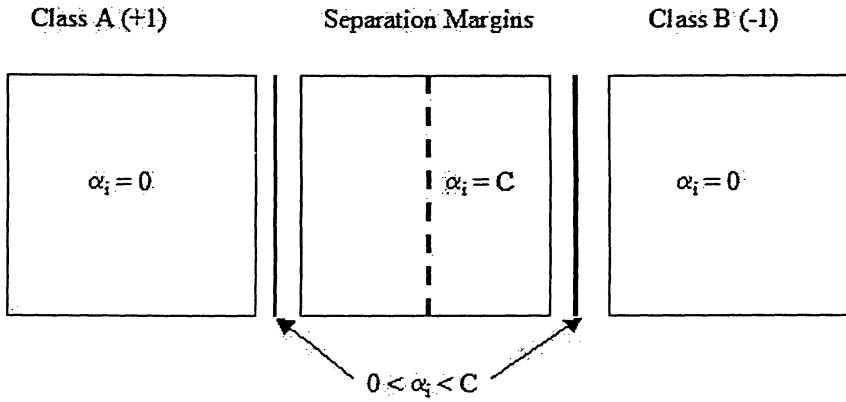
3.  $\alpha_i \cong C$ .

- This corresponds to  $d_i(x_i) < 1$ . In this case, the data element  $x_i$ , is inside the decision margin (though it may be still correctly classified). Accordingly,  $x_i$  may be called an error *support vector* of  $f(\mathbf{x})$ .

The last two groups of vectors are known as support vectors and define the separation hyper-plane between the classes.

Most of the training samples in a typical problem are correctly classified by the trained classifier (Case 1), i.e. only a few training samples would be support vectors. For simplicity, let the set  $s_j$  contain  $\alpha_j^*$ , where  $j=1,2,\dots,N_s$ , which denotes those support vectors and their corresponding nonzero Lagrange multipliers respectively. The decision function in Equation 3.17 can now be simplified as:

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i^* d_i K(s_i, \mathbf{x}) + b. \quad (3.22)$$



**Figure 3.5:** Separation Margins of Lagrange multipliers given the two classes of A & B.

As it can be seen in Equation 3.19 and 3.22, the nonlinear mapping  $f(\cdot)$  never appears explicitly in either dual form of SVM training problem or the resulting decision function. The mapping  $f(\cdot)$  only enters the problem implicitly through the kernel function  $K(\cdot, \cdot)$ . Thus, it is only necessary to define  $K(\cdot, \cdot)$  which implicitly defines  $f(\cdot)$ . However, when choosing a kernel function, it is necessary to check that it is associated with the inner product of some linear mapping. Mercer's theorem [22] states that such a mapping indeed underlies a kernel  $K(\cdot, \cdot)$  provided that  $K(\cdot, \cdot)$  is a positive integral operator. That is, for every square-integrable function  $g(\cdot)$  defined on  $\mathcal{R}^n$  the kernel  $K(\cdot, \cdot)$  satisfies the following condition of:

$$\int \int K(x, y)g(x)g(y)dxdy \geq 0. \quad (3.23)$$

### Common Kernel Functions

Given the above conditions, shown in the Table 3.1 are a few common kernels most commonly in practice and applications of the SVM kernel [43]. The functions shown avoid the increased computational complexity and curse of dimensionality, a kernel-trick or kernel function  $K(\mathbf{x}, \mathbf{y})$  is employed which, computes an equivalent kernel value in the input space such that no explicit mapping is required [43].

**Table 3.1:** Common Kernel Functions.

Gaussian RBF	$k(\mathbf{x}, \mathbf{y}) = e^{\frac{-\ \mathbf{x}-\mathbf{y}\ ^2}{c}}$
Polynomial	$((\mathbf{x} \cdot \mathbf{y}) + \theta)^d$
Sigmoidal	$\tanh(\kappa(\mathbf{x} \cdot \mathbf{y}) + \theta)$
Inv. Multiquadric	$\frac{1}{\sqrt{\ \mathbf{x}-\mathbf{y}\ ^2 + c^2}}$

### 3.3 SVM Applications

Being a late member into the classifier community, SVM has received much attention because of its ability to overcome dimensionality issues, while keeping the accuracy of the results at a high level. The heart of the SVM engine is based upon the mathematical and statistical theory. This sound base in theory gives SVM its power to be applicable to many different areas and domains of research.

There has been many papers written since its inception and even more work has been published in the past five years. The domains SVM has covered over those years are in the same category as with other popular classifiers such as Neural Networks. The main reason for its gain to popularity was by comparison and providing its ability to reach high classification accuracies versus other classifiers.

SVM does not depend explicitly on the dimensionality of the problem, but to the *margin* with which it separates the data. Superior performance is achieved even on high dimension data sets like US postal service database of handwritten digits using 16x16 bitmap as input [44, 45]. Face Detection experiments [46] where dimensionality was 283 [47]. Some of the other domains are:

- Texture Classification [48, 49, 50].
- Medical Imaging: Detection of Microcalcifications [51] and Oral Lesion Classification [52].
- Satellite Imagery: Synthetic Aperature Radar (SAR) [53].



- Voice Recognition [54].

As stated earlier, most of these published works provided comparisons to popular traditional classifying methods. As SVM becomes more popular in the research community and more research is put into its mathematical derivations and applicability, the future is bright with possibilities.

## 3.4 Summary

We have described the Support Vector Machines as in terms of its Mathematical fundamentals and how it is applied in classification theory. The SVM is a relatively new classifier in terms of history and interest from the research community. With its powerful methods, it can only gain more acceptance by more researchers studying it and applying it to their respective fields.

# Chapter 4

## Optimal Filter Bank Representation

**T**EXTURE analysis plays an important role in many tasks. The main difficulty of texture analysis in the past was the lack of adequate tools to characterize different scales of texture effectively. To overcome this fact, researchers have attempted to duplicate the ability of the human brain, to understand the content of images by interpreting the shape and texture of the object in the scene, by developing image understanding algorithms for varied applications from robotic vision, industrial monitoring, remote sensing to assisted medical diagnosis.

### 4.1 Introduction

With its importance in many fields from real to synthetic images, there is no universal definition of texture due to the sheer diversity of the patterns involved, given the set of natural and artificial textures defined. The definition of a texture can be defined in many ways:

- A texture may be regarded as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule.
- A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying or approximately

periodic.

- The image texture which is considered is non figurative and cellular. An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of its tonal primitives.

However, these definitions cover many of the same ideas that a *texture* should encompass for it to be passed as a one. Sklansky [55] summed them and derived a new definition of a texture. It is defined as *an image region that has a constant texture if a set of local properties in that region is constant, slowly varying or approximately periodic*. These local statistics or properties that is repeated over the textured region is called a texture element or *texel*. This meaning of texture has both local and global meaning, that it is characterized by invariance of certain local attributes that are distributed over a region of an image.

The proper analysis of texture requires the identification of proper attributes or features that differentiate the textures for classification, segmentation and recognition. Various feature extraction and classification techniques have been suggested in the past for the purpose of texture analysis and is reviewed next.

## 4.2 Non-Filtering

### 4.2.1 Statistics

Texture analysis was based on first order or second order statistics of textures introduced by Haralick [4]. He and the respective authors [4] proposed generating features from the co-occurrence matrix. In this method, the relative frequencies of gray-level *pairs* of pixels at certain relative displacements are computed and stored in a matrix,  $\mathbf{P}$ . For  $G$  gray-levels in the image,  $\mathbf{P}$  is of size  $G \times G$ . If the number of gray-levels is low, much of the texture information is lost in the image quantization. The possible combinations of the nearest neighbor pairs used most commonly are at orientations of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$  [12]. There were 14 features [4] defined (i.e. - Contrast, Energy, Entropy, Local Homogeneity, Cluster Shade, Cluster Prominence, Max Probability, etc.). However, their computation for different

distances at different orientations increases the computational and time complexity. Even when all the features were used, the accuracy of the classification rate was only at 60-70%.

### 4.2.2 Model-Based

With model based features, some image model is assumed, its parameters estimated for a subimage, and the model parameters, or attributes derived from them, are used as features. Gaussian Markov Random Fields (GMRF) as it was proposed by Chatterjee and respective authors [6] to characterize textures in using the models. Building upon the work for GMRF, local linear transformations were used to compute texture features by Laws [56] and Unser [57]. These traditional statistical approaches to texture analysis were **restricted** to the analysis of spatial interactions over relatively small neighborhoods on a single scale, which resulted in their relative poor performance.

## 4.3 Fixed Filters

### 4.3.1 Laws Filter Masks

One of the first approaches to filtering for texture identification was presented in the work by Laws [56]. Laws and respective authors suggested using a bank of separable filters, five in each dimension, for a total of 25 filters. The filter masks suggested in the work were:

- $h1 = [1, 4, 6, 4, 1]$ ,
- $h2 = [-1, -2, 0, 4, 1]$ ,
- $h3 = [-1, 0, 2, 0, -1]$ ,
- $h4 = [-1, 2, 0, -2, 1]$  and
- $h5 = [1, -4, 6, -4, 1]$ .

### 4.3.2 Discrete Cosine Transform

The discrete cosine transform (DCT) is popular in image coding such as JPEG compression standard due to good performance and fast implementation [58]. Ng [58] suggested using a 3x3 DCT for texture feature extraction. They, furthermore, suggested excluding the low-frequency component of the DCT, thus, yielding 8 features. These image transforms are equivalent to critically sampled filter banks. The filter bank can be separable, which are determined by the one-dimensional filter masks used for the operations [12].

### 4.3.3 Gabor Filter

Jain and Farrokhnia [9] suggested a bank of Gaussian shaped band-pass filters, with dyadic coverage of the radial spatial frequency range and multiple orientations termed *Gabor filters*. This choice was justified by the relation to models for the early vision of mammals as well as the filters' joint optimum resolution in time and frequency [9]. The basic even-symmetric Gabor filter oriented at  $0^0$  is a band-pass filter with unit pulse response

$$h(k, l) = e^{-\frac{1}{2}(\frac{k^2}{\sigma_x^2} + \frac{l^2}{\sigma_y^2})} \cos(2\pi f_0 k), \quad (4.1)$$

where  $f_0$  is the radial center frequency. Other orientations are obtained by rotating the reference coordinate system,  $h(k, l)$ . This filter has an infinite unit pulse response, but in practice, it is approximated by a finite filter. Five radial frequencies are suggested in [9] (for images of size 256 x 256) at four orientations of  $0^0$ ,  $45^0$ ,  $90^0$ , and  $135^0$ . The discrete radial center frequencies were:

$$\frac{\sqrt{2}}{2^6}, \frac{\sqrt{2}}{2^5}, \frac{\sqrt{2}}{2^4}, \frac{\sqrt{2}}{2^3}, \text{ \& } \frac{\sqrt{2}}{2^2}. \quad (4.2)$$

### 4.3.4 Wavelet Transforms, Packets, and Frames

A major disadvantage in using Gabor Transform is that the output of the Gabor filter banks are not *mutually orthogonal*, which results in a significant correlation between texture features. Plus, Gabor filtering is not a reversible process, which limits their applicability for texture synthesis. These deficiencies were avoided with the introduction of Wavelet Transform, which provided a precise and unifying framework for the analysis and characterization

of signal at different scales. Another major advantage of Wavelet Transform over Gabor filter was its use of low-pass and high-pass filters remained the same between consecutive scales, while the Gabor approach required filters of different parameters and proper tuning of its parameters at different scales.

An unpleasant consequence of the filter bank approaches is that they are to varying degree computationally demanding. For these reasons, the design of optimal filters for texture discrimination is gaining increased interest [12]. Optimization of the filters offers the potential of reduced feature dimensionality and, hence, reduced computational complexity and/or better feature separation. Other popular filter bank approaches are based on least mean squared linear prediction error filters [59]. The linear prediction error filters are optimized with respect to the prediction error. This filter bank design approach is applicable to problems with arbitrary numbers of textures, and the number of filters is equal to the number of textures. Prediction error filtering and its family of least squares auto regressive (AR) parameter estimation is not the focus of this thesis and is not reviewed.

Many authors have proved that Wavelet Transform is a viable transformation for texture classification. Chang and Kuo [18] provided the conventional pyramid structure wavelet transform, whose classification rate was higher than other methods such as Gabor filter and Tree structured wavelet transform. They have also indicated that texture features are most prevalent in intermediate frequency bands, where Wavelet Packets would become of optimal use in the respective task. Unser [57] and Randen [12] also proved that texture classification using wavelet transform was comparable if not better than other methods used.

## 4.4 Applications

Many have used the wavelet transform texture analysis structure for many image classification needs. Medical Imaging has produced much interest from the image processing community. Wavelet Transform has been used in classification of MR Images of brain [60], Myocardial Tissue [33], and many more which can be easily found in various journals and conferences.

## 4.5 Related Work

### Feature Extraction

In the texture classification domain, Wavelet Transform has been well documented and applied. Besides having good empirical performance, wavelets has a solid foundation based on a formal mathematical theory for multi-resolution image analysis. The orthonormal design of wavelet transform has the implied property which is of high importance for texture analysis. The data is decomposed into independent frequency channels, which allows a coarse selectivity of three spatial orientations in image processing of horizontal, vertical & diagonal, and the orientations correspond to maximum sensitivity of the Human Visual System [62].

Laine and authors [10] introduced an approach to characterize textures at multiple scales. They had used *Wavelet Packet Signature* to generate energy and entropy metrics of natural textures, which were incorporated into distinct scale space representations. These results were compared with the standard wavelet decomposition. Major disadvantage of this method was its use of samples sizes which were 128x128 pixels, and the smallest window size of study can be no less than 32x32 pixels.

Kuo and respective authors [18] developed the multi-resolution approach based on modified wavelet transform called *Tree-Structured Wavelet Transform*. This was developed because of a large class of natural textures that can be modelled as signals whose dominant frequencies are located in the middle frequency channels. This gave the ability to zoom into any desired frequency channel for further decomposition. However, the classification results are dependent on the test set having the same wavelet tree structure of the training set.

Unser [17, 57] developed the approach to characterize texture properties at multiple scales using the over-complete wavelet decomposition, which yielded translation-invariant features. He also implemented the tight frame approach, which yielded a fast iterative algorithm. However, the classification experiments were only conducted with 12 natural textures. A more detailed approach was needed to hold the results to a standard.

Randen [12] had reviewed most major filtering approaches to texture feature extraction and performed a comparative study. He compared Laws masks, Gabor filter banks, wavelet

transforms, wavelet packets, discrete cosine transforms and linear predictors. They were all tested under extensive experiments and were given a ranking based on their results. The overall conclusion was that wavelet transforms ranked at the top of most of these experiments.

Utilizing this, several filter bank approaches and related schemes have been proposed, as stated above. Similar approaches and extensions to these works can be found in literature by Mojsilovic [33], Li et al. [62], and Rajpoot [63]. However, there are many others who have implemented the wavelet transform filtering method, which will be too many to state here.

Arivazhagan [64] took a different approach. He provided a merger of the filtering with the non-filtering approaches. He implemented use of Mallat's [15] wavelet transform in conjunction with Haralick's [4] co-occurrence matrix features. He used the wavelet transform decomposed images to generate features based statistical and co-occurrence features, which were used as a feature database approach in texture classification. The results provided promising results.

## Classification

Another crucial component in texture classification is the classifier being used. In work stated above, they each used a different classifier to obtain the overall classification accuracies. Laine and Arizvahagan had used the Minimum Distance classifier. Kuo and Unser had used the Bayes, minimum error Bayes and Neural Network classifiers. Randen mainly used the LVQ classifier. Mojsilovic implemented a distance metric suited towards their scheme, which was similar to Mahalabonis distance classifier.

In most of the work, they had dimensionality problems beyond the feature extraction phase and some of the authors had further used transforms to reduce the dimensionality. Rajpoot and Kwok had proposed the use of the SVM classifier because of its robust capabilities to handle the dimensionality issue. We used the SVM classifier because of its ability to remove this variable out of the texture classification equation.



## 4.6 Motivation

The first motivation is to implement a standardized testing, similar to [62] that does not embellish the final classification results. In doing so, a platform is generated to compare the different methods and conclusions can be derived.

Secondly, it is clear that replacing DWT with DWFT into Arivazhagan’s [64] method of feature extraction, the WSFs and WCFs would better be represented because of the over-complete representation and shift-invariant nature of the DWFT elements. Furthermore, the feature space can be increased in the feature vector representation, thus, providing more discriminatory characterization of the texture classes.

Finally, the basic assumption for the filtering approaches stated above was that the energy distribution in the frequency domain identifies a texture. Hence, if the frequency spectrum is decomposed into a sufficient number of subbands, the spectral energy signatures of different textures are different, as per Equation 4.3, where “ $I$ ” represents the decomposed image  $M \times N$ . Thus using these values in a Feature Vector (FV) to be used in the classifier, the feature vectors provides enough discriminating power to separate itself from other textures.

$$Energy = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N I^2(x, y). \quad (4.3)$$

Arivazhagan [64] proposed to use the DWT features and co-occurrence features (Appendix A.1 through A.10) computed out of the subbands of wavelet transformed images. The intuition behind this approach was that the chances for correct classification is considerably improved if the higher order statistical features are used, as they normally have good discriminating ability than the lower order ones.

Many experiments in the research community implement the DWT to generate features to characterize the texture for optimal discrimination. The clear fact that DWT’s inherent property of critical sampling at each stage of decomposition can be seen as savior to time and computational complexity, however, it has forsaken the spatial information integrity when the features are being generated. This was resurrected by Unser’s [17] research work into Discrete Wavelet Frame Transform, DWFT. By restoring the spatial veracity at each decomposition

level, the respective decomposed images can enhance the discrimination ability of the feature vector corresponding to the texture. In doing so, this adds to the computational cost and time expense, however, the classification accuracy results which are discussed in Section 5.2 justify this reasoning.

Revisiting Arivazhagan's approach at adding additional features to improve the success rate can be furthered by the proper implementation of the DWFT. His approach was only able to use subbands of level-one decomposed images because of the critical sampling caused by the inherent properties of the Discrete Wavelet Transform. Much information is lost with each level of decomposition even at the start of the decomposition at level-one. For example, an image of small dimension of size 32x32 pixels is decomposed using DWT. The third level of decomposition generates images of 8x8 pixels because of the critical dyadic sampling. If one were to continue to use the DWT decomposed images for co-occurrence feature generation, the spatial information that had been lost would not be suffice to support the calculations required by Haralick's co-occurrence matrix and subsequent features (Appendix A.3 to A.8).

By implementing the DWFT with Arivazhagan's approach, the ability to capture more features per decomposition can be variably increased. A five-level DWFT decomposition produces 5 Approximations, and 15 Detail images (5 levels multiplied by 3 details per decomposition), all of which are of same size as the original image. This totals to 20 decomposed images. By generating the co-occurrence matrix for each image and deriving the co-occurrence features (A.3 through A.8), there is a possibility of accumulating 140 ( $7 \times 20$ ) features from a single five-level decomposition. This is in comparison to any one feature generation as entropy or energy from Equation 4.3, which only produces 20 ( $20 \times 1$ ) features.

Furthermore, we are motivated at aiming at providing more discriminatory information that has now abundantly been generated by the DWFT and ensuing co-occurrence features. The goal is to capture the data that most provides discrimination ability of the feature in regards to the texture. This is accomplished by focussing and localizing only on certain areas of the image to generate the necessary features (energy, co-occurrence matrix features) using a filtering technique with a specified structuring element.

## 4.7 Background

The heuristically designed filter banks of DWFT imply large numbers of features. Consequently, the computational complexities are large in feature extraction stage. Hence, it may be desirable to attempt to discriminate by effect of reducing the data of interest operated upon. Focussing on certain elements at the filter banks may yield a maximized feature separation by the removal of the redundant data. This step is compensated with information derived from the multiple input bands and the subsequent decomposed images are analyzed using texture analysis techniques for proper discrimination, as stated above. The number of input bands in the feature extraction stage are incorporated to enhance the discrimination of the textures.

If frequency bands are biased with different weights, one can dynamically control the receptive behavior of texture window at distinct resolutions. Decomposition using DWFT in Time-Frequency domain is the *over-complete* representation of features that describe the texture at that localized position in the signal. When attempting to focus by effect of reducing the information contributed from the subbands, it is important that the feature extraction stage derives more discriminatory information.

It is a common understanding of texture as an attribute representing the spatial arrangement of the gray levels of pixels in a region. The investigated region is called a *texture window*, which is configured in its size by the specific technique implemented. With respect to wavelet transform filtering approaches, DWT texture window decreases with each level of decomposition, while DWFT texture window stays the same size as the original sample image with each level of decomposition.

The traditional approach is to calculate the feature from the whole texture window by accumulating all the wavelet coefficients over the decomposed image/slice, as is the case using Equation 4.3. By focussing the elements used for calculation of a feature, it not only reduces the data needed to be operated upon, but it also preserves the discrimination ability of the features corresponding to the particular texture. This is dependent on the *window/filter* utilized that maximizes the number of neighborhood pixels into account for

feature calculation.

## 4.8 Methodology

In the texture characterization perspective, maximizing the amount of information used in the calculation of the features is key. One needs to match a prescribed position and where the percentage of neighborhood pixels is high. This is usually at the center of the window. If the percentage of neighborhood pixels equals to 1, this means that the whole image or texture window has been taken into account. This task is completed by applying a structuring element or *filter* on the image centered by the prescribed position and retaining only those pixels encompassed by the filter. The shape of this filter is discussed in the next section.

### Texture Window Filter

Before we define the filters that are used on the filter bank textured windows of the DWFT method, the definition of texture is defined in terms of mathematical spaces. The textures that are used in our experiments are from the Brodatz collection [68, 69]. A texture is simply a combination and permutation of gray-level values arranged in a manner that a distinctive shape has been derived [60, 61]. The gray-values are simply the intensity levels of the pixel, which range from 0 meaning the color black to 255 meaning the color white to total 256 separate values. The values in the middle are considered the different levels of the color gray.

An image space  $F$  is the set containing all the  $K$  possible gray values:

$$\text{gray space} : F = [f_1, f_2, \dots, f_k]. \quad (4.4)$$

A gray set of  $G$  of all possible subsets of  $F$  is defined as:

$$\text{gray set} : G = [\{\emptyset\}, \{f_1\}, \dots, \{f_k\}, \{f_1, f_2\}, \dots, \{f_1, f_2, \dots, f_k\}]. \quad (4.5)$$

As a result, one can define a gray-coded textured image as an object and an associated function  $A$ , which is defined as a projection from  $A$  to the set  $F$ :

$$\text{gray coded textured image } A : [A \rightarrow F]. \quad (4.6)$$

According to this definition, function  $A$  assigns a gray-code to each pixel. The gray code simply named gray value, is one element from the gray set  $F$ . The gray value itself might consist of an arbitrary number of color values  $\{f_1, f_2, \dots, f_k\}$  from the gray space. Therefore,  $A$  represents the collection of those possible intensity gray-values from the set  $G$ , which have been fashioned in a manner that has represented a textured image covering a certain spatial space. This is clear by observing any of the textured images from Brodatz collection [68]. There are clear sections in the image, which are repeated over the entire image, as it can be seen Figure 4.1.

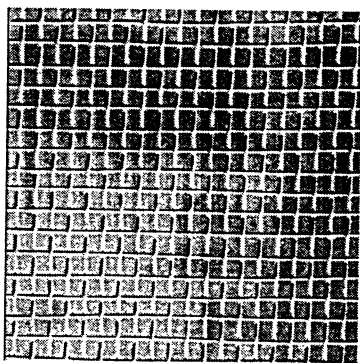
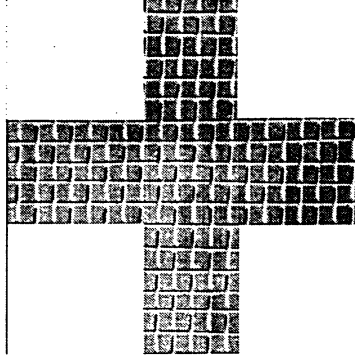


Figure 4.1: D1, Brodatz image collection.



Figure 4.2: The *Cross* Filter.

This proposition, by localizing on the specific area of the image would improve the ability to characterize the image by using the features defined. To localize the image with respect



**Figure 4.3:** D1 intersected with *Cross* Filter.

to the filter, the filter captures only those areas of interest in the image to generate a new image. This new image is further processed for feature extraction.

This can be defined using the Equation 4.7 and 4.8, where  $A$  is represented as the image, and  $B$  is represented as the filter, both of which are of same dimensions. The filter  $B$  contains only two states, *active* (1), associated with the color black and *non-active* (0), associated with the color white. This can be seen with filters used in the experiment in Appendix C. When the two images are intersected, only the active regions in  $B$  retain the information in  $A$ . The rest of the pixels are discarded.

$$\text{If } A(x, y) \cap B(x, y) = 1, A(x, y) \text{ is retained.} \quad (4.7)$$

$$\text{If } A(x, y) \cap B(x, y) = \emptyset, A(x, y) \text{ is not retained.} \quad (4.8)$$

Operating with filter  $B$  (Figure 4.2) on image  $A$  (Figure 4.1) one uses the operations which are similar to those of binary operations. These operations enable us to achieve this localization effect on image  $A$  (Figure 4.3).

### Selection Criteria

The filter may enable to achieve more description of the image under operation. The filter  $B$  retains only a percentage of  $A$  pixels. This percentage,  $n$ , is given with respect to the cardinality of the filter  $B$ . If  $n$  is set to 1, the full description of the image has been fully

retained. This would represent a filter that is active in all regions. The question remains on the shape of the structuring filter. As discussed earlier in Section 4.8, that by positioning the filter in the center of the image, it maximizes the ability to cover more and retain the spatial characteristics of the image. This is why the center position is the prescribed position because of its ability to acquire more of the neighborhood pixels into account. In doing so, the resultant post filtered image retains majority of the discriminatory characteristics, thus, generating better feature vectors.

## Data Reduction

A DWFT decomposition of a  $N \times N$  image generates decomposed images of  $N \times N$ . In the unfiltered approach, all of the image elements ( $N^2$ ) are utilized in calculating the respective features. By filtering, we have focussed only on those certain areas in the decomposed image that are utilized in further calculations. If the filter only retains 50% of spatial information in the decomposed image, only  $\frac{1}{2}N^2$  of the image elements are utilized in calculating the respective features from the original unfiltered approach. In effect, by data reduction also results in a computational reduction.

## Filter Shapes

There are many possibilities of shapes and designs of the filter. The balance between spatial reduction and discrimination has to be addressed in the design. Some of the filter design has taken into account the selection criteria stated above and some have been designed void of this criteria. The designs could be endless, however, we have selected a vast array of designs which cover many of the possibilities in obtaining the optimal filters through a manual design process.

All the filters used in the experiments have been shown in Appendix C. The results of applying these filters to the texture classification method are shown in Section 5.2. Those filters which cover more of the center of the image would better localize the information, thus, providing better discriminatory information and the results support this criteria.

## 4.9 Summary

The main idea of the method is to localize and focus on the respective decomposed images to achieve better discriminatory features. We propose using filters of different shapes and sizes that maximize the neighborhood pixels in the image. This approach balances the reduced amount of information with information extracted from the data itself as features. In effect, by reducing the spatial characteristics of images that contribute to the features, the texture classification method still has the ability to preserve the classification accuracy through the classification reliability of the SVM classifier. The proposed method is tested with images from the Brodatz collection in the next chapter and classification results obtained are promising.



## Chapter 5

# Texture Classification: Comparative Experiments

### 5.1 Experimental Setup

THE data is analyzed as texture features from windows on different bands, which are contributing factors to a single element in the feature vector. The popularity of the Wavelet Transform is due to its excellent properties of the transformation such as localization of the representations and multi-resolution analysis. It allows to derive localized contributions of energy and other features to the textured signal in *well* separated frequency channels.

In this thesis, we use SVMs for texture classification, with texture features generated by the translation-invariant DWFT using the daubechies  $D_{20}$  wavelet basis. Choosing the right wavelet filters for decomposition and synthesis for texture classification has been an on going discussion in the research community with mixed and conflicting results [12]. It is also dependent on the data set used for training/testing and the final application of the texture classification method. The Daubechies family of filters has been shown to be slightly more efficient in representation and discrimination because of its analyzing function of large regularity when compared to Coiflets, Bi-orthogonal, etc [12]. In all the experiments to follow, the popular  $D_{20}$  set of filters [63, 10] has been chosen for wavelet image decomposition.

The first set of tests that have been completed is of the four types of wavelet trans-

form from the respective three authors Randen, Laine and Kuo [12, 10, 18], who have given the research community implementations of Discrete Wavelet Transform, Discrete Wavelet Packet Transform, Discrete Wavelet Frame Transform and the natural extension towards Discrete Wavelet Frame Packet Transform, DWFPT. DWFPT works under the same theory as DWPT was built upon DWT, where the details are further decomposed as well. The calculation of DWFPT is very expensive in terms of memory and time with the many convolution operations that are involved. The next set of tests consist of applying Arivazhagan's method of WSFs and WCFs on our datasets. His method is extended to incorporate translation-invariant features from DWFT. Finally, we have incorporated our method of filtering the decomposed images to better localize and generate discriminatory features.

The complete setup has been explained in the flow chart in Figure 5.3.

### 5.1.1 Data

In this experiment, we select a set of 30 natural textures (Appendix B) from the Brodatz collection [68]. Each texture image is of size 640x640 with 256 gray levels. Each image is subdivided into 400 non-overlapping 32x32 subimages. The specific images chosen from the collection are based upon the fact that their visual properties are similar over the whole image. This step was motivated by the work of Mojsilovic [33], where the samples were taken from a very small areas of interest. Her work consisted of implementing a CAD application for tissue characterization as analysis of the myocardium or cancer recognition. The implementation also made use of DWFT operators, which were insensitive to noise and image distortion, yet reliable in order to estimate texture quality from the small number of image points available.

Traditional texture classification includes texture training steps (Figure 5.1) where known texture images decomposed and corresponding features extracted. These are, then, stored in a Features Library to be utilized in the next step called texture classification step (Figure 5.2). In this step, the unknown texture image is decomposed the same way as in the first step and the same corresponding features extracted. The classifier would, then, compare this

unknown texture image to the known texture image features stored in the feature library. The *minimum distance* between the unknown texture image to the known texture image is given to the unknown texture image with respect to the classifier. The unknown texture image set generally includes the known training set.

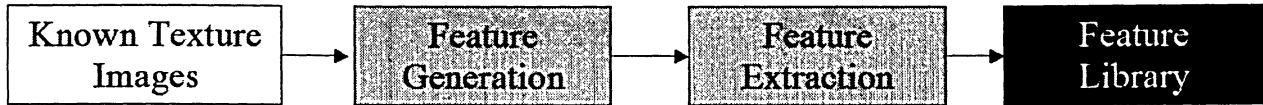


Figure 5.1: Texture Training.

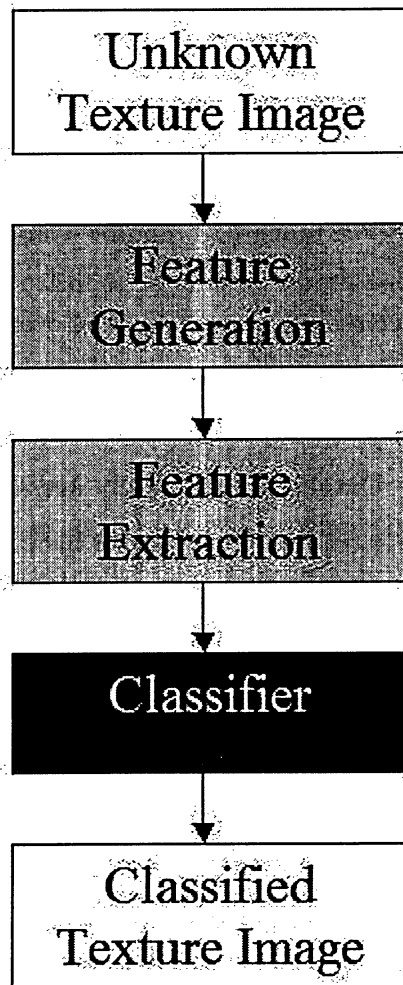
Many previous texture classification studies used overlapping training and test sets, and this is likely to yield unreliable and over-optimistic performance results. Others, used global and local information and overlapping of images to generate databases. This further increases the classification accuracy. Databases are usually averaged over many samples, and these same samples are repeated for testing. This **boosts** the classification ability to better provide results for the respective method.

Here, we use a complete separation between the training and test sets. A small fraction from 1.25% to 10% of the 400 subimages are used in training the classifier, while the rest are used for testing. Moreover, to further randomize the experiment, 10 random partitions of the training and test sets are used. The classification accuracy, average over all 30 texture classes and also over these 10 random partitions is used to evaluate the results. It is found through extensive performance testing, a training set fraction of 10% produces the best results in the texture classification method.

### 5.1.2 Extracted Features

#### Energy

The original texture image is first decomposed using DWFT. Denote the  $M \times N$  image obtained in subband  $i$  by  $I_i$ . Deciding upon the number of decompositions and sub-bands of the sample to be characterized and used in the feature vector has been well documented



**Figure 5.2:** Texture Testing or Classification.

[12, 10, 18]. The general conclusion has been that a decomposition level of between 3 through 5 has been successful in comparison experiments.

Based on the common belief that a texture can be identified by its energy distribution in the frequency domain, the local energies

$$Energy = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N I_i^2(x, y), \quad (5.1)$$

from each filtered subband image are then used as texture features. By decomposing the image to  $d$  levels, we obtain a possibility of feature vector of length  $4d$ ; approximation and three detail: vertical, horizontal and diagonal. From performance testing it is found that the optimal components are derived from the details and the last approximation component. This leads to the feature vector length of  $3d + 1$ .

### Statistical and Co-Occurrence

The feature extraction is extended to incorporate statistical features such as mean and standard deviation, which are extracted from the approximation and detail regions of DWFT decomposed images at different scales. The various combinations of the above statistical features are applied for texture classification.

In order to improve success rate of classification, we have extended Arivazhagan's [64] method. In his original method, the statistical information (Equation A.1 and A.2) was extracted, for three-level DWT decomposition of textured images. Furthermore, co-occurrence matrices  $C(i, j)$  were generated for the original textured image and the one-level DWT decomposition of the textured image. The corresponding features (Equations A.3 to A.8) were extracted from the co-occurrence matrix, computed for different angles (i.e. -  $\theta = 0^\circ, 45^\circ, 90^\circ$  and  $135^\circ$ ) and are averaged. The statistical information were termed wavelet statistical features (WSFs) and the co-occurrence matrix information were termed wavelet co-occurrence features (WCFs). The feature vector per textured sample image consisted of 24 WSFs and 35 WCFs.

In the extended version of this method, a five-level DWFT decomposition is used to decompose the texture image to generate the details (LHk, HLk and HHk for  $k=1,2,3,4,5$ )

and the fifth level approximation, LL5. Mean (A.1) and Standard Deviation (A.2) are extracted from these images. WCFs such as contrast (A.3), energy (A.4), entropy (A.5), local homogeneity (A.6), cluster shade (A.7) and maximum probability (A.8) are derived from the co-occurrence matrices,  $C(i, j)$ , computed for different angles (ie -  $\theta = 0^\circ, 45^\circ, 90^\circ$  and  $135^\circ$ ) and are averaged. The total feature vector per sample texture image consists of 32 WSFs and 96 WCFs for a total of 128 elements.

### Filtered Decomposed Images

Our filtering method is built upon the feature extraction methods of Energy and WSFs & WCFs. These methods incorporate the full use of the spatial elements of the decomposed images. The filtering method localizes and focusses on the respective decomposed images as shown in Section 4.8. This reduction in spatial information is dependent on the filter used, as listed in Appendix C. The filter is used on the full set of decomposed images from the five-level DWFT decomposition. Then, the corresponding features are extracted.

#### 5.1.3 Classification

Typically, the number of texture classes is greater than two. Thus, unlike the binary classification problem discussed in Chapter 3, we have a multi-class classification problem. Here, we adopt the conventional approach of casting this as a number of binary classification problems. In other words, we have one classifier for each texture class, each attempting to separate samples belonging to this class from samples belonging to the other classes. On classifying a new sample, the classifier with the largest output is selected as the winner, and this new sample is assigned to winner's corresponding texture class.

In the experiments, the Gaussian kernel, shown in Table 3.1, is used in the SVM. Preliminary results suggest that the Gaussian kernel outperforms the polynomial kernel. However, one still has to determine a proper values for the kernel parameter  $\sigma$  and the parameter  $C$ . Typically, this is tuned manually. A more disciplined approach is to use a validation set, or by data-resampling techniques such as cross-validation and bootstrapping. However, these methods can be very expensive in terms of computation time and/or training data.

Alternatively, one can utilize an upper bound on the generalization error predicted by the theory of SRM [22]. Li et al. [62] has used a Fused ensemble classifier approach. Three SVM classifiers were implemented with three values of  $\sigma$ . This increases the final classification complexity by three times and no attempt was made for an optimal parameter selection for the classifiers. Comparison of the Gaussian and Polynomial kernels are shown in Table 5.6, in keeping with the same procedure as explained for Table 5.3. The optimal parameters were searched manually through training. For the polynomial kernel, there are two parameters that have to be searched, the  $\theta$  value and the degree ( $d$ ) of the polynomial. In the gaussian kernel, only the value of  $\sigma$  needs to be searched. This clearly shows the gaussian kernel outperforming the polynomial kernel.

For simplicity, we have used the default value of 1000 for the parameter  $C$  and have conducted a optimal parameter selection and set the  $\sigma$  parameter to 0.001 for the Gaussian Kernel. Moreover, all the features are normalized to the range 0-1.

We have also shown in Table 5.4, 5.5 and 5.6, of the different classifiers working under the same procedure as Table 5.1, 5.2 and 5.3. It shows the robust capability of the SVM classifier over other traditional classifiers. When the decomposition levels increases from 4 to 5, the performance of the Bayes classifier [2] and LVQ [3] do not improve much or in some cases deteriorates. The SVM classifier shows a marked improvement and are less affected by the curse of dimensionality.

## 5.2 Results

### Energy

By comparing the previous decomposition methods used by the authors [18, 17, 10], we have sought out statistical establishment for the choice for DWFT by applying simple texture description analysis and applying the spectral properties or texture energies as features on small sized texture windows with the SVM classifier as stated in Section 5.1.

Performance of the classifier using 10% fraction of training samples, with 3-5 levels of wavelet decomposition are shown in Tables 5.1, 5.2 and 5.3, respectively. If the decomposition

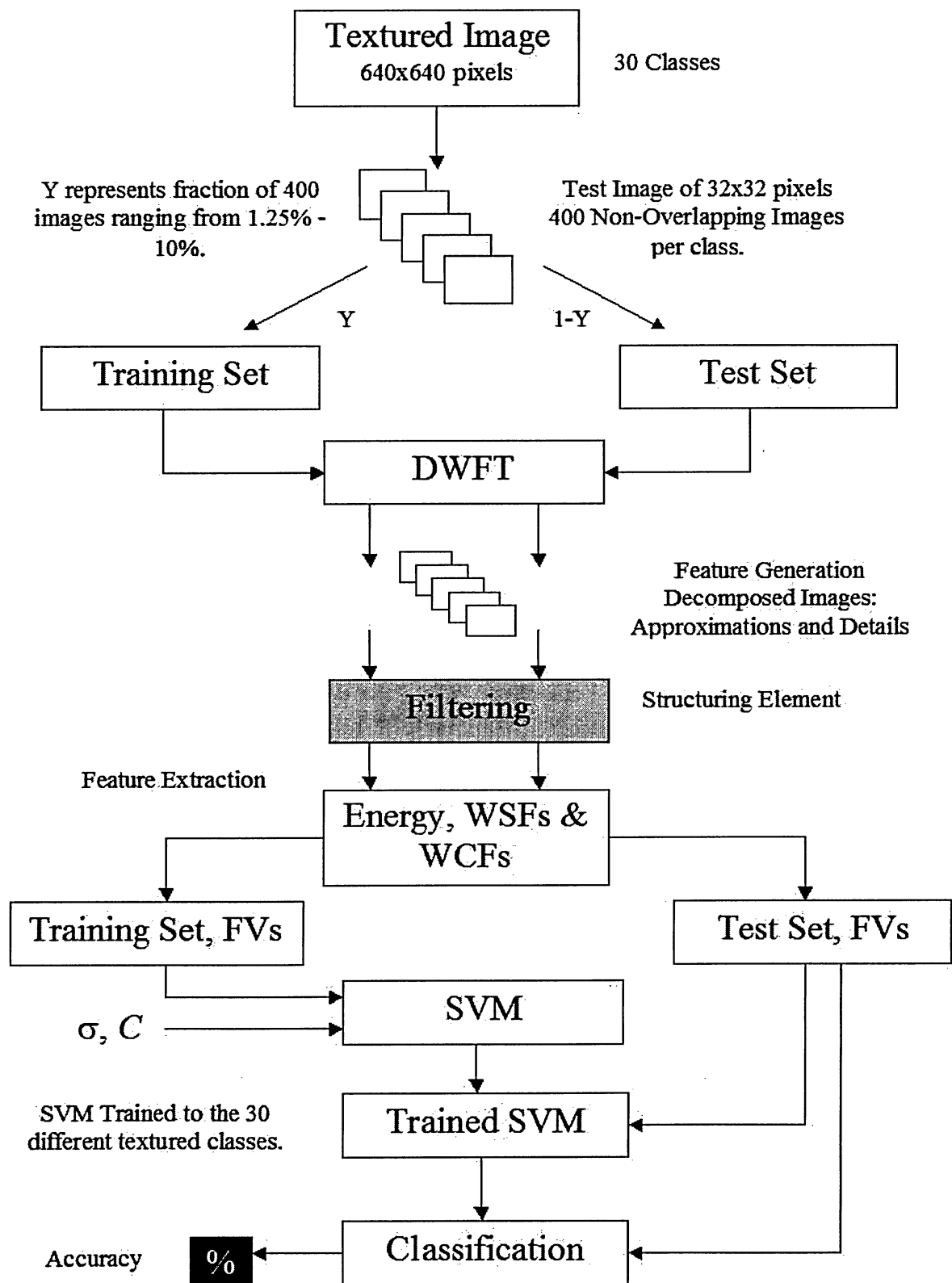


Figure 5.3: The complete texture classification implementation.



levels increase from 3 to 5, there is a more marked improvement. This demonstrates that SVMs are less affected by dimensionality and are able to assemble useful information from the additional features. Traditional classifier's performance would not improve much or in some cases even deteriorate [62, 63], which is true in traditional pattern recognition techniques, that increasing the number of texture features may not be beneficial.

From results shown in Table 5.3, the DWFT has outperformed DWFPT. The feature extracted from DWFT still maintained the discriminatory ability from the other textures. DWFPT produced vast amount information that to be dealt with. By including the detail decompositions to the fifth-level, it produced a possible feature vector of 1364 elements. It is generalized that this feature vector had overlapped in the feature dimension space that SVM was not able to overcome, which caused the lag in the classification accuracy.

**Table 5.1:** Average Classification Accuracy with the four different types of decomposition types with three decomposition levels.

Decomposition Type	Classification Correct (%)
Discrete Wavelet Transform	41.43
Discrete Wavelet Packet Transform	54.27
Discrete Wavelet Frame Transform	65.40
Discrete Wavelet Frame Packet Transform	67.33

**Table 5.2:** Average Classification Accuracy with the four different types of decomposition types with four decomposition levels.

Decomposition Type	Classification Correct (%)
Discrete Wavelet Transform	47.01
Discrete Wavelet Packet Transform	63.76
Discrete Wavelet Frame Transform	75.93
Discrete Wavelet Frame Packet Transform	83.77

**Table 5.3:** Average Classification Accuracy with the four different types of decomposition types with five decomposition levels.

Decomposition Type	Classification Correct (%)
Discrete Wavelet Transform	52.34
Discrete Wavelet Packet Transform	74.22
Discrete Wavelet Frame Transform	<b>88.40</b>
Discrete Wavelet Frame Packet Transform	85.53

**Table 5.4:** Different Classifier comparisons with different SVM kernels of Gaussian and Polynomial with their optimal parameters searched. Average Classification Accuracy with three decomposition levels of DWFT.

Classifier	Classification Correct (%)
SVM (Gaussian $\sigma = 0.001$ )	65.40
SVM (Polynomial $\theta = 1, d = 20$ )	62.11
Bayes Distance	57.83
LVQ	57.81

**Table 5.5:** Different Classifier comparisons with different SVM kernels of Gaussian and Polynomial with their optimal parameters searched. Average Classification Accuracy with four decomposition levels of DWFT.

Classifier	Classification Correct (%)
SVM (Gaussian $\sigma = 0.001$ )	75.93
SVM (Polynomial $\theta = 1, d = 20$ )	74.80
Bayes Distance	72.81
LVQ	71.91

**Table 5.6:** Different Classifier comparisons with different SVM kernels of Gaussian and Polynomial with their optimal parameters searched. Average Classification Accuracy with five decomposition levels of DWFT.

Classifier	Classification Correct (%)
SVM (Gaussian $\sigma = 0.001$ )	88.40
SVM (Polynomial $\theta = 1, d = 20$ )	82.22
Bayes Distance	78.15
LVQ	70.91

## WSFs and WCFs

By applying the Arizvahagan’s method and our extension of the method explained in Section 4.6, it is clear that by making use of translation-invariant features, it not only gives the texture better discriminatingly, but also we are able to generate more contributing elements in the feature vector. The results are broken on a per texture basis in Table 5.7. In certain cases, Arivazhagan’s method does match or has better classification ability of certain textures (i.e. - D17, D32, D47, D56 and D85). This is due to the possible overlap in the feature space caused by the feature extraction method, where the SVM was not able to distinguish the different textures in the test samples. However, as a complete texture classification task, our extension has a definite advantage over the previous method with an overall classification ability of 92.62% over the 80.14% of the previous implementation.

## Filtering

Most of the results have been summarized in Appendix C along with the corresponding filter shapes. The most important significant results have been brought to attention and summarized in this section.

The filters localize and focus on the decomposed images. These filters reduce the data by an amount prescribed by the filter’s non-active regions. Only those elements in the decomposed images are furthered processed for feature extraction that correspond to active regions in the filter. Depending on the filter, the active ranges are somewhere between 40%

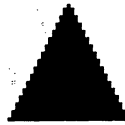
**Table 5.7:** Comparison of Arivazhagan (DWT) method versus the extension method (DWFT). Correct Classification Accuracy is stated.

Texture	DWT (%)	DWFT (%)
D1	75.00	98.44
D101	72.50	99.75
D104	98.61	99.53
D11	57.50	94.25
D16	99.72	100.00
D17	95.83	94.28
D20	100.00	100.00
D21	100.00	100.00
D24	78.05	95.17
D28	53.05	64.39
D29	56.38	87.83
D3	66.38	91.11
D32	99.44	98.67
D34	81.11	99.67
D35	60.27	93.47
D49	73.61	100.00
D47	89.00	85.36
D51	92.50	97.25
D53	98.33	100.00
D55	96.67	96.86
D56	76.11	73.31
D6	59.72	100.00
D65	73.05	75.08
D52	83.61	99.00
D84	58.33	88.75
D78	66.38	100.00
D82	91.11	92.22
D85	89.44	85.67
D46	84.38	87.22
D57	75.43	84.33
Overall	80.14	92.62

to 80%.

The first significant results that was concluded upon was that, if the threshold broke the 40% level, the texture classification method was unable to overcome this drop in spatial information, and neither the feature extraction method or the SVM classifier was able to surmount and provide respectable results. The classification results for these filters were below the 50% level.

Another interesting result was the fact that those filters which were localized at the center of the decomposed image and utilized the selection criteria, resulted in higher classification accuracy. This is shown by providing contrasted filters with the original ones, as shown in Figure 5.4 and 5.5. Their relative classification accuracy is shown in Table 5.8. This fact is also shown in Table 5.9, where contrasted diamond filters were used and in Table 5.10, where contrasted circle filter were utilized. This is also true for other filters whose spatial characteristic are not centered based. The Hour Glass filters, shown in Figure 5.10 and 5.11 provide evidence of this fact with their respective comparison in Table 5.11.



**Figure 5.4:** Triangle. Black represents pixels of interest. 50% Active.



**Figure 5.5:** Triangle. Black represents pixels of interest. 50% Active.

**Table 5.8:** Correct Classification Accuracy of the two contrasted filters.

Feature Type	Figure 5.4 (%)	Figure 5.5 (%)
Energy	86.45	64.64
WSF & WCF	90.36	68.44



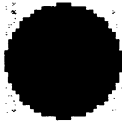
**Figure 5.6:** Inverted Diamond. 48% Active.



**Figure 5.7:** Diamond. 52% Active.

**Table 5.9:** Correct Classification Accuracy of the two contrasted filters.

Feature Type	Figure 5.7 (%)	Figure 5.6 (%)
Energy	88.40	50.46
WSF & WCF	92.55	61.36



**Figure 5.8:** Circle. 71% Active.



**Figure 5.9:** Inverted Circle. 41% Active.

**Table 5.10:** Correct Classification Accuracy of the two contrasted filters.

Feature Type	Figure 5.8 (%)	Figure 5.9 (%)
Energy	88.38	48.57
WSF & WCF	92.60	62.11



**Figure 5.10:** Hour Glass. 50% Active



**Figure 5.11:** Inverted Hour Glass. 50% Active

**Table 5.11:** Correct Classification Accuracy of the two contrasted filters.

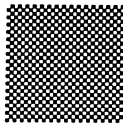
Feature Type	Figure 5.10 (%)	Figure 5.11 (%)
Energy	70.45	69.96
WSF & WCF	71.56	69.44



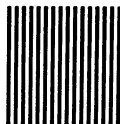
**Figure 5.12:** Cross Template. 55% Active.



**Figure 5.13:** Dual Triangle. 77% Active.



**Figure 5.14:** Checkerboard. 50% Active.



**Figure 5.15:** Vertical Lines. 50% Active.



**Figure 5.16:** Horizontal Lines. 50% Active.

The tested filters were meant to cover many different spatial arrangements of the image as possible, while still maintaining spatial integrity. They each represented and focussed on the different regions of the image. Some were focussed on the center of the image, the corners, the left/right side, top/bottom or a checkered image, where the information was gathered from all sectors of the image.

The filters that were optimum, in the sense that it was able to conserve the classification accuracy of the pre-filtered image, were the: Cross (5.12), Diamond (5.7), Circle (5.8), Dual Triangle (5.13), Horizontal Lines (5.16), Vertical Lines (5.15) and the CheckerBoard (5.14) filters.

Those filters which conversed the accuracy, while having the data reduction of close to 50% were the: Diamond (5.7), Horizontal Lines (5.16), Vertical Lines (5.15) and the CheckerBoard (5.14) filters. Their results are summarized in Table 5.12.

**Table 5.12:** Correct Classification Accuracy of the optimum filters.

Feature	Non-Filter	Figure 5.7	Figure 5.14	Figure 5.15	Figure 5.16
Energy	88.40 %	88.40 %	88.32 %	88.30 %	88.31 %
WSF & WCF	92.62 %	92.55 %	92.62 %	92.44 %	92.45 %

From Table 5.12, the CheckerBoard filter (5.14) proves to have the best filtering effect on the decomposed images and to the overall classification accuracy. The CheckerBoard filter only utilized the 50% of the spatial supplied by the decomposed image, however, it was still able to match the accuracy of the non-filtered decomposed images.



### 5.3 Multi-Texture Classification

In this work, we have employed the DWFT for a single-texture classification problem with a total of 30 textures. This is not applicable to multi-texture classification problem, where there are more than one texture classes in a single image. To accomplish this task, a five level DWFT decomposition is used to generate feature images from the subbands. A local energy function is computed for each pixel of the feature images. This yields a feature vector which contains the major properties of each of the texture classes. Explained in Section 5.1.3, known feature vectors are used to train the SVM classifier which is ultimately used to label unknown samples. The training subset are taken from the original texture class images. We have experimented with 2-class, 4-class and 8-class problems shown in Figure 5.17 to 5.22. The results show that the misclassification error grows with the increase in the number of classes.

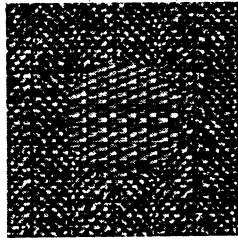
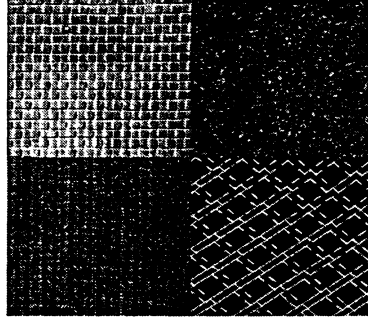


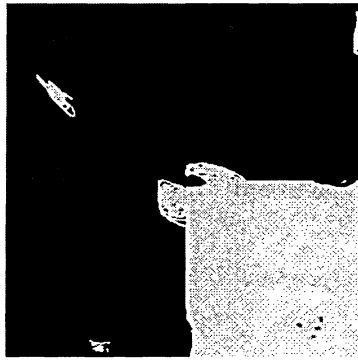
Figure 5.17: 2-class multi-texture image.



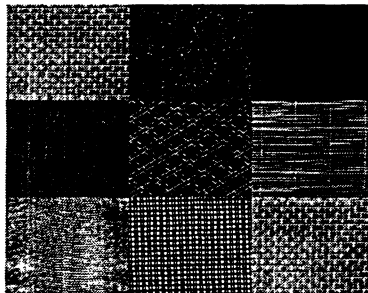
Figure 5.18: Misclassification Error Rate of 7.80%.



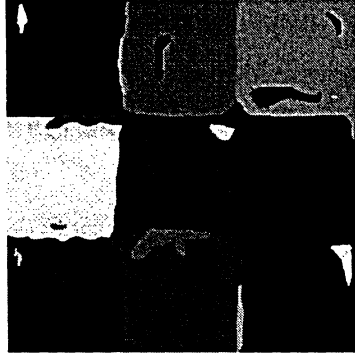
**Figure 5.19:** 4-class multi-texture image.



**Figure 5.20:** Misclassification Error Rate of 12.44%.



**Figure 5.21:** 8-class multi-texture image.



**Figure 5.22:** Misclassification Error Rate of 16.71%.

## 5.4 Summary

We have shown a general progression from the different types of wavelet decompositions and provided their respective ability to classify small-sized windows taken from 30 different texture types from the Brodatz collection with the SVM classifier. It was shown that DWFT was best able to balance computational complexity and classification accuracy in the texture classification method. Furthermore, we improved the classification by implementing the use of WSFs and WCFs. Finally, we attempted to better localize the decomposed images for better discrimination between the textures by the use of filters that capture only those elements from the image that are active in the filter. The results are comparable to the original performance when certain filters are used. The full summary of classification results is given in Appendix C.

# Chapter 6

## Conclusions

WITH increasing popularity of digital libraries, multimedia databases, texture analysis has become a focus area of research since image regions can be described by their textural properties. It can also be extremely useful in numerous other related areas: content-based image retrieval, remote sensing, image segmentation and interpretation, object recognition, and industrial applications.

### 6.1 Contributions of This Work

The contributions of this thesis are:

- Implementing the standardized testing procedure for different decomposition methods operating on small windows of textures instead of global ones and providing an unboosted classification accuracy by separating the training and testing sets.
- Extending Arivazhagan's algorithm from utilizing DWT features to shift-invariant DWFT features for generating WSFs and WCFs. Ultimately, increasing the feature space of the feature vector for better characterization of the texture classes.
- Proposing a method for optimally representing the filter banks of the DWFT decomposed images by utilizing specially selected structuring elements.

The standardized comparison of the different wavelet transform image decompositions has been conducted. The test consisted of utilizing non-overlapping small-sized windows

from the original texture space as the sample set taken from the Brodatz collection. This sample set was further fully separated in fraction of training set and testing set. Once again, no overlap was done either. In the testing, the fraction of training and testing set that produced the best results was 10%.

The first set of tests was based upon the spectral features (energy) being extracted from the frequency domain. DWFT was the clear winner because of its ability to generate higher content in the feature space of the decomposed elements because the lack of critical sampling. It also has the advantage of balancing complexity and generating a high classification accuracy.

With this conclusion, we applied and extended the DWFT technique to feature extraction method implemented by Arizvahagan. With the non-restriction of spatial issues in the lower decomposed levels, we were able to generate more features to characterize the texture. This increase in the dimension was a non-factor because of the usage of the SVM classifier and its learning algorithm that has been proven to outperform other texture classification algorithms which are based on supervised learning i.e. - neural networks. This extension was able to surpass the previous method by a margin of 12%.

A method of filtering the wavelet transform filter banks was proposed to better localize the decomposed images, by reducing the data and providing better discrimination ability. This filtering method was incorporated into the extension method stated above. The same features of WSFs and WCFs were extracted, thus producing the same feature dimension. Many different filter types were tested, each with its reduction in the spatial coordinates, and each corresponding to different areas of images and/or combinations. The conclusion was that those filters which covered more of the center of the image were able to achieve higher classification rates than others. This is in accordance with the selection criteria defined in Section 4.8. This method yielded good classification quality when compared to the filters that did not meet this criteria. The filtering approach is optimal with respect to representation and not optimal with discrimination, where the final results were comparable to the non-filtered implementation.

## 6.2 Conclusions of This Work

The current work has focused on algorithmic development and experimental justification. We have approached the general methodology of texture classification and attempted to make genuine results. Making a clear separation of training and test set has been a first step as reported here. In addition, we have clearly separated the images in the image space, such that overlapping does not occur, thus, decreases the homogeneity of the feature extraction stage and generating similar feature vectors.

Increasing the feature space of the textures with the use of translation-invariant features from the DWFT and increasing the feature dimension with the use of external features has produced a high-level of classification accuracy. The use of filters has concluded that by retaining the majority of the pixels in the center region of the decomposed images, it is possible to retain the classification ability of the original method. This method reduces the data by localizing on the image with respect to the filter. Several filters approaches are evaluated and certain optimal filter design do, in these cases, yield features that are comparable to the non-filtered decomposed images. Furthermore, considerable reductions in computations are obtained by many of the filter approaches.

## 6.3 Future Research

Here in this experiment, relatively simple subband energies and co-occurrence matrices obtained from DWFT have been used as texture features. One can use more complicated texture features, including using wavelet transforms that are not only translation-invariant, but are also rotation and scale-invariant, with applications to content-based image retrieval.

Having to search the optimal parameter for SVM can become quite tedious. Ensemble Classifiers can be a vital component because of the discriminating capabilities of different methods combined into a single result. A combination of such methods would reduce the risk of failures stemming from a specific technique. The fused SVM approach does alleviate this problem. However, one would need a more proactive approach in determining the proper

values. Moreover, there are many possible ways of conducting the final result of a fused SVM. Voting scheme are just one possible way to combine multiple classifiers in general. Other possibilities such as weighted voting, bagging and mixture of experts may also be investigated.

This thesis has employed the DWFT and work has been presented for a single-texture classification problem with a total of 30 texture classes. The technique for feature extraction is computed by energy distribution limits and co-occurrence matrix features to only a detailed single-texture case and shown on a simple application to the multi-texture classification problem (where there are more than one texture classes in a single image). A detailed study of this method to texture segmentation should be the next investigation.

We are dealing with large training and testing sets with the SVM classifier, it generates many support vectors, which cause the complexity of the final classifier to increase. In doing so, the calculations and time involved in solving a test set increases exponentially. There are reductional and optimization methods that incorporate external algorithms on the data set, that reduce the final support vectors to be used on the SVM classifier [27]. In doing so, it improves the time and accuracy of training and testing, and the overall classification ability of the classifier.

The problem of specifying regions of interest in an image is a complex issue. The method by which one specifies regions of interest for the filters and how this method can be automated is a major research area to be followed later. The actual performance of different strategies for region of interest selection is to be investigated in real world application such as biomedical image processing, which deals with abnormality detection.

SVM's mathematical fundamentals and its inherent nature is designed basically for two-class classification problems, a binary classification problem. This is why SVM is potentially a good and natural choice for several different areas in medical imaging, where the majority of the problems in this area deal with abnormality detection, as in the case of Breast Cancer in a Mammogram. The SVM classifier would generally classify areas of image as Malignant (cancerous regions) or Benign (non-cancerous regions). With the promising classification

results of the texture analysis method defined in this thesis, porting it to the biomedical applications is another research area to be followed later.



# Bibliography

- [1] M. Tuceryan, A.K. Jain, *Texture Segmentation using Voronoi polygons*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 12, pp. 211-216, 1990.
- [2] S. Theodoridis, *Pattern Recognition*, Academic Press, Toronto, 1999.
- [3] C. G. Looney, *Pattern Recognition using Neural Networks*, Oxford University Press, New York, 1997.
- [4] R. Haralick, K. Shanmugam and I. Dinstein, *Textural features for image classification*, IEEE Trans. Systems Man Cybernet., Vol. 3, pp. 610-621, 1973.
- [5] M. Tuceryan and A.K. Jain, *Texture Analysis: Handbook of Pattern Recognition and Computer Vision*, World Scientific, Singapore, 1993.
- [6] R. Chellappa and S. Chatterjee, *Classification of textures using Gaussian Markov random fields*, IEEE Trans. Acoust. Speech Signal Process, Vol. 33, pp. 959-963, 1985.
- [7] J.M. Keller, S. Chen and R.M. Crownover, *Texture description and segmentation through fractal geometry*, Computer Vision Graphical Image Process, Vol. 45, pp. 150-166, 1989.
- [8] J. Mao and A.K. Jain, *Texture Classification and Segmentation using multiresolution simultaneous autoregressive models*, Pattern Recognition, Vol. 25, No. 2, pp. 173-188, 1992.
- [9] A.K. Jain and F. Farrokhnia, *Unsupervised texture segmentation using Gabor filters*, Pattern Recognition, Vol. 24, No. 12, pp. 1167-1186, 1991.

- [10] A. Laine and J. Fan, *Texture Classificatino by Wavelet Packet Signatures*, IEEE Transactions on Pattern and Machine Intelligence, Vol. 15, No. 11, pp. 115-130, November 1993.
- [11] S.G. Mallat, *A Wavelet Tour of Signal Processing*, Second Edition, Academic Press, San Diego, 1999.
- [12] T. Randen and J.H. Husoy, *Filtering for texture classification: a comparative study*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, pp. 680-698, 1999.
- [13] J. Strand and T. Taxt, *Local Frequency features for texture classification*, Pattern Recognition, Vol. 27, No. 10, pp. 1397-1406, 1994.
- [14] J.S. Weszka, C.R. Dyer and A. Rosenfeld, *A Comparative Study of Texture measures for terrain classification*, IEEE Trans. Systems Man Cybernet, Vol. 6, pp. 269-285, 1976.
- [15] S.G. Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Trans. Pattern Anal. Mach. Intell. Vol. 11, No. 7, pp. 674-693, 1989.
- [16] S Mallat and Z Zhang. *Matching pursuit with time-frequency dictionaries*, IEEE Transactions on Acoustics, Vol. 7, pp. 254-268, 1993.
- [17] M. Unser, *Texture Classification and Segmentation using Wavelet Frames*, IEEE Transactions on Image Processing, Vol. 4, No. 11, pp. 112-124, November 1995.
- [18] T. Chang and C.C.J. Kuo, *Texture Analysis and Classification with Tree-Structured Wavelet Transform*, IEEE Trans. Image Process, Vol. 2, No. 4, pp. 429-441, 1993.
- [19] N.D. Kim and S. Upda, *Texture Classification using rotated wavelet filters*, IEEE Trans. Systems Man Cybernet, Part A: Systems Hum., Vol. 30, No. 6, pp. 847-852, 2000.
- [20] Y. Chitre and A.P. Dhawan, *M-Band wavelet discrimination of natural textures*, Pattern Recognition, Vol. 32, No. 5, pp. 773-789, 1999.

- [21] D.A. Clausi and M.E. Jernigan, *Designing Gabor Filters for optimal texture separability*, Pattern Recognition, Vol. 33, No. 11, pp. 1835-1849, 2000.
- [22] V. Vapnick, *The Nature Statistical Learning Theory*, Springer, New York, 1995.
- [23] V. Vapnick, *Statistical Learning Theory*, John Wiley and Sons, New York, 1998.
- [24] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Nijogi, T. Poggio, V. Vapnick, *Comparing Support Vector Machines with Gaussian Kernels to radial basis function classifiers*, IEEE Trans. Signal Process, 45, No. 11, 2758-2765, 1996.
- [25] S. Dumais, *Using SVMs for text categorization*, IEEE Intell. Systems, 13, No. 4, 21-23, 1998.
- [26] H. Drucker, D. Wu and V. Vapnick, *Support Vector Machines for spam categorization*, IEEE Trans Neural Networks, Vol. 10, No. 5, 1048-1054, 1999.
- [27] A.J. Smola and B. Scholkopf, *A tutorial on Support Vector Regression*, NeuroCOLT2 Technical Report NC2-TR, Royal Holloway College, 1998.
- [28] K.R. Muller, A.J. Smola and G. Ratsch, *Predicting time series with support vector machines*, Proceedings of the International Conference on Artificial Neural Networks, Lausanne, Switzerland, Vol. 17, pp. 785-790, 1997.
- [29] S.D. Meyers, B.G. Kelly and J.J. O'Brien, *An introduction to wavelet analysis in oceanography and meteorology*, Monthly Weather Review, Vol. 12, pp. 40-54, 1993.
- [30] J.N. Bradley, C.M. Brislawn and T. Hopper, *The FBI wavelet/scalar quantization standard for gray-scale fingerprint image compression*, Proc. SPIE 2743, 1993.
- [31] C.M. Brislawn, J.N. Bradley, R.J. Onyshczak and T. Hopper, *The FBI compression standard for digitized fingerprint images*, Proc. SPIE 3516, pp. 897-904, 1996.
- [32] T. Mzaik and J.M. Jagadeesh, *Wavelet-based detection of transients in biological signals*, Proceedings of SPIE 4553, pp. 1050-1058, 2003.

- [33] A. Mojsilovic, M. Popovic, A. Neskovic and A. Popovic, *Wavelet Image Extension for Analysis and Classification of Intarcted Myocardial Tissue*, IEEE Trans. on Biomedical Engineering, Vol. 44, No. 9, pp. 1178-1191, September 1997.
- [34] M. Misti, Y. Misiti, and G. Oppenheim, *Wavelet Toolbox: For use with Matlab®*, Mathworks User's Guide, Version 3, 2004.
- [35] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, 1997.
- [36] A.V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [37] M.V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, IEEE Press; New York, 1994.
- [38] R. Coifman and V. Wickerhauser, *Entropy based algorithms for best basis selection*, IEEE Transactions on Information Theory, Vol. 32, pp. 712-718, March 1992.
- [39] I. Daubechies, *Orthonormal bases of compactly supported wavelets II: Variations on a theme*, SIAM J. Math. Anal., Vol. 24, pp. 499-519, 1993.
- [40] A. Cohen and I. Daubechies, *Orthonormal bases of compactly supported wavelets III: Better frequency localization*, SIAM J. Math. Anal., Vol. 24, pp. 520-527, 1993.
- [41] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, *Image Coding using Wavelet Transform*, IEEE Transaction of Image Processing, pp. 1150-1164, 1992.
- [42] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, 1992.
- [43] K.R. Muller, *An introduction to Kernel-Based Learning Algorithms*, IEEE Trans. on Neural Networks, Vol. 12, No. 2, pp. 436-441, March 2001.

- [44] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi and V. Vapnick, *Comparing support vector machines with gaussian kernels to radial basis function classifiers*, IEEE Trans. Signal Process, Vol. 45, No. 11, pp. 2758-2765, 1996.
- [45] A.R. Ahmad, M. Khalid and R. Yusof, *Kernel Methods and Support Vector Machines for Handwriting Recognition*, Student Conference on Research and Development Proceedings, Vol. 5, pp. 40-52, 2002.
- [46] Z. Sun, G. Bebis, X. Yuan and S.J. Louis, *Genetic Feature Subset Selection for Gender Classification: A Comparative Study*, Sixth IEEE Workshop on Applications of Computer Vision, pp. 66-78, 2002.
- [47] E. Osuna, R. Freund and F. Girosi, *Training Support Vector Machines: An Application to face detection*, Proceedings of Computer Vision and Pattern Recognition, Vol. 7, pp. 550-562, June 1997.
- [48] Y. Ma and T.F.K. Fang, *Texture Image Classification Based on Support Vector Machine and Distance Classification*, Proceedings of the 4<sup>th</sup> World Congress on Intelligent Control and Automation, pp. 346-354, June 2002.
- [49] O. Chapelle, P. Haffner and V.N. Vapnick, *Support Vector Machines for Histogram-Based Image Classification*, IEEE Trans. on Neural Networks, Vol. 10, No. 5, pp. 98-104, September 1999.
- [50] R. Reyna and M. Cattoen, *Segmenting Images with Support Vector Machines*, IEEE Trans. on Neural Networks, Vol. 11, No. 4, pp. 551-558, April 2000.
- [51] I.E. Naqa, M.N. Wernick and Robert M. Nishikawa, *A Support Vector Machine Approach for detection of Microcalcifications*, IEEE Trans. on Medical Imaging, Vol. 21, No. 12, pp. 48-64, December 2002.
- [52] A. Chodorowski and Ulf Mattsson, *Support Vector Machines for Oral Lesion Classification*, IEEE Trans. on Medical Imaging, Vol. 21, No. 2, pp. 24-32, February 2002.

- [53] S. Fukuda, R. Katagiri, and H. Hirose, *Unsupervised Approach for Polarimetric SAR Image Classification Using Support Vector Machines*, Institute of Space and Astronautical Science, Japan, Vol. 14, 2002.
- [54] X. Wang and K.K. Paliwal, *Feature Extraction and Dimensionality Reduction Algorithms and their Applications in Vowel Recognition*, Pattern Recognition Society, Vol. 36, pp. 2429-2439, 2003.
- [55] J. Sklansky, *Image Segmentation and Feature Extraction*, IEEE Trans. System Man Cybernet, Vol. 8, pp. 237-247, 1978.
- [56] K.L. Laws, *Rapid texture identification*, Proc. SPIE 238, pp. 376-380, 1980.
- [57] M. Unser, *Local Linear Transforms for texture measurements*, Signal Processing, Vol. 11, pp. 61-79, 1986.
- [58] I. Ng, T. Tan, and J. Kittler, *On local linear transform and Gabor filter representation of texture*. In Proc. Int. Conf. on Patt. Rec., Vol. 12, pp. 627-631, 1992.
- [59] J.H. Hu, Y. Wang and P.T. Cahill, *Multispectral Code Excited Linear Prediction Coding and its Application in Magnetic Resonance Images*, IEEE Transactions on Image Processing, Vol. 6, No. 11, pp. 1028-1039, November 1997.
- [60] C. Busch, *Wavelet Based Texture Segmentation of Multi-Modal Tomographic Images*, Elsevier Science Computer and Graphics, Vol. 21, No. 3, pp. 347-358, 1997.
- [61] C. Busch and M. Eberle, *Morphological operations for color-coded images*, Computer Graphics Forum, Vol. 14, pp. 193-204, 1995.
- [62] S. Li, J.T. Kwok, H. Zhu and Y. Wang, *Texture Classification using the Support Vector Machines*, Pattern Recognition Society, Vol. 36, pp. 2883-2893, 2003.
- [63] K.M. Rajpoot and N.M. Rajpoot, *Wavelets and Support Vector Machines for Texture Classification*, Thesis, Department of Computer Science, University of Warwick, UK, 2004.

- [64] S. Arivazhagan and L. Ganesan, *Texture Classification using Wavelet Transform*, Pattern Recognition Letters, Vol. 24, pp. 1513-1521, 2003.
- [65] V. Manian and R. Vasquez, *Scaled and Rotated texture classification using a class of basis function*, Pattern Recognition, Vol. 31, No. 12, pp. 1937-1948, 1998.
- [66] G. Menegaz, A. Rivoldini and J.P. Thiran, *Dyadic frames of directional wavelets as texture descriptors*, Wavelets Applications Signal Image Processing VIII, No. 4119, pp. 263-273, 2000.
- [67] C.W. Hsu, C.C. Chang and C.J. Lin, *A Practical Guide to Support Vector Classification*, A Technical Report, Department of Computer Science & Information Engineering, National Taiwan University, Taiwan, 2001.
- [68] P. Brodatz, *Textures: A Photographic Album for Artists and Desginers*, New York: Dover, 1966.
- [69] *MIT Vision and Modelling Group*,  
<http://vismod.media.mit.edu/vismod/>, 1998.

# Appendix A

## Co-Occurrence Matrix Equations

$$Mean(m) = \frac{1}{N^2} \sum_{i,j=1}^N p(i, j) \quad (A.1)$$

$$StandardDeviation(sd) = \sqrt{\frac{1}{N^2} \sum_{i,j=1}^N [p(i, j) - m]^2} \quad (A.2)$$

where  $p(i,j)$  is the transformed value in  $(i,j)$  for image of size  $N \times N$  for Equations A.1 and A.2.

$$Contrast = \sum_{i,j=1}^N (i, j)^2 C(i, j) \quad (A.3)$$

$$Energy = \sum_{i,j=1}^N C(i, j)^2 \quad (A.4)$$

$$Entropy = - \sum_{i,j=1}^N \log_2 C(i, j) \quad (A.5)$$

$$LocalHomogeneity = \sum_{i,j=1}^N \frac{1}{1 + (i - j)^2} C(i, j) \quad (A.6)$$

$$ClusterShade = \sum_{i,j=1}^N (i - M_x + j - M_y)^3 C(i, j) \quad (A.7)$$

$$MaximumProbability = \text{Max}[C(i, j)] \quad (A.8)$$



where

$$M_x = \sum_{i,j=1}^N iC(i,j) \quad (\text{A.9})$$

$$M_y = \sum_{i,j=1}^N jC(i,j) \quad (\text{A.10})$$

# Appendix B

## Brodatz Image Collection

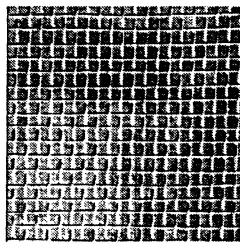


Figure B.1: D1.

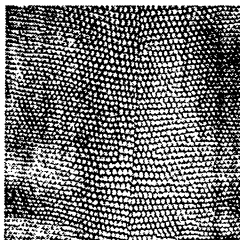


Figure B.2: D3.

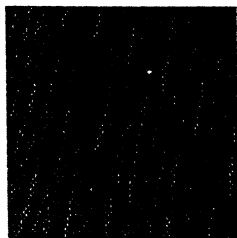


Figure B.3: D6.

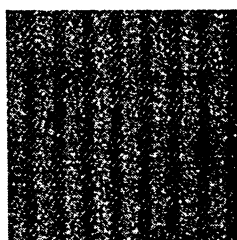


Figure B.4: D11.

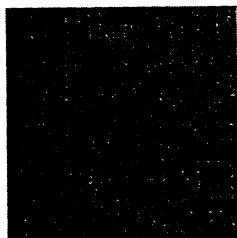


Figure B.5: D16.

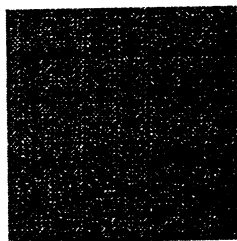
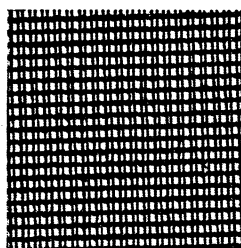
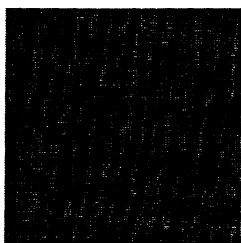


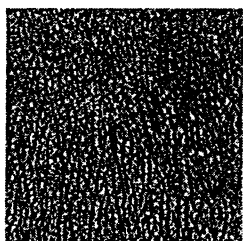
Figure B.6: D17.



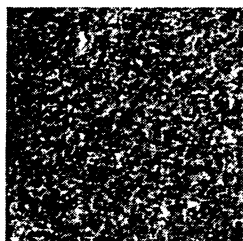
**Figure B.7: D20.**



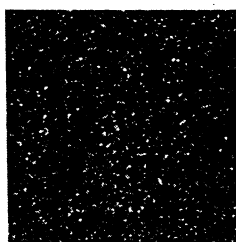
**Figure B.8: D21.**



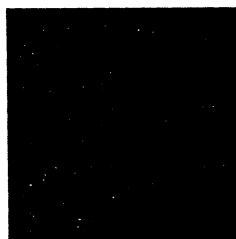
**Figure B.9: D24.**



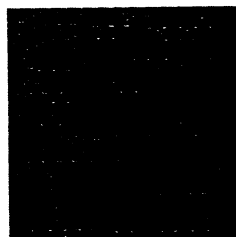
**Figure B.10: D28.**



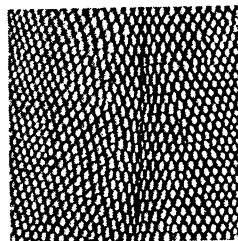
**Figure B.11: D29.**



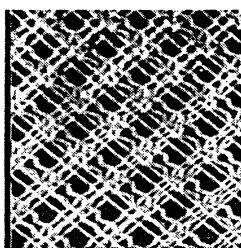
**Figure B.12: D32.**



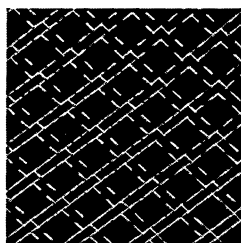
**Figure B.13: D34.**



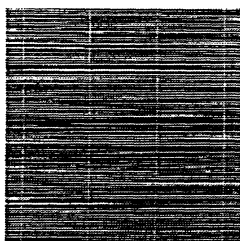
**Figure B.14: D35.**



**Figure B.15: D46.**



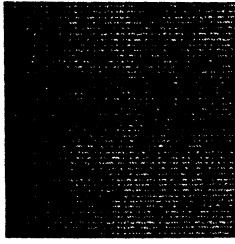
**Figure B.16: D47.**



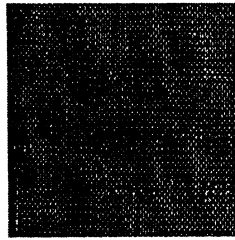
**Figure B.17: D49.**



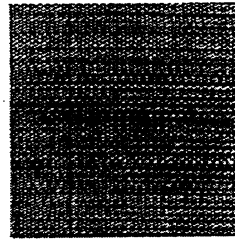
**Figure B.18: D51.**



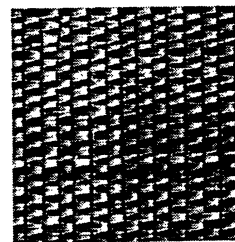
**Figure B.19: D52.**



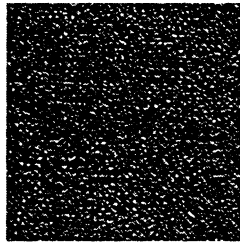
**Figure B.20: D53.**



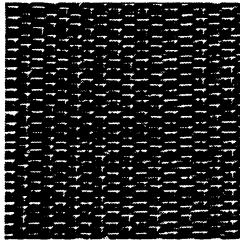
**Figure B.21: D55.**



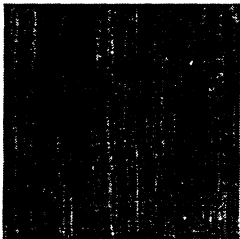
**Figure B.22: D56.**



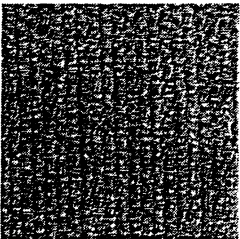
**Figure B.23: D57.**



**Figure B.24: D65.**



**Figure B.25: D78.**



**Figure B.26: D82.**



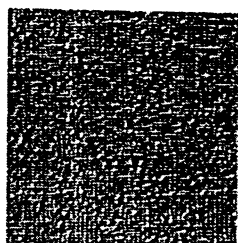


Figure B.27: D84.

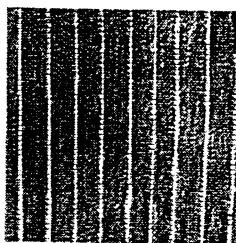


Figure B.28: D85.

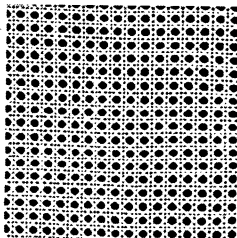


Figure B.29: D101.

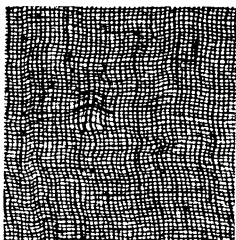


Figure B.30: D104.

# Appendix C

## Filter Shapes



**Figure C.1:** Cross Template. 55% Active.

**Table C.1:** Filter Cross Template.

Feature Type	Classification Correct (%)
Energy	88.33
WSF & WCF	91.86



**Figure C.2:** Triangle. Black represents pixels of interest. 50% Active.

**Table C.2:** Filter Triangle.

Feature Type	Classification Correct (%)
Energy	86.45
WSF & WCF	90.36



**Figure C.3:** Inverted Triangle. 50% Active.

**Table C.3:** Filter Inverted Triangle.

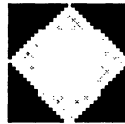
Feature Type	Classification Correct (%)
Energy	64.64
WSF & WCF	68.44



**Figure C.4:** Diamond. 52% Active.

**Table C.4:** Filter Diamond.

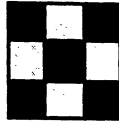
Feature Type	Classification Correct (%)
Energy	88.40
WSF & WCF	92.55



**Figure C.5:** Inverted Diamond. 48% Active.

**Table C.5:** Filter Inverted Diamond.

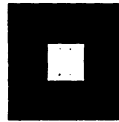
Feature Type	Classification Correct (%)
Energy	50.46
WSF & WCF	61.36



**Figure C.6:** Box Template. 65% Active.

**Table C.6:** Filter Box Template.

Feature Type	Classification Correct (%)
Energy	87.50
WSF & WCF	89.04



**Figure C.7:** Inverted Box Template. 80% Active.

**Table C.7:** Filter Inverted Box Template.

Feature Type	Classification Correct (%)
Energy	80.34
WSF & WCF	79.22



**Figure C.8:** Circle. 71% Active.

**Table C.8:** Filter Circle.

Feature Type	Classification Correct (%)
Energy	88.38
WSF & WCF	92.60



**Figure C.9:** Inverted Circle. 41% Active.

**Table C.9:** Filter Inverted Circle.

Feature Type	Classification Correct (%)
Energy	48.57
WSF & WCF	62.11



**Figure C.10: Half Triangle. 50% Active.**

**Table C.10: Filter Dual Triangle.**

Feature Type	Classification Correct (%)
Energy	79.80
WSF & WCF	80.33



**Figure C.11: Other-Half Triangle. 50% Active.**

**Table C.11: Filter Other-Half Triangle.**

Feature Type	Classification Correct (%)
Energy	79.45
WSF & WCF	81.78



**Figure C.12:** Hour Glass. 50% Active

**Table C.12:** Filter Hour Glass.

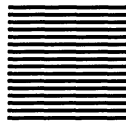
Feature Type	Classification Correct (%)
Energy	70.45
WSF & WCF	71.56



**Figure C.13:** Inverted Hour Glass. 50% Active

**Table C.13:** Filter Inverted Hour Glass.

Feature Type	Classification Correct (%)
Energy	69.96
WSF & WCF	69.44

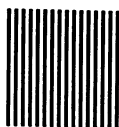


**Figure C.14:** Horizontal Lines. 50% Active.

**Table C.14:** Filter Horizontal Lines.

Feature Type	Classification Correct (%)
Energy	88.31
WSF & WCF	92.45

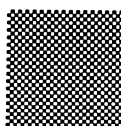




**Figure C.15:** Vertical Lines. 50% Active.

**Table C.15:** Filter Vertical Lines.

Feature Type	Classification Correct (%)
Energy	88.30
WSF & WCF	92.44



**Figure C.16:** Checkerboard. 50% Active.

**Table C.16:** Filter Checkerboard.

Feature Type	Classification Correct (%)
Energy	88.40
WSF & WCF	92.61



**Figure C.17:** Dual Hour Glass. 42% Active.

**Table C.17:** Filter Inverted Dual Hour Glass.

Feature Type	Classification Correct (%)
Energy	55.67
WSF & WCF	65.26



**Figure C.18:** Inverted Dual Hour Glass. 58% Active.

**Table C.18:** Filter Dual Hour Glass.

Feature Type	Classification Correct (%)
Energy	66.88
WSF & WCF	78.21



**Figure C.19:** Dual Triangle. 77% Active.

**Table C.19:** Filter Dual Triangle.

Feature Type	Classification Correct (%)
Energy	86.25
WSF & WCF	90.22