MULTI-SCALE TEXTURE ANALYSIS OF REMOTE SENSING IMAGES

USING

GABOR FILTER BANKS AND WAVELET TRANSFORMS

A Thesis

by

RAHUL RAVIKUMAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2008

Major Subject: Geography

MULTI-SCALE TEXTURE ANALYSIS OF REMOTE SENSING IMAGES USING GABOR FILTER BANKS AND WAVELET TRANSFORMS

A Thesis

by

RAHUL RAVIKUMAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Hongxing Liu
Committee Members,	Andrew Klein
	Jim Ji
Head of Department,	Douglas Sherman

December 2008

Major Subject: Geography

ABSTRACT

Multi-scale Texture Analysis of Remote Sensing Images Using Gabor Filter Banks and Wavelet Transforms. (December 2008)

Rahul Ravikumar, B.E.; B.Tech., College of Engineering, Guindy - Anna University, India

Chair of Advisory Committee: Dr. Hongxing Liu

Traditional remote sensing image classification has primarily relied on image spectral information and texture information was ignored or not fully utilized. Existing remote sensing software packages have very limited functionalities with respect to texture information extraction and utilization.

This research focuses on the use of multi-scale image texture analysis techniques using Gabor filter banks and Wavelet transformations. Gabor filter banks model texture as irradiance patterns in an image over a limited range of spatial frequencies and orientations. Using Gabor filters, each image texture can be differentiated with respect to its dominant spatial frequency and orientation. Wavelet transformations are useful for decomposition of an image into a set of images based on an orthonormal basis. Dyadic transformations are applied to generate a multi-scale image pyramid which can be used for texture analysis. The analysis of texture is carried out using both artificial textures and remotely sensed image corresponding to natural scenes.

This research has shown that texture can be extracted and incorporated in conventional classification algorithms to improve the accuracy of classified results. The applicability of Gabor filter banks and Wavelets is explored for classifying and segmenting remote sensing imagery for geographical applications. A qualitative and quantitative comparison between statistical texture indicators and multi-scale texture indicators has been performed. Multi-scale texture indicators derived from Gabor filter banks have been found to be very effective due to the nature of their configurability to target specific textural frequencies and orientations in an image. Wavelet transformations have been found to be effective tools in image texture analysis as they help identify the ideal scale at which texture indicators need to be measured and reduce the computation time taken to derive statistical texture indicators.

A robust set of software tools for texture analysis has been developed using the popular .NET and ArcObjects. ArcObjects has been chosen as the API of choice, as these tools can be *seamlessly integrated* into ArcGIS. This will aid further exploration of image texture analysis by the remote sensing community

DEDICATION

I dedicate this thesis to my father, Ravikumar Venkatasubramanian, and my mother Sarmistha Ravikumar. Without their patience, understanding, love and support the completion of this work could not have been possible.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my committee chair, Dr. Hongxing Liu, who guided me in every step of this research and helped enrich my research capability.

I sincerely thank my committee members, Dr. Andrew. G. Klein, Dr. Jim Ji for their insightful advice, guidance and support throughout the course of this research.

Special thanks to Dr. Lei Wang, for his advice and support in implementing statistical texture indicators.

I would like to express my gratitude to Dr. Robert Coulson, Dr. Maria Tchakerian and Dr. Andrew Birt for giving me an opportunity to do quality research at the Knowledge Engineering Lab, Texas A&M University.

Thanks also go to my friends Yige Gao, Haibin Su, Elvis Takow, Hariharan Subramanian, Hariprasad Nandakumar, Sahil Bhatia, Manish Singh, Sushanth Ramesh, Srinath Sudharshan Narasimhan, Amrish Deep Ravidas and Jagannath Panigrahy for their support and the wonderful time that we spent together at Texas A&M University.

I would also like to thank all the department faculty and staff for enriching my knowledge and making my time at Texas A&M a great experience.

Finally, thanks to my family for their constant encouragement, love and support.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Conventional algorithms for feature extraction, image classification and segmentation are far from satisfactory. Based on Gabor Filter banks and Wavelet transformations this research intends to derive multi scale image texture information to improve the efficiency and performance of existing algorithms. Conventional algorithms mainly rely on color, or spectral information of remotely sensed multispectral images for classification, segmentation and automated feature extraction. It is well known that the results from these algorithms are inferior when compared those interpreted by human experts. The question is "Why is there a marked improvement in the accuracy of classification or segmentation, where the human vision and manual interpretation is applied?" The answer to this question is that, in addition to color or spectral information, human experts also use other clues in their interpretation such as texture, shape and geometric properties, spatial pattern (arrangement) and local knowledge about the scene of the image. This observation implies that we need to look beyond spectral information and incorporate other information clues in order to improve the performance of existing algorithms. The computer vision and pattern recognition research community has addressed the use of texture information in image segmentation and classification. The concepts, methods and techniques for handling texture information developed by the

---

This thesis follows the style of *Remote Sensing of Environment*.

computer vision community are largely unknown to or not fully utilized by the remote sensing research community at present. These concepts, methods and techniques need to be transferred and applied in processing remote sensing imagery. Also, many technical issues need to be addressed when applying these concepts, methods and techniques to the processing of remote sensed imagery. This is because the texture indicators and techniques developed by the Computer Vision community are aimed to deal with indoor or artificial images. The natural scenes in remotely sensed images are much more complex, and the natural features have vastly different scales from artificial objects. Therefore, we need to validate, modify, expand and enrich the texture concepts, methods and techniques developed in computer vision community when we apply them to the processing of remotely sensed images of natural scenes.

1.1 Research Problem

This research tests multi-scale image texture analysis of remotely sensed images at different spatial resolutions to address the following research question, "How can texture be extracted and incorporated in conventional remote sensing algorithms for improving the accuracy of classification, segmentation and feature extraction?"

1.2 Research Objectives

To address the stated research question, the following objectives are pursued:

a. Compare and evaluate conventional texture indicators, including gray level co-occurrence matrices, laws masks and energy indicators.

b. Perform multi scale image texture analysis using the Gabor Filter Banks and wavelet transformations (Harr and Daubechies Wavelets).

c. Implement the above techniques as a software tool and apply it on to different image scenes with varying spatial resolutions and conduct comprehensive comparisons of these techniques.

1.3 Methods for Accomplishing Research Objectives

1.3.1 Evaluation of Conventional Texture Indicators

In order to look beyond spectral information, the first is to determine which set of texture indicators are best suited for the analysis of texture in the case of remotely sensed imagery. For this, various texture indicators like gray level co-occurrence matrices, Laws masks and energy indicators are evaluated.

Evaluation of gray level co-occurrence matrices involves the construction of the gray level co-occurrence matrix which consists of information about occurrences of image pixels that have similar grey levels. This matrix is normalized to derive conditional probabilities and various statistical indicators of texture including contrast, variance, correlation, entropy and inverse difference moment are derived from the matrix. Evaluation of Laws masks involves determining which spatial convolution masks are best suited for derivation of texture based indicators. For evaluating energy indicators, the co-occurrence matrix for the image is derived and then an appropriate spatial convolution mask is used for the derivation of energy.

A comprehensive evaluation of conventional texture indicators is performed to determine which set of texture indicators are best suited for the extraction of texture information from remotely sensed data; using images of different spatial resolutions.

1.3.2 Multi-scale Texture Analysis Using Gabor Filter Bank and Wavelet Transforms

After evaluating various statistical texture indicators, this research examines texture analysis using the Gabor filter bank and wavelet transformations, thus addressing the issue of spatial scale. For this, multi-scale image texture analysis techniques like Gabor filter banks and wavelet transforms (Haar and Daubechies wavelets) are used.

Performing multi-scale texture analysis of an image using the Gabor filter banks and wavelet transforms requires the following:

a. Understanding and analyzing the characteristics of an image when transformed from the spatial domain to the frequency domain with Fourier transformations.
b. Understanding wavelet theory, Harr and Daubechies Wavelets for the transformation of an image into the frequency domain.
c. Understanding and using the Gabor Filter bank wavelet.
d. Deriving image pyramids in multiple resolutions, for applying various texture indicators to perform multi-scale texture analysis for an image.

e. Determining which wavelet transformations are better suited for multi-resolution analysis to produce better texture indicators and better classification accuracies.

1.3.3 Performance Evaluation with Various Remote Sensing Scenes

A comprehensive performance evaluation of the various texture analysis techniques is conducted to compare and contrast the performance of various techniques. Images of different types and corresponding to various different scenes are analyzed. Both artificial textures and natural remotely sensed images from Landsat and Quickbird corresponding to different scenes like urban residential areas, urban commercial areas, agricultural and forested areas are evaluated.

1.3.4 Software Development

Software tools capable of supporting the research experiments and enable further adoption of texture analysis techniques in remote sensing research are developed.

1.4 Outcomes

This research presents an understanding of multi-scale image texture indicators to the remote sensing community. It also determines which algorithms are best suited for multi-scale analysis of image texture and the classification accuracies are compared to traditional classification algorithms.

CHAPTER II

CURRENT STATUS OF TEXTURE RESEARCH

2.1 Introduction

Conventional algorithms make simplifying assumptions about the uniformity of gray levels in local image regions (Tuceryan and Jain 1993). Most images in the real world, however, do not exhibit uniformity in gray levels, as physical objects do not tend to reflect electromagnetic radiation uniformly. Therefore there is a need to look beyond spectral information or color. The objective of this literature review is to look at how image texture has been used for the image classification and segmentation. Several methods for texture analysis like Laws masks, gray level co-occurrence matrices and multi-scale image analysis using Gabor filter banks and wavelets have been reviewed.

Texture plays a fundamental role in classifying objects and identifying the boundaries of significant regions in a gray level image (Wechsler 1980; Reed and du Buf 1993; Baraldi, Parmiggiani, and Imga-Cnr 1995; Angelo and Haertel 2003). Although texture is clearly important in the context of image classification and segmentation, it is very difficult to define texture. One of the popular definitions in literature define texture to be generated by one of the more basic local patterns that are repeated in a periodic manner over an image region (Wechsler 1980). However, most authors agree that this definition of texture is most applicable to deterministic forms of texture like line arrays, hexagonal tilings and checker boards (Reed and du Buf 1993; Baraldi, Parmiggiani, and Imga-Cnr 1995). Texture can also be defined as a set of local statistics or other local

properties of an image that are constant, slowly varying or approximately periodic (Tuceryan and Jain 1993).

The papers by Wechsler (1979) and Reed and Buf (1993) describe the various fundamental problems in the process of texture analysis to be: (1) Given a textured region is there a way that a region can be assigned to or classified as a sample of a particular class from a given number of classes? (2) How can a textured region be described? and (3) Given a textured region how can the boundaries between major textured regions be established?

The first problem is known as the texture classification problem. This problem is usually approached by using an algorithm for extracting prominent characteristics for identification of a given texture class. These identifying characteristics become inputs to well known techniques of pattern recognition (Reed and du Buf 1993). The second problem is more conceptually complicated. This is because, one may easily find two perceptually different textures but it could still be very mathematically difficult to describe the differences between different textured regions (Wechsler 1979). The third problem is called the texture segmentation problem or the texture grouping problem.

As a result of the dichotomies in the definition of texture, there have been several approaches for analysis of texture that have evolved over a period of time (Reed and Buf 1993; Wechsler 1979). Identification of dominant characteristics of an image region is one of the basic steps in order to build mathematical models for texture analysis (Tuceryan and Jain 1993). The various approaches to this problem can be categorized into statistical approaches (Duin, Mao, and Jain ; Baraldi, Parmiggiani, and Imga-Cnr

1995; Chen, Nixon, and Thomas 1995; Van de Wouwer, Scheunders, and Van Dyck 1999; Chica-Olmo and Abarca-Hernández 2000; Wei and Bartels 2006), local linear operators (Lam and Li 1995), Fourier transforms and multi-channel filtering methods and multi-scale (wavelet based) approaches (Clark, Bovik, and Geisler 1987; Daugman 1988; Jain and Farrokhnia 1991; Dunn, Higgins, and Wakeley 1994; Dunn and Higgins 1995; Jain, Ratha, and Lakshmanan 1997; Weldon and Higgins 1999; Clausi and Ed Jernigan 2000; Idrissa and Acheroy 2002; Kyrki, Kamarainen, and Kälviäinen 2004; Arivazhagan, Ganesan, and Priyal 2006).

2.2 Statistical Approaches for Texture Analysis

The statistical approaches for texture analyze the stochastic and periodic pattern of texture in order to describe texture. The use of statistics to describe identifying characteristics for a texture class was one of the earlier approaches used for texture classification. The various mathematical models that are used include first order statistics (gray level difference method) and second order statistics (gray level co-occurrence matrices).

The first order gray level difference method estimates the probability density function for differences between subsequent pixel values (Wechsler 1979; Randen and Husoy 1999).

Second order statistics on an image, or gray level co-occurrence matrices is one of the most popular methods used for describing texture. This method looks at computing a spatial dependence probability distribution matrix (Haralick, Dinstein, and

Shanmugam 1973). This method assumes that information about image texture is adequately specified by the matrix of relative frequencies P_{ij} with the two gray cells separated by a distance *d* and an angle *alpha* occur on the image, one with the gray level *i* and the other with the gray level *j* (Haralick, Dinstein, and Shanmugam 1973). Using the gray level spatial dependence matrices various texture features like energy, entropy, contrast, homogeneity and correlation are calculated.

Although statistical approaches are quite useful to describe texture classes, they suffer from inherent limitations. First order gray level difference method only looks at estimating the probability distribution function of adjacent pixels in an image, and therefore does not work well for coarse grained textures. There are many disadvantages of using gray level co-occurrence matrices. There is no established method for choosing a displacement vector *d* and the angle *alpha.* Also, there are many texture features that can be computed from the gray level co occurrence matrices, which would mean that some form of intelligent feature selection methodology has to be used for the process of texture classification (Tuceryan and Jain 1993).

The major stumbling block in the analysis of texture using statistical methods is determining the shape and size of the area from which the textural features need to be extracted (Wechsler 1980). The subset of an image from which meaningful measurements of texture information can be carried out is referred to as a *texel* or a *tex*ture *el*ement. This brings us to the problem of addressing spatial scale. For any smooth surface there exists a scale such that when the surface is examined – it has no texture (Wechsler 1980). As the spatial resolution decreases, coarse textures become

finer. Thus a solution to the problem of texture analysis would have to determine the *optimal spatial scale* at which texture measurements are made. However, statistical methods do not address the problem of spatial scale and address texture analysis only in the spatial domain.

Algorithms which operate simultaneously in both the spatial and frequency domains are more suitable for the analysis of texture as they address the problem of spatial scale (Randen, Husoy, and Stavanger 1995; Randen and Husoy 1999). Therefore multi-channel filtering methods and multi-scale texture analysis approaches are gradually replacing statistical methods of texture analysis.

2.3 Multi-Channel Filtering Methods

Multi channel filtering methods model texture as irradiance patterns that are identified by a concentration of distinct dominant localized frequencies (Bovik, Clark, and Geisler 1990). Each texture has a characteristic dominant spatial frequency (Bovik, Clark, and Geisler 1990). In this approach textured regions are encoded into many channels, each with a very narrow spatial frequency and orientation (Bovik, Clark, and Geisler 1990). The output of each filter is a complex sub-image whose amplitude and phase envelopes describe the spatial support of the frequencies and orientations to which this channel is tuned (Bovik, Clark, and Geisler 1990).

Multi-channel filtering methods involve the use of local linear operators like Laws Masks, Fourier transforms and dyadic filter banks like the Gabor filter banks that can be used to analyze textured images in a manner that agrees with visual perception.

Laws masks involve the use of specific masks which are convolved with texture to obtain feature vectors that can be used to describe a textured region (Lam and Li 1995). The variance or the absolute value of the convolution evaluated at each pixel over a window is known as texture energy (Lam and Li 1995). Texture energy is defined as the spatial activity measure for an image (Lam and Li 1995). Laws masks have been mainly used to analyze and classify artificial textures as they are based on a specific spatial scale and rotation set. Scale dependence can be reduced by normalizing the feature vectors with the minimum moving averages occurring in the window (Lam and Li 1995). Lam and Li (1995) have used Laws masks to classify artificial textures with accuracies of over 90% based on a feature vector consisting of more than 30 different features.

To be able to address the issue of spatial scale, texture analysis is best done in the Fourier domain. However Fourier transforms by themselves are not very useful for texture analysis as they are not *localized* in the spatial domain. To address this problem, algorithms that achieve joint localizations in both the spatial and frequency domains are required. One such filter that has been used very successfully is the dyadic Gabor filter bank. Gabor elementary signals are Gaussian modulated sinusoids in the spatial domain and shifted Gaussian sinusoids in the frequency domain (Clark, Bovik, and Geisler 1987). A Gabor filtered image is a narrow band signal which is tuned to a specific

frequency and orientation. The envelope of this filter coincides with the regions containing the textural features that this filter has been tuned to (Clark, Bovik, and Geisler 1987). By using multiple filters tuned to specific dominant spatial frequencies and orientations, the identifying characteristics of dominant textural regions can be obtained and passed on to a classification or pattern recognition algorithm. Gabor filters have successfully been used for the classification artificial textures and medical images (Turner 1986; Bovik, Clark, and Geisler 1990; Jain and Farrokhnia 1991; Farrokhnia and Jain 1991; Randen, Husoy, and Stavanger 1995). Post processing Gabor filtered images with Gaussian low pass filters has also been done in order to improve the performance of Gabor filters (Clark, Bovik, and Geisler 1987).

Although multi-channel filtering techniques like Gabor filters offer very distinct advantages, very little research has been done in the use of these techniques for classifying complex natural textures occurring in satellite images and aerial photographs (remotely sensed data). A comprehensive evaluation of multi-channel filtering techniques needs to be carried out in order to better understand texture analysis with respect to remotely sensed data.

2.4 Wavelet Based Methods for Texture Analysis

Multi scale methods use wavelets for the analysis of image texture at multiple spatial resolutions. Wavelet theory is the mathematics associated with building a model for a signal, a system or a process using a set of special functions called wavelets (Myint 2001). These functions are constructed by translating and scaling a single mother

wavelet localized both in the spatial and the frequency domains (Livens et al. 1997). Texture analysis can be carried out at multiple spatial resolutions using orthonormal wavelet basis. A multi resolution wavelet decomposes a signal into a lower frequency approximation and high frequency detail information at a coarser resolution (Myint 2001). For the analysis of satellite images using a 2 dimensional wavelet transformation, rows and columns of the image are considered as signals which need to be decomposed into coarser resolutions. An image of size $2^k * 2^k$ bits can be decomposed into *k* resolutions thus constructing an image pyramid of *k* levels.

In order to decompose an input signal into multiple resolutions, a filter bank is used. A filter bank is composed of a set of low pass and high pass filters and separates the input signal into distinct frequency bands using a process called sub-band coding (Myint 2001). The major advantage of wavelet based multi resolution signal processing when compared to Gaussian or Laplacian pyramids is that there is no loss of information in the process of decomposition of the signal. Complete reconstruction of the signal is possible using an inverse wavelet transformation. The issue of spatial scale can be addressed by using wavelets to determine an *optimal spatial resolution* for the evaluation of texture measures. Once this optimal spatial resolution has been determined, gray level co-occurrence matrices or similar statistical techniques can be used for determining identifying characteristics of different textural regions.

The issue of spatial scale can be addressed by using wavelets to determine an *optimal spatial resolution* for the evaluation of texture measures. Once this optimal spatial resolution has been determined, gray level co-occurrence matrices or similar

statistical techniques can be applied to the wavelet transformed images for determining identifying characteristics of different textural regions.

2.5 Research Gaps

Algorithms which operate simultaneously in both the spatial and frequency domains are more suitable for the analysis of texture as they address the problem of *spatial scale* (Randen, Husoy, and Stavanger 1995; Randen and Husoy 1999). Therefore multi-channel filtering methods and multi-scale texture analysis approaches are gradually replacing statistical methods of texture analysis.

A survey of the articles published in four major remote sensing journals including International Journal of Remote Sensing, Photogrammetric Engineering and Remote Sensing, IEEE Transactions on Geosciences and Remote Sensing, and Remote Sensing of the Environment during 1980-2008, was performed in order to determine the most popular choices of texture indicators and strategies used for texture analysis (Fig. 2.1). The journals were categorized into groups based on the nature of texture indicators used for analysis of remotely sensed data. Elementary texture statistics represents texture analysis techniques like Sum and Difference histograms, Normalized Sum and difference histograms, Mean, Variance and other similar histogram based indicators. Complex statistical indicators represent Markov random fields, Local window based operators, Gaussian and Gamma Markov random fields, Fractal indicators, Edge density, Kriging surfaces and Moran's I index. Statistical indicators put together account for over 85% of all papers published and therefore from Fig. 2.1, it is very clear that multi

channel filtering techniques and wavelet based methods have been very sparingly used by the remote sensing community for image texture analysis.

One possible reason for this could be the lack of significant texture processing capabilities in remote sensing packages. A survey of the texture processing capabilities of popular remote sensing software packages and GIS packages is done. Most of the packages surveyed have very limited capabilities for texture analysis. Only ENVI has modules capable of processing of co-occurrence matrices, but had only very limited support. Other packages only support for first order statistical indicators.

The texture processing capabilities of major remote sensing packages are shown in Table 2.1.

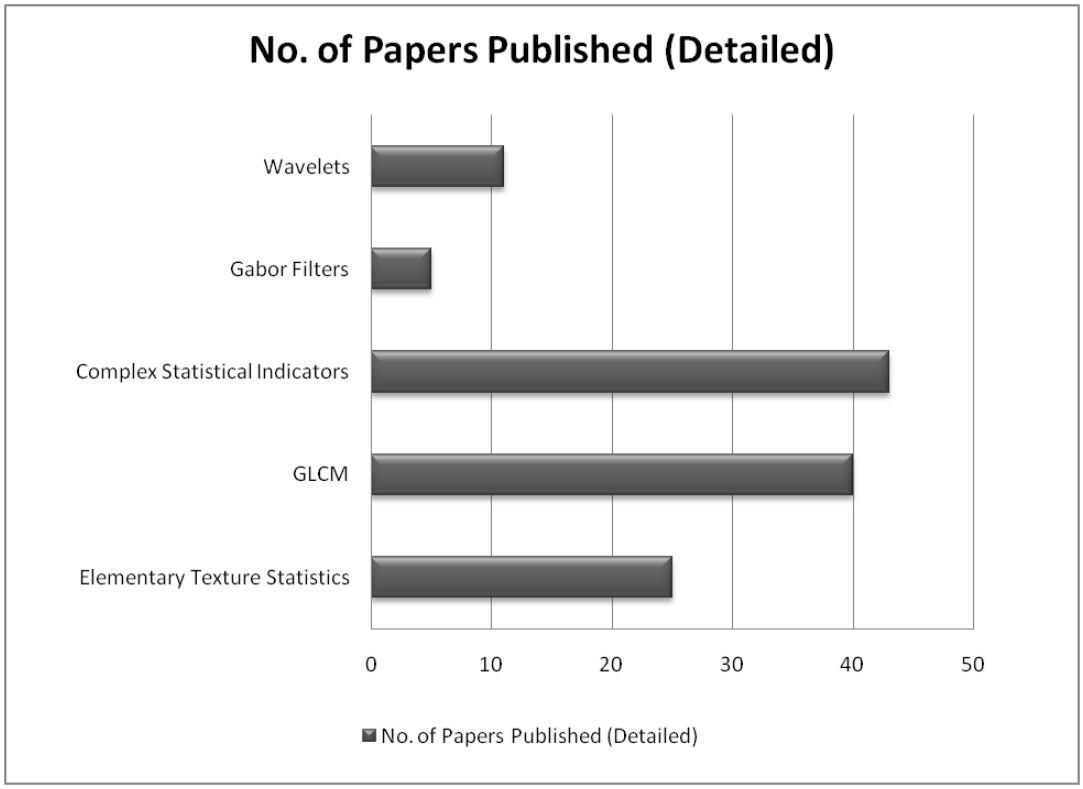


Figure 2.1 – Texture indicators used in papers published from 1980 - 2008 in IJRS, PERS, RSE and IEEE Trans. on Geosciences and RS

Table 2.1 – Texture processing capabilities of leading remote sensing and GIS packages

Software	Texture Processing Capabilities
ArcMap (ArcGIS 9.3)	First order statistical indicators. Modules for GLCM (Co-occurrence measures): **NONE** Modules for Multi-resolution texture analysis : **NONE**
ENVI (v 4.5)	First order statistical methods and local window operators. GLCM (Mean, Variance, Homogeneity, Contrast, Dissimilarity, Entropy, Angular Second Moment and Correlation). Additional GLCM Measures (Cluster Shade, Cluster Prominence etc.) : **NOT AVAILABLE** Customizing options for GLCM Statistics: **NONE** Modules for Multi-resolution texture Analysis: **NONE**
ERDAS Imagine (v 9.0)	Elementary Statistics : Mean, Variance and Std. Deviation GLCM Measures: **NONE**
Geomatica (v 10.1)	Elementary Statistics : Mean, Variance and Std. Deviation GLCM Measures: **NONE**
E-Cognition	First order texture statistics. Modules for GLCM (Co-occurrence measures): **NONE** Modules for Multi-resolution texture analysis : **NONE**

CHAPTER III

STATISTICAL TEXTURE INDICATORS AND LAWS MASKS

One of the defining quality of texture is the spatial distribution of pixel values (Tuceryan and Jain 1993). Statistical indicators have therefore been one of the oldest methods to describe texture. The aim of statistical methods of texture analysis is to characterize the stochastic properties of the spatial distribution of pixel values in an image (He and Wang 1991).

3.1 Gray Level Co-occurrence Matrices

Indicators derived from Gray level co-occurrence matrices are the most widely used texture indicators. These indicators are based on the assumption that texture information in an image (*I)* is contained in the overall or the *average* spatial relationship that gray level values in the image have with respect to each other (Haralick, Dinstein, and Shanmugam 1973). Indicators are derived from a spatial probability distribution matrix of an image block or a *texel* (texture unit). Haralick, Dinstein, and Shanmugam (1973) suggested the use of texture indicators based on this spatial probability distribution matrix and computed 14 textural features that could be used for texture analysis. These features reveal information about an image such as, homogeneity, gray tone spatial dependencies (linear structure), contrast, number and nature of boundaries present and the complexity of an image (Haralick, Dinstein, and Shanmugam 1973).

3.1.1 Mathematical Background

To describe how gray level co-occurrence matrices, let $I(x, y: 0 \leq x \leq N-1, 0 \leq y \leq N-1)$ be used to represent an image of size N * N with G gray levels. The image I with G gray levels is quantized to a G' levels, and let $G_x = \{1, 2, 3\ldots, G_x'\}$ be used to represent the horizontal spatial domain (range of pixel values) and $G_y = \{1, 2, 3\ldots, G_y'\}$ be used to represent the vertical spatial domain.

The image block used to derive Gray level co-occurrence matrices is based on the *nearest neighborhood resolution cells* (Haralick, Dinstein, and Shanmugam 1973). The neighborhood resolution cells for a pixel at $I(x, y)$ are shown in Fig. 3.1.

In this method we assume that texture information is adequately specified by a matrix of relative frequencies P_{ij} where two neighboring resolution cells that are separated by a distance d and an angle α occur in the image block, one with a gray level i and the other with a gray level j (Haralick, Dinstein, and Shanmugam 1973). These matrices are therefore a function of the angular relationship between neighboring cells as well as the distance between the cells (Haralick, Dinstein, and Shanmugam 1973).

Mathematically this relationship can be represented as:

$$P(i, j, d, 0^o) = \#\{((k, l), (m, n) \in N, \text{where } k - m = 0, |l - n| = d$$

$$P(i, j, d, 45^o) = \#\{((k, l), (m, n) \in N, \text{where } k - m = d, l - n = -d$$

$$P(i, j, d, 90^o) = \#\{((k, l), (m, n) \in N, \text{where } |k - m| = d, l - n = 0$$

$$P(i, j, d, 135^o) = \#\{((k, l), (m, n) \in N, \text{where } k - m = -d, l - n = d$$

where **#** represents the number of elements in the set, and k, l, m, n $\in$ G and $I(k, l) = i$, $I(m, n) = j$.

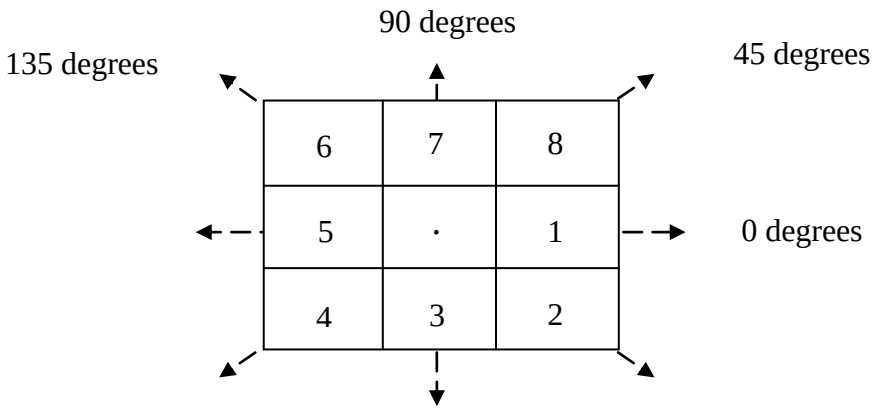


Figure 3.1 – Neighborhood resolution cells for a pixel form a texel

To illustrate the above with an example, let us consider a simple 4 * 4 image chip with 3 gray levels:

$$I(x,y) = \begin{Bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 3 & 3 & 2 & 2 \end{Bmatrix}$$

In the above example, the spatial probability distribution matrix would measure the number of times gray levels 1, 1 occur together when separated by a distance *d* and an angle α. Similarly co-occurrences for all other gray level combinations can be computed. The gray level co-occurrence matrices for the above image chip where $d = 1$ and $\alpha = 0^o$, 90^o, 45^o and 135^o respectively are shown below:

$P_{(d=1,\alpha=0^o)} =$

	0	**1**	**2**	**3**
0	4	2	1	0
1	2	4	0	0
2	1	0	6	1
3	0	0	1	2

$P_{(d=1,\alpha=90^o)} =$

	0	**1**	**2**	**3**
0	4	1	2	1
1	1	4	1	0
2	2	1	4	1
3	1	0	1	0

$P_{(d=1,\alpha=45^o)} =$

	0	1	2	3
0	2	2	2	0
1	2	2	0	0
2	2	0	2	2
3	0	0	2	0

$P_{(d=1,\alpha=135^o)} =$

	0	1	2	3
0	2	1	1	1
1	1	2	2	0
2	1	2	4	0
3	1	0	0	0

Co-occurrence matrices are symmetric in nature, i.e. *P (i, j; d, α) = P (j, i; d, α).* The number of operations required to compute a gray level co-occurrence matrix is directly proportional to the number of resolution cells *n* and the number of gray levels present in the image (Haralick, Dinstein, and Shanmugam 1973). In comparison, the number of operations to compute a Fourier transformation of an image using a fast Fourier transformation would be of the order of n *log* n.

After the neighborhood resolution cells for a pixel have been obtained, the gray tone spatial-dependence matrix (GLCM) is computed. The gray level co-occurrence matrix is then normalized by a factor *R* which is normally the sum of all the elements in the co-occurrence matrix (Haralick, Dinstein, and Shanmugam 1973). All the information required for characterizing image texture can be obtained from the gray tone spatial-dependence matrices. Therefore, all the textural descriptors are extracted from these matrices (Haralick, Dinstein, and Shanmugam 1973).

The equations defining the texture measures that can be computed from gray level co-occurrence matrices are shown below:

$$\text{Mean} = \sum_i \sum_j P(i,j) * i \tag{3.1}$$

$$\text{Variance} = \sum_i \sum_j P(i,j) * (i - \mu^2) \tag{3.2}$$

$$\text{Homogeneity} = \sum_i \sum_j \frac{P(i,j)}{1+|i-j|} \tag{3.3}$$

$$\text{Contrast} = \sum_i \sum_j P(i,j) * (i-j)^2 \tag{3.4}$$

$$\text{Entropy} = \sum_i \sum_j -P(i,j) * \log_e P(i,j) \tag{3.5}$$

$$\text{Angular Second Moment} = \sum_i \sum_j P(i,j)^2 \tag{3.6}$$

$$\text{Correlation} = \sum_i \sum_j (i - \mu_x) * (j - \mu_y) * P(i,j) \tag{3.7}$$

$$\text{Dissimilarity} = \sum_i \sum_j \frac{1}{1+(i-j)^2} * P(i,j) \tag{3.8}$$

where,

$$\mu = \frac{\sum P(i,j)}{n} \tag{3.9}$$

$$\mu_x = \frac{\sum_j P(i,j)}{n}, \mu_y = \frac{\sum_i P(i,j)}{n} \tag{3.10}$$

and *n* being the no of elements.

3.1.2 Implementation of the Software Tool

Let *I(x, y)* be used to represent a quantized image with *M* rows and *N* columns. The input parameters required for the computation of texture indicators are the distance between the neighborhood resolution cells *d*, the orientation α and the size of the rectangular neighborhoods' *n*. The general steps followed in the implementation are:

1. Read the input image *I(x, y)*.
2. For every pixel *P(i, j)* in *I(x, y)* do:
    a) Get the rectangular neighborhood for *P(i, j)* of size *n*
    b) Determine the maximum gray level (*max_value)* and minimum gray level (*min_value*) in the neighborhood.
    c) Create a GLCM of size (*max_value – min_value)*
    d) Count all the occurrences of gray levels *i, j* separated by distance *d* and an orientation α.
    e) Normalize GLCM
    f) Compute statistical indicator using GLCM: *output(i, j)*
    g) Replace *P(i, j)* with *output(i, j)*.
3. End

The detailed steps in the implementation of the algorithm are as shown in Fig. 3.2.

The computation of texture indicators based on gray level co-occurrence matrices is implemented as a software tool using Microsoft Visual C# and ESRI ArcObjects (Fig 3.3). The computational algorithm for gray level co-occurrence matrices is as follows.

```
calculate_glcm_measure(I, d, alpha){
	for(i = 0; i < M; i++){
		for(j = 0; j < N; j++){
			//Get Neighborhood resolution cells for a pixel at I(i,j)
			window = get_window(I, i, j, n)
			//Determine size of the gray level co-occurrence matrix
			min_value = get_min(window)
			max_value = get_max(window)

			//Size of the Gray level co-occurence matrix is_
			//proportional to the variation in the_
			//pixel values of the neighborhood resolution cells.
			size = max_value - min_value
			//Allocate (size + 1) * (size + 1) cells for the_
			//Gray level co-occurrence matrix.
			//(size + 1) is used to take care of special_
			//condition where (max value = min value)
			glcm_matrix = new float[size + 1] [size + 1]

			//For Window evaluate a gray level co-occurrence matrix
			//Compute Gray level co-occurrence matrix for distance
			// ‘d’ and angle 'alpha'
			glcm_matrix = calculate_glcm(window, d, alpha)
			//Detemine sum of all the values in the glcm to determine_
			//the normalizing factor.
			total_value = sum(glcm_matrix)

			//Example calculation of the indicator entropy
			//For each and every valid entry in the glcm the
			//probability function is calculated as under:
			for(k = 0; k < size + 1; k++){
				for(l = 0; l < size + 1; l++){
					// Normalizing coefficient total_value is R
					probability=glcm(j,k)/total_value
					entropy += -probability * log (probability)
				}
			}
			//Got the texture measure.
			glcm_measure(i,j) = entropy
		}
	}
}
```

Figure 3.2 – Algorithm describing implementation of GLCM indicators

```
calculate_glcm(window, d, alpha){
	//Using 'd' and 'aplha' determine the deviation in x and y
	dx = x_deviation(d,alpha);
	dy = y_deviation(d, alpha);
	for(i = 0; i < size + 1; i ++){
		for (j = 0; j < size + 1; j++){
			k = i + dx
			l = j + dy
			//Add entry for gray level pairs in the glcm_matrix
			++glcm_matrix(window(i,j), window(k,l))
		}
	}
	//Return the resultant glcm_window
	return glcm_matrix
}
```

Figure 3.2 (Continued)

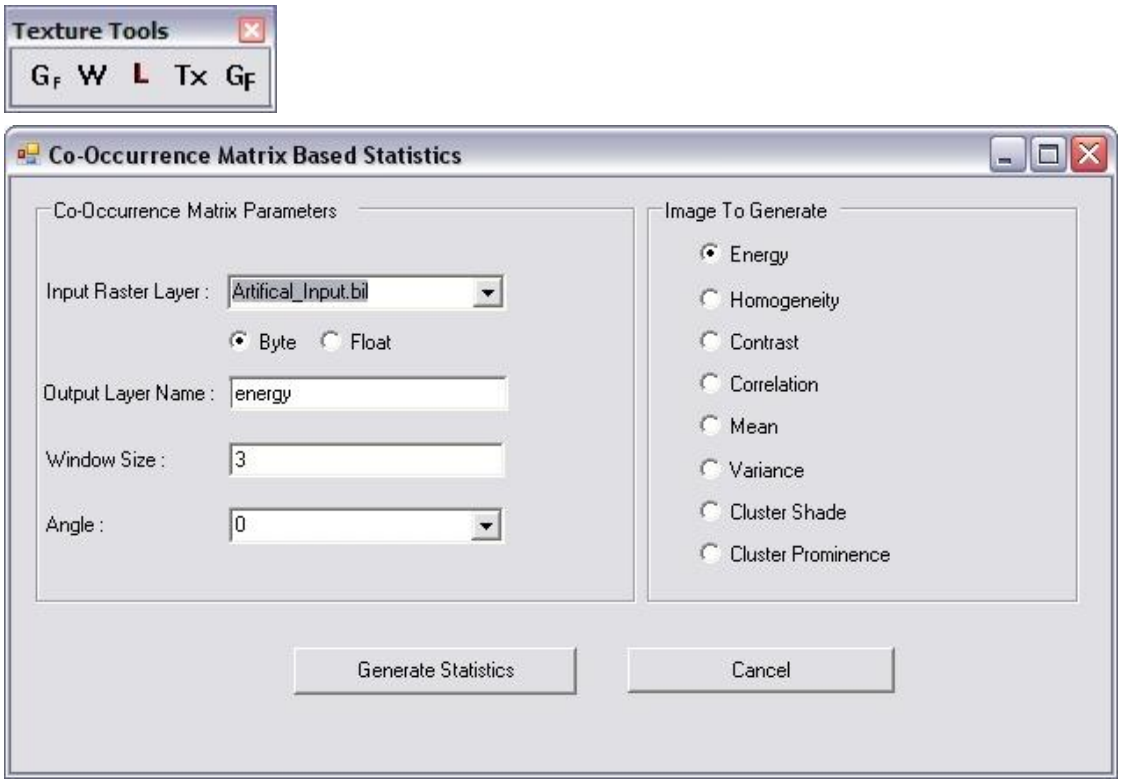


Figure 3.3 - Interface for GLCM computation (ArcGIS component)

3.1.3 Significance of Gray Level Co-occurrence Texture Measures

Each texture measure derived from the Gray level co-occurrence matrix is uniquely interpretable. The indicators *mean, homogeneity and angular second moment* are a measure of similarity in pixel values of the neighborhood resolution cells in an image block (Haralick, Dinstein, and Shanmugam 1973). If the image block being analyzed has fewer dominant gray level transitions then the co-occurrence matrix will have fewer entries of a large magnitude, which would mean lower values for the mentioned texture indicators (Haralick, Dinstein, and Shanmugam 1973). The *contrast* feature is a measure of the amount of local variations present in an image (Haralick, Dinstein, and Shanmugam 1973). The *correlation* measure is a measure or predictability of pixel values in the horizontal and vertical domains. It could also be described as a measure of linear dependencies in an image (Haralick, Dinstein, and Shanmugam 1973). *Entropy* can be described as a measure of the complexity or the measure of information in an image. The greater the variations in the neighborhood resolution cells, the greater the *entropy* values.

3.2 Laws Masks

Laws masks are a set of convolution masks used for deriving texture indicators by using a set of gradient operators (Lam and Li 1995). These gradient operators or local linear operators are based on combining the results of first and second order derivatives on an image block (Lam and Li 1995). A set of 25 convolution masks are

possible; using combinations of 5 one dimensional vectors as suggested by Laws (Lam and Li 1995).

Table 3.1 – One dimensional vectors used for Laws masks analysis

Level (L5)	[1 4 6 4 1]
Edge (E5)	[-1 -2 0 2 1]
Spot (S5)	[-1 0 2 0 -1]
Wave (W5)	[-1 2 0 -2 1]
Ripple (R5)	[1 -4 6 -4 1]

The Level (L5) vector gives a center weighted local average. Edge (E5) is similar to gradient operators and responds to row or column stepped edges in an image. Spot (S5) is based on the second derivative and is similar to performing a Laplacian over Gaussian (LoG). Wave (W5) responds to slight changes in pixel intensities in an image and Ripple (R5) is used to detect ripples in an image.

The result of the two dimensional convolution operation evaluated over every pixel in an image using the combinations of the one dimensional vectors in Table 3.1, is termed as *texture energy* (Lam and Li 1995). Each of these one dimensional vectors can be subject to rotation to make the resultant texture energy sensitive to the orientation.

Laws convolution masks are derived by combinations (multiplications) of two vectors while considering one of the vectors as a row vector and the other as a column

vector respectively. A set of such combinations is in Table 3.2. The texture energy is derived by convolving the generated laws convolution mask (*Laws(i,j)*) with an image *I(x,y)* of size N*N, expressed mathematically as:

Energy(i, j) = I(i, j) * Laws(i, j) where i, j range from *0 … (N-1)*

The computation of texture indicators based on Laws masks is implemented as a software tool using Microsoft Visual C# and ESRI ArcObjects (Fig. 3.4B). The general algorithm used in the derivation of texture indicators from Laws masks is:

1. Select the two Laws texture masks that are to be used. Let them be represented as *U, V* respectively.
2. Multiply matrices: *U*V* to get the 5*5 Laws mask *M.*
3. Convolute image *I(x, y) with M* to get the texture energy.

3.3 Preliminary Results with Artificial Textures

To test the statistical indicators derived from a co-occurrence matrix, an artificial image consisting of three dominant textured regions derived from the Brodatz texture album is used. The input image is a combination of textures *D16, D49* and *D53***.** The input image used is shown in Fig. 3.5. The input image consists of a high spatial frequency component with dominant central wavelength of $8\sqrt{2}$ pixels per cycle in the centre and two low spatial frequency components with wavelength $20\sqrt{2}$ and $22\sqrt{2}$ pixels per cycle and orientations of 0^o and 90^o respectively.

A set of statistical indicators derived by using Gray level co-occurrence matrices with parameters (Mask size = 7*7, d = 1 and $\alpha = 90$) are given in Figures 3.6 – 3.13.

The statistical indicators derived from using Laws texture masks are in Figures 3.14 – 3.15.

Table 3.2 – Examples of 5 commonly used Laws masks

E5-E5	$\begin{Bmatrix} 1 & 2 & 0 & -2 & -1 \\ 2 & 4 & 0 & -4 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -4 & 0 & 4 & 2 \\ -1 & -2 & 0 & 2 & 1 \end{Bmatrix}$
R5-R5	$\begin{Bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{Bmatrix}$
E5-S5	$\begin{Bmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -0 & 4 & 0 & -2 \\ -1 & 0 & 2 & 0 & 1 \end{Bmatrix}$
L5E5	$\begin{Bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{Bmatrix}$
L5S5	$\begin{Bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{Bmatrix}$

The detailed steps followed in the implementation of the algorithm are mentioned in Fig 3.4A.

```
calculate texture_energy(I, vector_1, vector_2){
	//Image I is of size N*N
	for(i = 0; i < N; i++){
		for(j = 0; j < N; j++){
			//Get Neighborhood resolution cells for a pixel at I(i,j)
			//of window size 'n'
			window = get_window(I, i, j, n)
			//Generate Laws Mask
			laws_mask = multiply(vector_1, vector_2)
			//Perform 2-D Convolution and Normalize by 'R'
			result = convolute_2d(window,laws_mask)
			//Save normalized result to output
			n_result = result / R
			output(i,j) = n_result
		}
	}
}
```

Figure 3.4A – Algorithm describing implementation of Laws texture analysis

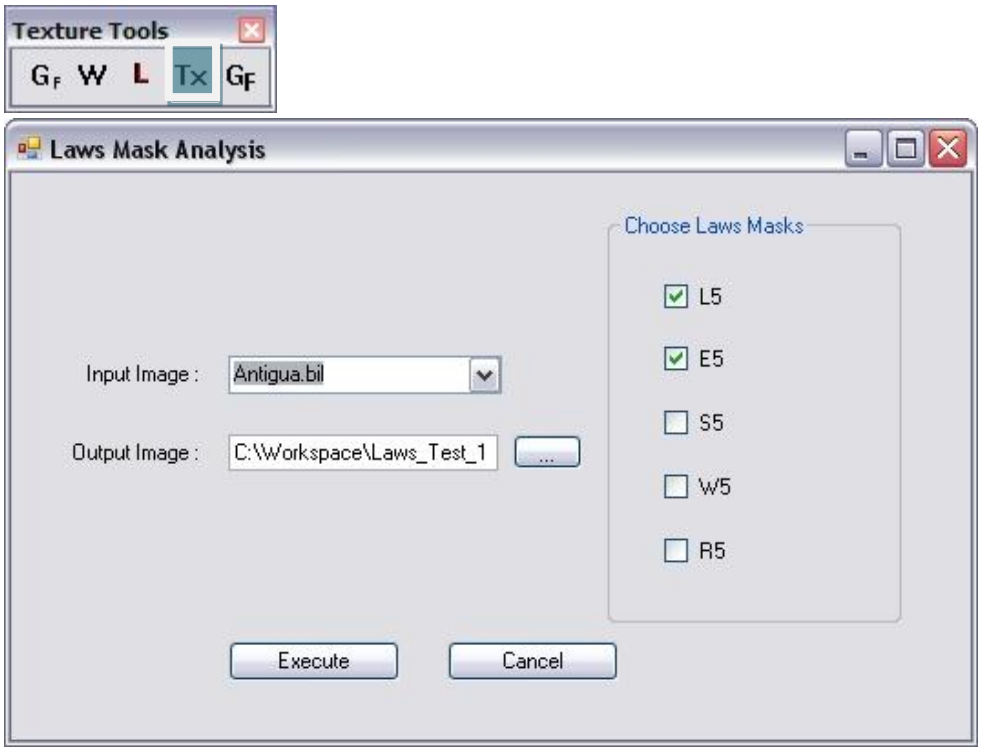


Figure 3.4B – User interface for derivation of indicators from Laws masks (ArcGIS component)

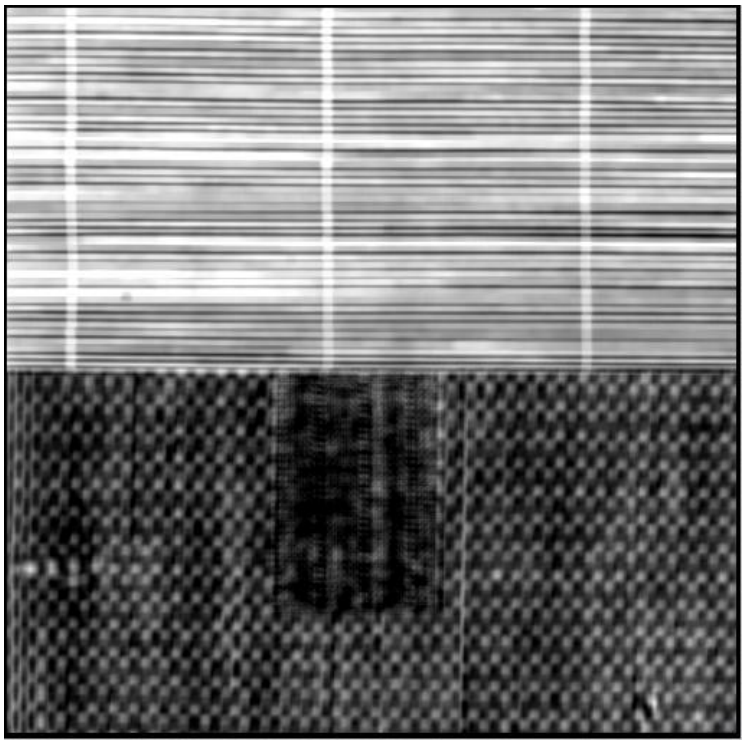


Figure 3.5 – Input image to test the statistical indicators

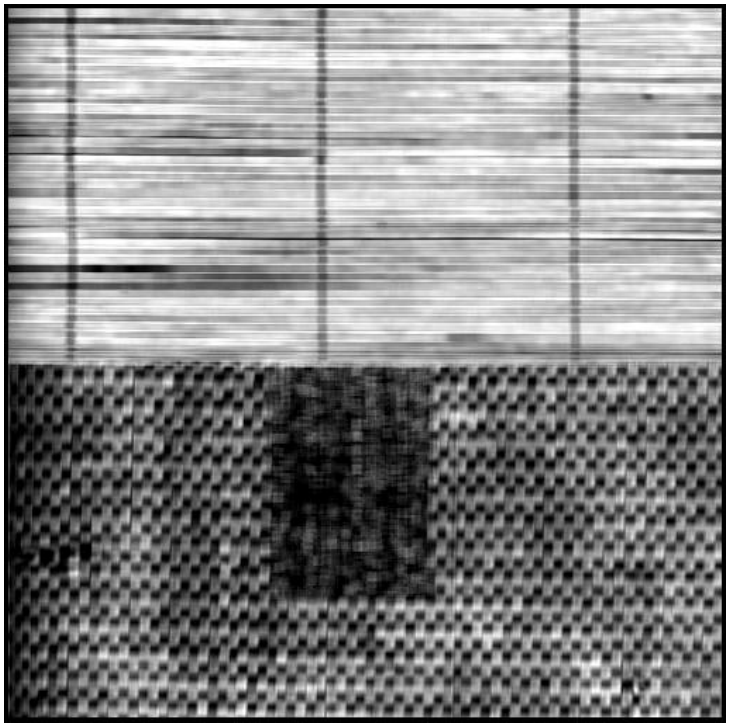


Figure 3.6 –GLCM indicator (Mean)

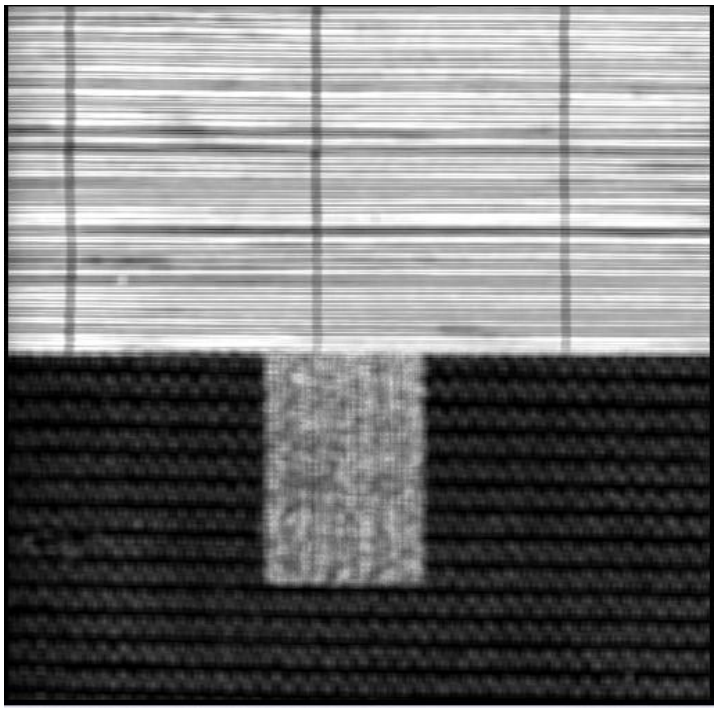


Figure 3.7 –GLCM indicator (Variance)

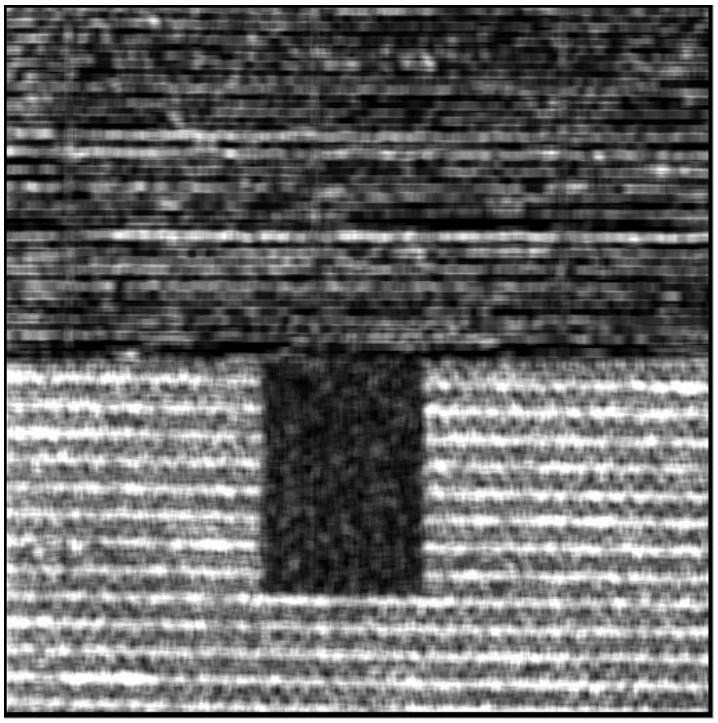


Figure 3.8 –GLCM indicator (Homogeneity)

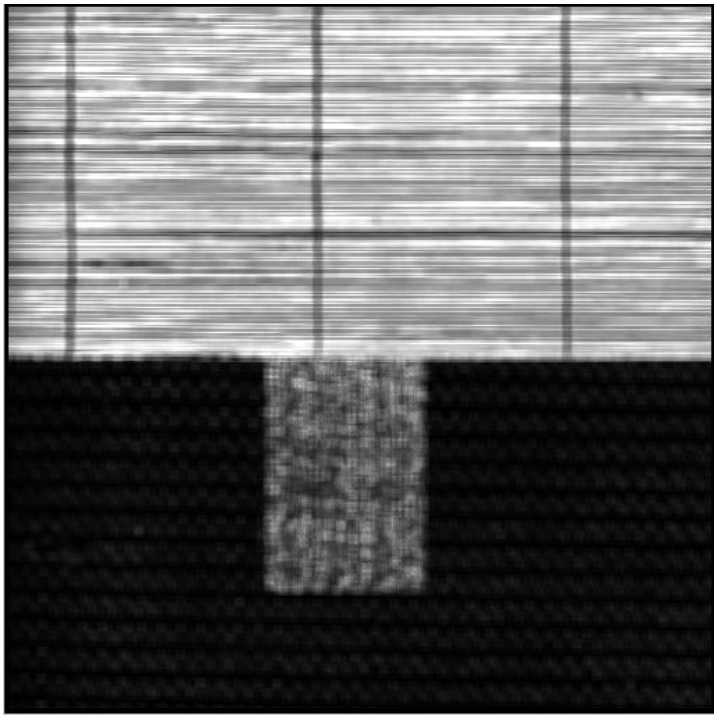


Figure 3.9 –GLCM indicator (Contrast)

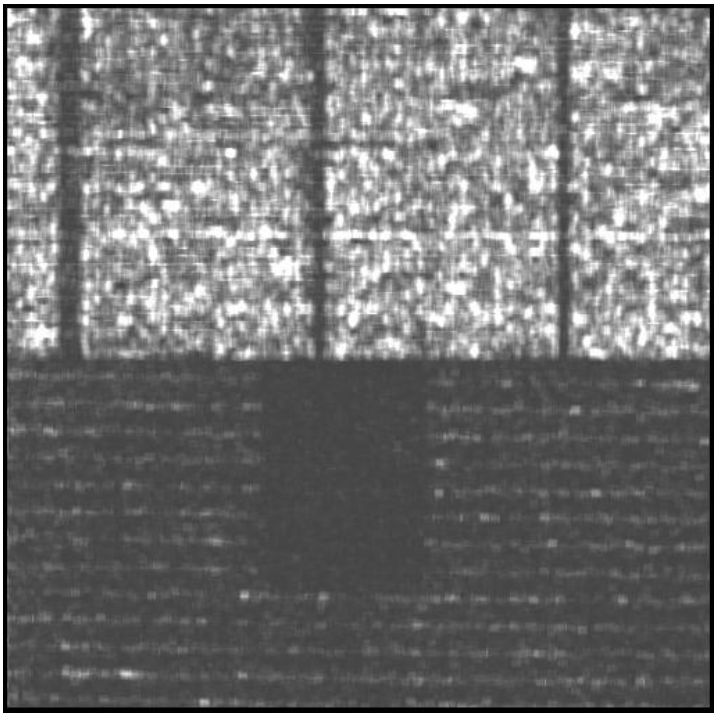


Figure 3.10 –GLCM indicator (Dissimilarity)

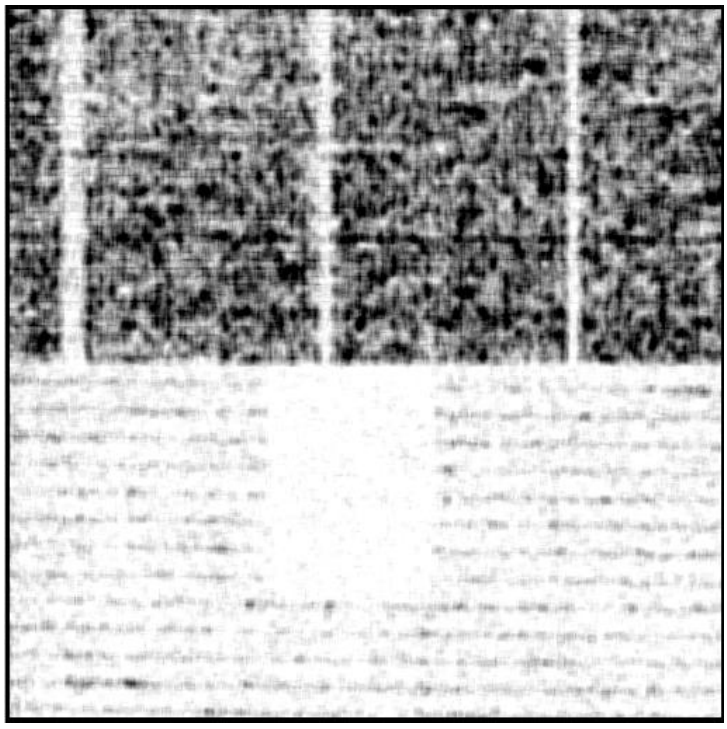


Figure 3.11 –GLCM indicator (Entropy)

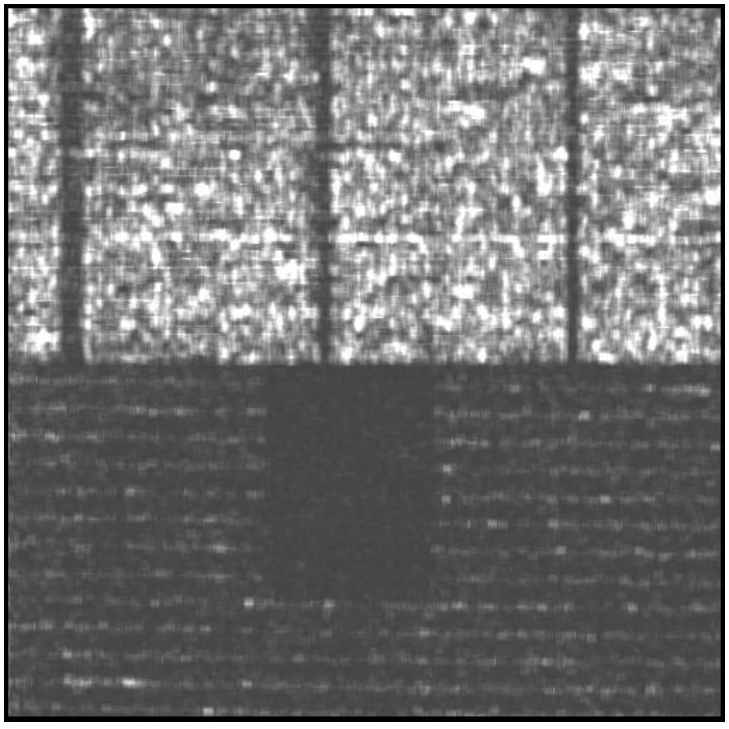


Figure 3.12 –GLCM indicator (Angular Second Moment)

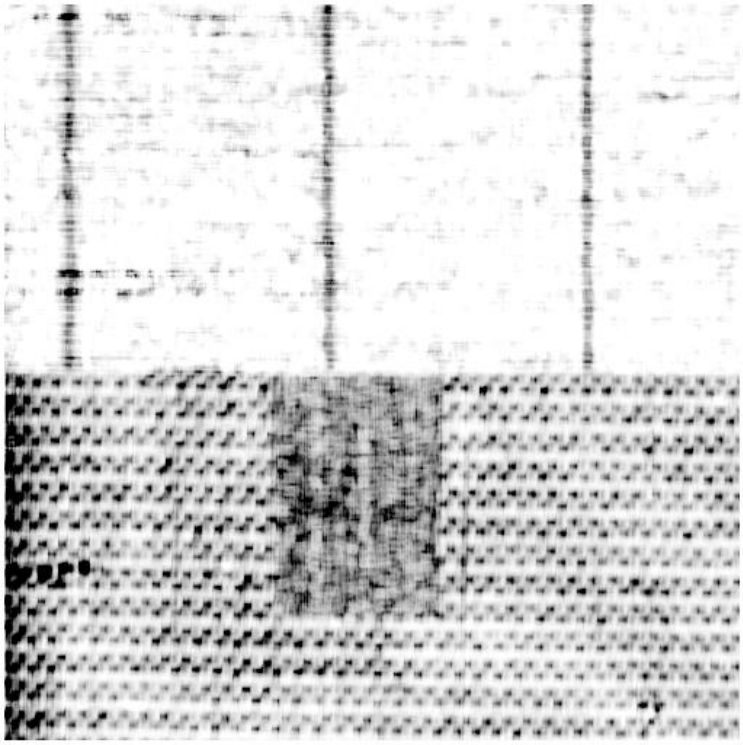


Figure 3.13 –GLCM indicator (Correlation)

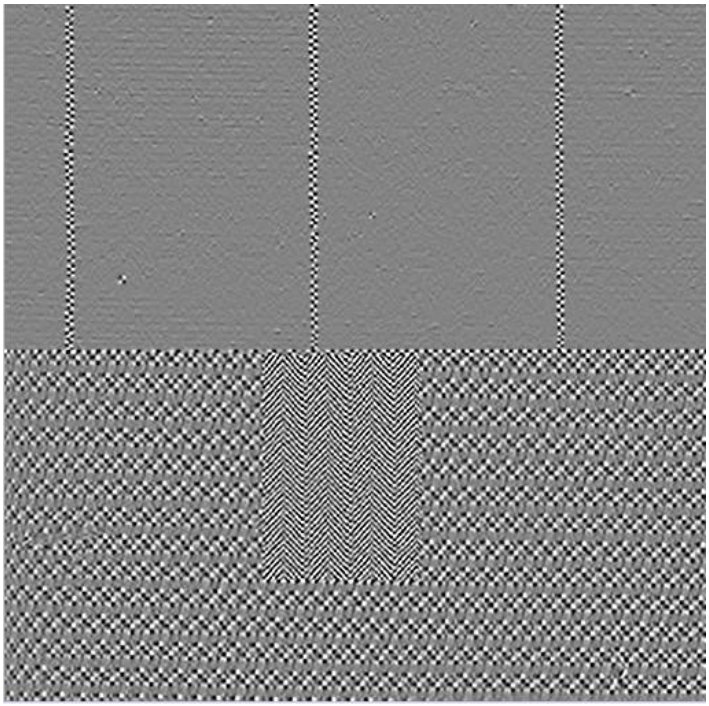


Figure 3.14 Laws texture indicator (E5E5)

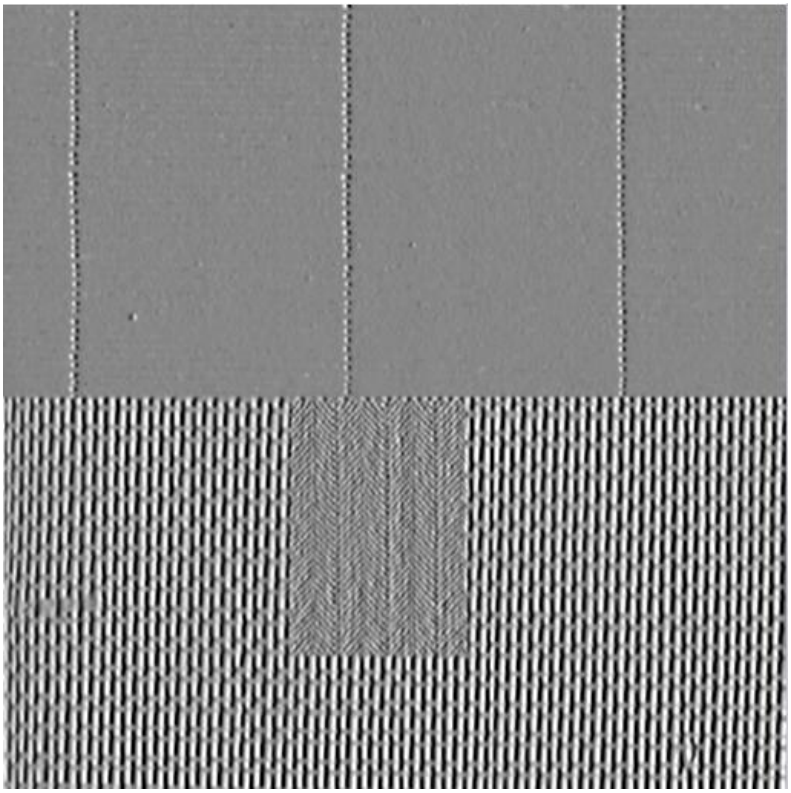


Figure 3.15 – Laws texture indicator (L5S5)

The classification results using statistical indicators derived from Gray level co-occurrence matrices and Laws masks are show in Fig. 3.16 and Fig. 3.17 respectively. The indicators used for classification in the case of Gray level co-occurrence matrices are homogeneity, contrast, dissimilarity and angular second moment. The indicators used for classification in the case of Laws masks analysis are derived from using the convolution masks *L5E5* and *L5S5*. These classified results are clearly an improvement when compared to the result derived from conventional gray scale thresholding methods (Fig. 3.18).

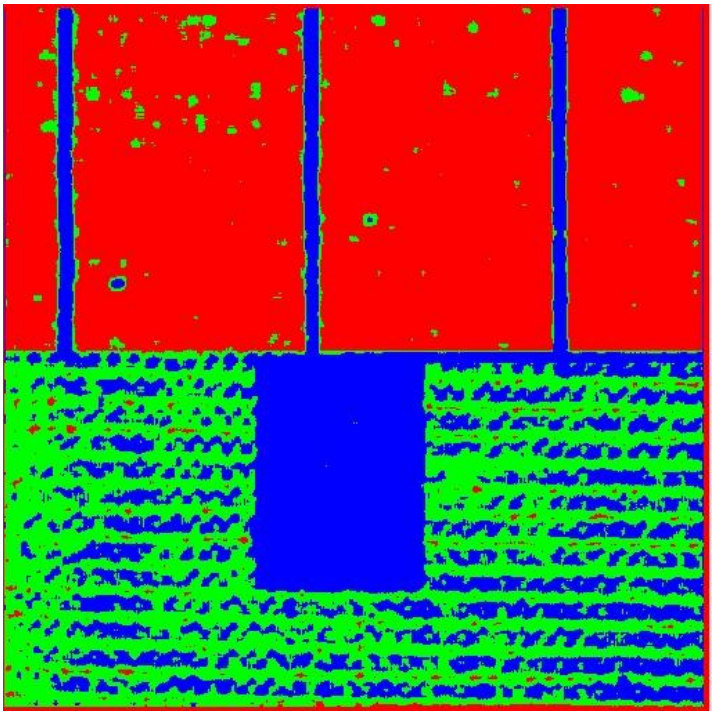


Figure 3.16 – Classification result using indicators derived from GLCM

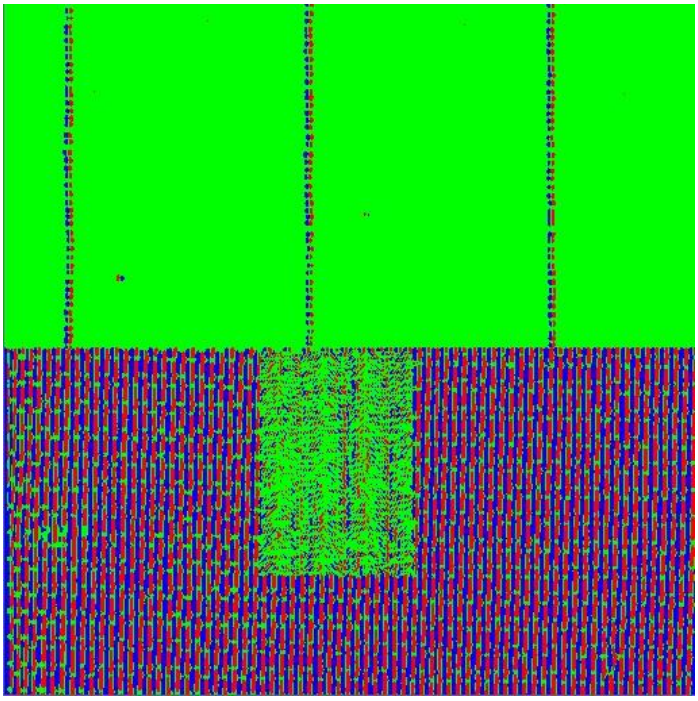


Figure 3.17 – Classification result using indicators derived from Laws masks

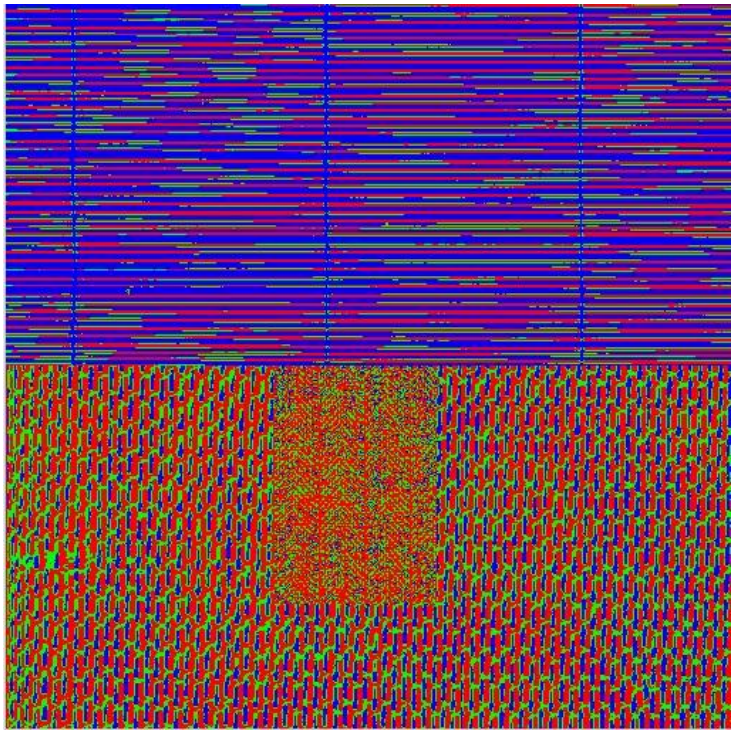


Figure 3.18 – Classification result derived from gray scale thresholding approach

CHAPTER IV

GABOR FILTER BANK METHOD FOR TEXTURE ANALYSIS

The exact definition of texture as a property of an image has never been formulated (Bovik, Clark, and Geisler 1990). It is very difficult to model texture in a stochastic or a probabilistic manner, as real world texture is exceedingly complex. The multi-channel filtering approach defines texture as a reflectance pattern that can be characterized by a high concentration of localized spatial frequencies (Bovik, Clark, and Geisler 1990). In this approach texture images are represented by multiple narrow frequency and orientation channels (Bovik, Clark, and Geisler 1990). The output of each channel is a complex modulated image whose amplitude and phase responses describe the spatial support of the frequencies or orientations to which this channel is tuned (Bovik, Clark, and Geisler 1990). The local structure of texture can therefore be described by the dominant frequency and orientation of the channel, while information describing the spatial extent is described by the channel envelope (Bovik, Clark, and Geisler 1990).

Texture is termed to be approximately periodic, and the rate of change of pixel intensities in an image is termed as *spatial frequency*. Although texture has a spatial frequency component associated with it, Fourier transforms are not suitable for analyzing the spatial frequency components of image texture. This is because the Fourier transformation of an image captures only global spatial frequency and is not capable of resolving or *localizing* spatial frequencies. Alternative methods like windowed Fourier

transformations are capable of localizing spatial frequencies to an extent. They are however restricted by the size of the window over which the Fourier transformation is evaluated. Filters that are capable of achieving joint resolution in both the spatial and the frequency domains are more suitable for the analysis of image texture.

Gabor functions are particularly useful when it comes to analysis of texture images consisting of highly specific frequency and orientation characteristics (Bovik, Clark, and Geisler 1990) (Fig 4.1). They have tunable frequency and orientations and can achieve joint resolution in the spatial and frequency domains (Clark, Bovik, and Geisler 1987). Receptive field profiles of simple cells are also known to approximate the one dimensional Gabor elementary functions which are essentially Gaussian modulated sinusoids in the spatial domain and shifted Gaussians in the frequency domain (Clark, Bovik, and Geisler 1987). Gabor signals are also known to minimize the space-frequency uncertainty principle so that they can be defined with very narrow frequency and orientation responses while maintaining very accurate spatial localization (Clark, Bovik, and Geisler 1987). Therefore, Gabor filters are important in the context of texture analysis.

4.1 The Multi-channel Filter Bank Model

2-D Gabor filters are defined as complex sinusoidal gratings modulated by 2-D Gaussian functions in the spatial domain and shifted Gaussians in the frequency domain (Clark, Bovik, and Geisler 1987; Bovik, Clark, and Geisler 1990). They are complex valued functions in R^2 except when the filter has zero central frequency, reducing to a 2-

D Gaussian function (Bovik, Clark, and Geisler 1990) (Fig. 4.1). Gabor filters can be configured to have various different frequencies, orientations, phase and bandwidths. By varying each of these properties, they can be used to filter any elliptical region of spatial frequencies (Bovik, Clark, and Geisler 1990). The amplitude and phase responses of each filter can be computed and analyzed separately, thereby forming a basis for application of other image processing techniques on the filtered images.

Filtering an image with a 2-D Gabor filter results in a complex image that consists of a limited range of spatial frequencies and orientations to which the filter has been tuned to (Clark, Bovik, and Geisler 1987). In the multi-channel model or a filter bank model, a set of filters are applied on an image in parallel to produce a series of channels of individual images (Dunn and Higgins 1995). These channels are then subject to further analysis based on the range of spatial frequencies contained in the channels. Thus if an input image consists of different textured regions, then the local spatial frequency variations in the textured regions get transformed to discontinuities in the output image channel (Dunn and Higgins 1995). Each of these discontinuities in the image channels represents a transition between different textured regions (Dunn and Higgins 1995).

Each textured region is *characterized* by a dominant spatial frequency, orientation and phase component. Different image textures will have different responses in different image channels as each Gabor filtered channel is tuned to a *specific* spatial frequency, orientation and phase. Figures 4.2 - 4.4 are examples of image textures with different frequency, orientation and phase components.

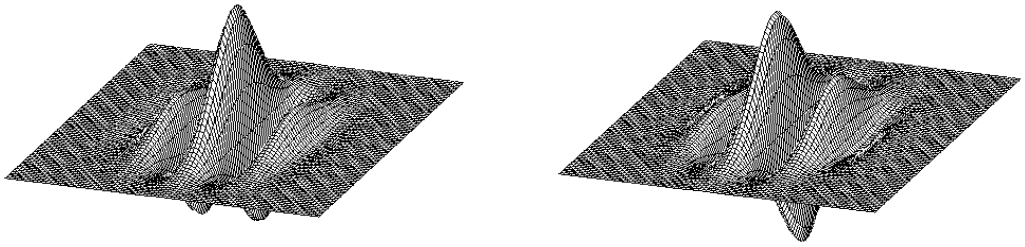


Figure 4.1 – Representations of real and imaginary components of a Gabor filter in the frequency domain

4.2 Mathematical Background

Daugman (1988) extended Gabor filters to 2-D and defined them in convolution form as:

$$h(x,y) = g(x',y') * \exp\ (2\pi j\ (ux + vy))$$

where,

$(x',y') = x\cos\emptyset + y\sin\emptyset, -x\sin\emptyset + y\cos\emptyset$ represent the rotated co-ordinates and

$$g(x,y) = \left(\frac{1}{2\pi\lambda\sigma^2}\right) * \exp\ -\left\{\frac{\left(\frac{x}{\lambda}\right)^2 + y^2}{2\sigma^2}\right\} \quad (4.1)$$

where the scales of the Gabor filter along both the x and y axes are defined by σ. The ratio λ determines the width to height ratio of the Gaussian in the frequency domain. This parameter can be used to vary the orientation sensitivity of the Gaussian (Clark, Bovik, and Geisler 1987). The central frequency of the filter is $\sqrt{u^2 + v^2}$. In the frequency domain, a Gabor filter can be represented as

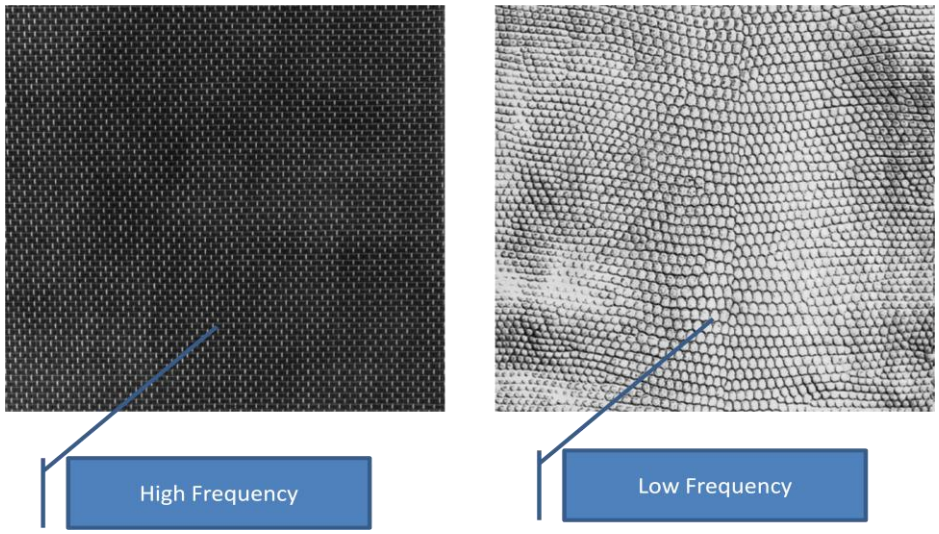


Figure 4.2 – Spatial frequency in image texture

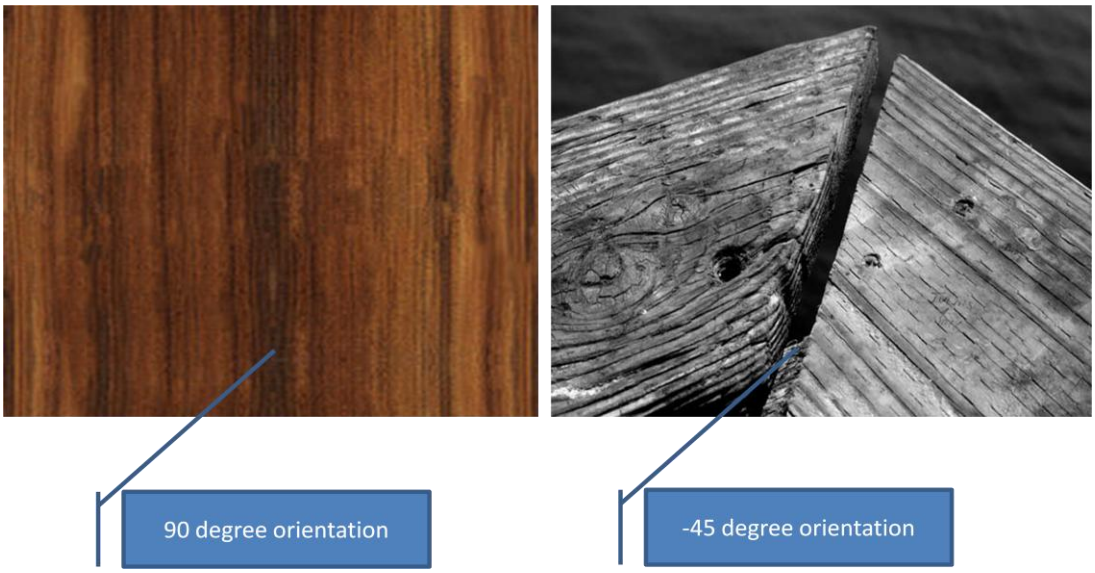


Figure 4.3 – Orientation in image texture

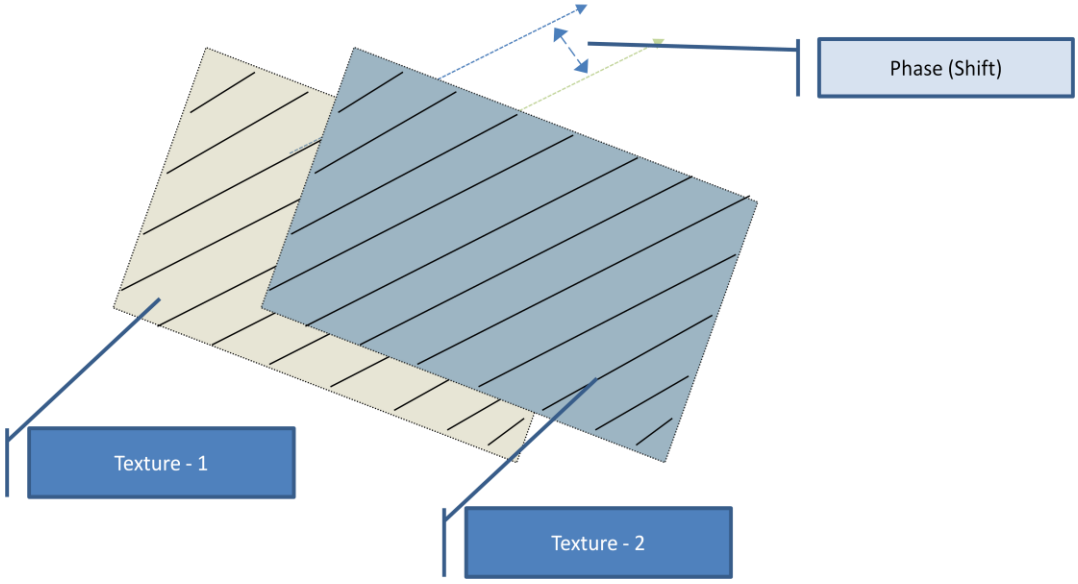


Figure 4.4 – Phase shifts in image texture

$$H(u,v) = \exp(-2\pi^2\,[\sigma^2\,\{(u-u_o)^2 + \,(v-v_o)^2]) \tag{4.2}$$

which is a 2-D Gaussian centered at (u_o, v_o). For convenience, the convolution version of the Gabor filter can also be represented as

$$h(x,y) = h_r\,(x,y) + \,i * \,h_i\,(x,y) \tag{4.3}$$

where $h_r\,(x,y)$is the real component and $h_i\,(x,y)$ is the imaginary component. Now,

$$h_r\,(x,y) = g(x,y) * \cos(2\pi u_o x) \tag{4.4}$$

and

$$h_r\,(x,y) = g(x,y) * \sin(2\pi u_o x) \tag{4.5}$$

when the Fourier domain representation in (4.2) has $v_o = 0$. The real and imaginary components of the Gabor filter have identical responses except for a phase difference of (π/2) (Clark, Bovik, and Geisler 1987).

4.3 Texture Classification Based on the Gabor Filter Bank

Let us assume that an input image represented by *f(x, y)* of infinite extent consists of *N* perceptually distinct regions $R_{1,\ldots},R_N$ that consists of different texture types $T_{1,\ldots},T_N$. If we assume that each texture can be identified via a narrow range of spatial frequencies thereby making them perceptually unique in the image then, according to Clark, Bovik, and Geisler (1987) the image *f(x, y)* can be represented as

$$f(x,y) = \,\sum_{n=1}^{N} f_i(x,y) * z_i\,\,(x,y) \tag{4.6}$$

where $z_i(x,y)$ is an indicator associated with region R_i and $f_i(x,y)$is a narrow band 2D image uniquely associated with textured region T_i consisting of a unique range of spatial

frequencies. Now an approximated narrow band texture component $f_i(x,y)$ can be represented as

$$f_i(x,y) = 2E_i(x,y) * \cos[2\pi\,(u_i\,x + v_i\,y)] \tag{4.7}$$

where i extends from 0…N-1 and $E_i(x,y)$ is assumed to be smooth and positive function. By using a Gabor filter appropriately (h_i) tuned to a central frequency (u_i, v_i) to extract texture T_i we can say that the Gabor expansion of the narrow band image $k_i(x,y)$ is,

$$k_i(x,y) = h_i(x,y) * f_a(x,y) \ll h_i(x,y) * f_b(x,y) \tag{4.8}$$

where a ≠ b as f_a is deficient in those frequencies that make up R_b thus making it *perceptually different* from other regions. We also have the Gabor expansion of $k_i(x,y)$ as ,

$$k_i(x,y) = h_i(x,y) * f_i(x,y)$$

$$= [g_i(x,y) * E_i(x,y) * z_i(x,y)]$$

$$* \exp(2\pi i(u_i x + v_i y)) \tag{4.9}$$

where $g_i(x,y)$ is the modulating Gaussian function used in the Gabor filter $h_i(x,y)$. Thus, in order to achieve complete classification, the magnitude of the channel response $|k_i(x,y)|$ has to be compared for ∀ (x, y), i.e.

$$(x_i, y_i) \in R_i \; if \; i = \arg\{\max\,[|k_i(x,y)|]\} \tag{4.10}$$

Clark, Bovik, and Geisler (1987) also suggest that the channel outputs of each filter are post processed with a Gaussian low pass filter to negate the effects of energy leakage and noise. The block diagram representing a model for multi-channel texture analysis is given in Figures 4.5 and 4.6.

In Fig. 4.5, if the input image *I(x, y)* consists of *N* perceptually different textured regions, then each of those textured regions can be represented by a unique range of spatial frequencies, orientations and phase components. The texture indicators pertaining to these textured regions can therefore be extracted by using an appropriate Gabor filter $h_i(x, y)$ specifically tuned to those spatial frequencies, orientations and phase components. Therefore N unique regions would require *N* specifically tuned Gabor filters, forming a Gabor filter bank.

Fig. 4.6 shows an image with two dominant textured regions, thereby consisting of a filter bank with two unique Gabor filters.

4.4 Filter Parameter Selection

The multi-channel model for image classification is not automated. This method needs human input for determining the right set of filter parameters for texture analysis. The main parameters that need to be selected for each filter are spatial frequency, orientation and phase. These parameters are dependent on the texture being processed. The automation of this system would require some form of pre-processing or the use of a very lengthy implementation to sort responses from various combinations of filter parameters (Bovik, Clark, and Geisler 1990).

Spatial frequency of a given texture is estimated by calculating an inverse of the wavelength of the texture measured in terms of pixels per cycle. The bandwidth of the filter σ is *proportional* to $\frac{\tan 22.5^{\circ} * u_0}{\sqrt{2 * ln\ 2}}$ (Randen and Husoy 1994).

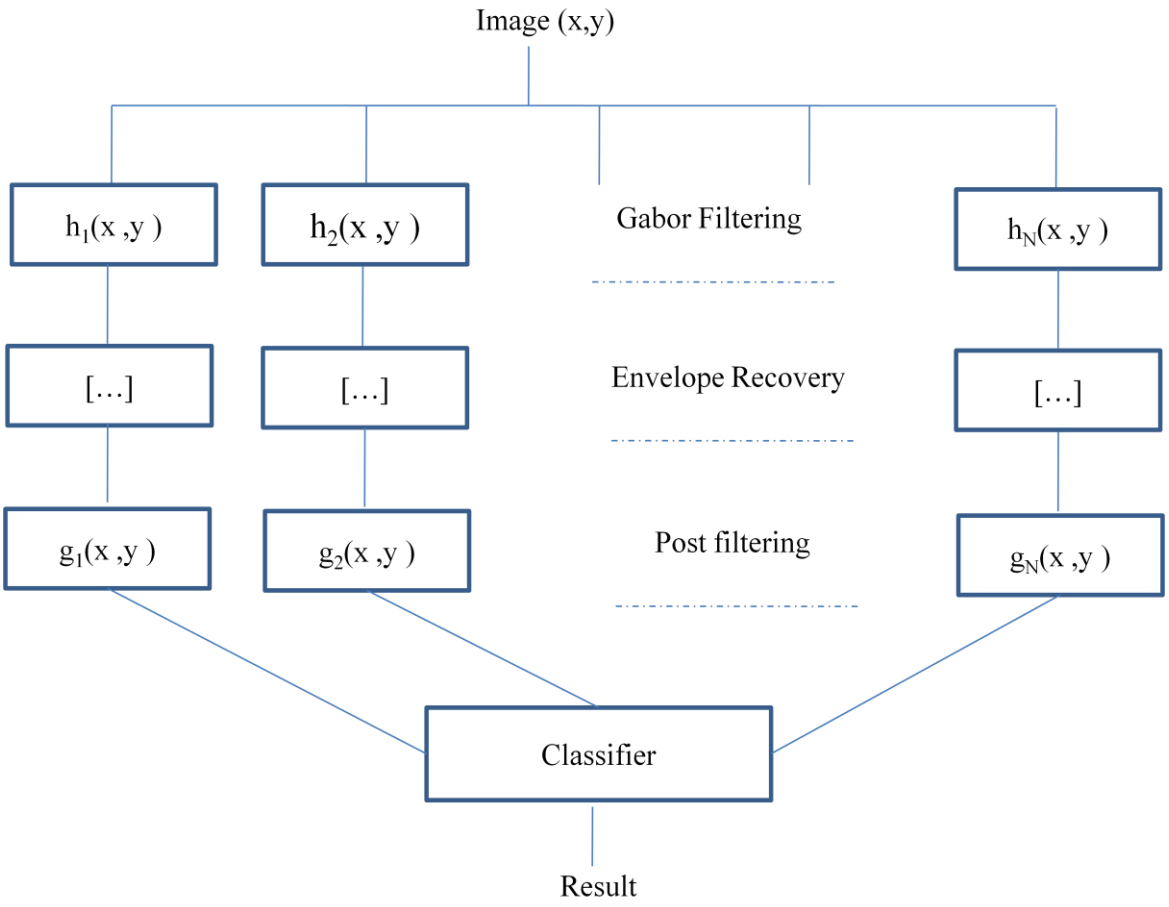


Figure 4.5 – Block diagram illustrating the multi-channel model for image classification

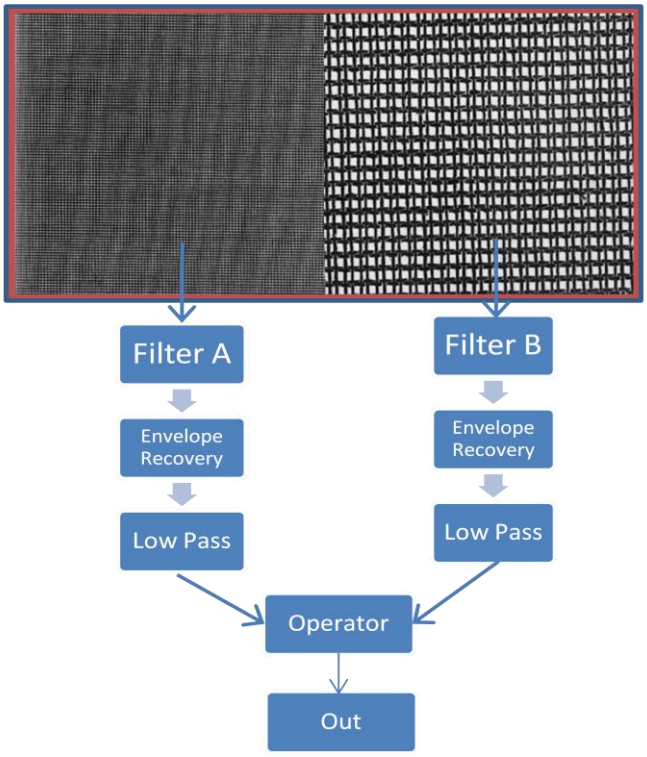


Figure 4.6 – Image classification model for an image with 2 dominant texture components

4.5 Implementation of Software Tool

The Gabor filter has been implemented in the spatial domain by using discrete convolution. The software is written using Microsoft Visual C# and ESRI ArcObjects (Fig 4.7). Post filtering of the image, after the application of the Gabor filter is done using a Gaussian low pass filter. The Gaussian low pass filters is also implemented using Microsoft Visual C# and ESRI ArcObjects (Fig. 4.8).

The general steps followed for the implementation of the Gabor filter are:

1. For every unique texture component in the input image do:
    a) Choose a set of Gabor filter parameters (*frequency, orientation, phase, standard deviations and aspect ratios)* based on the image texture being analyzed.
    b) Generate Gabor filter convolution masks (real and imaginary components) by modulating a Gaussian function with the input parameters.
    c) Convolute the input image *I(x, y)* with the generated Gabor filter masks to get filtered images.
    d) Post process the filtered images by using an appropriate Gaussian low pass filter whose standard deviation is *proportional* to W/u_0 where W is the width of the image in pixels and u_0 being the central frequency of the filter (Jain and Farrokhnia 1991).
2. Use all filtered images in the classification of image texture.
3. End

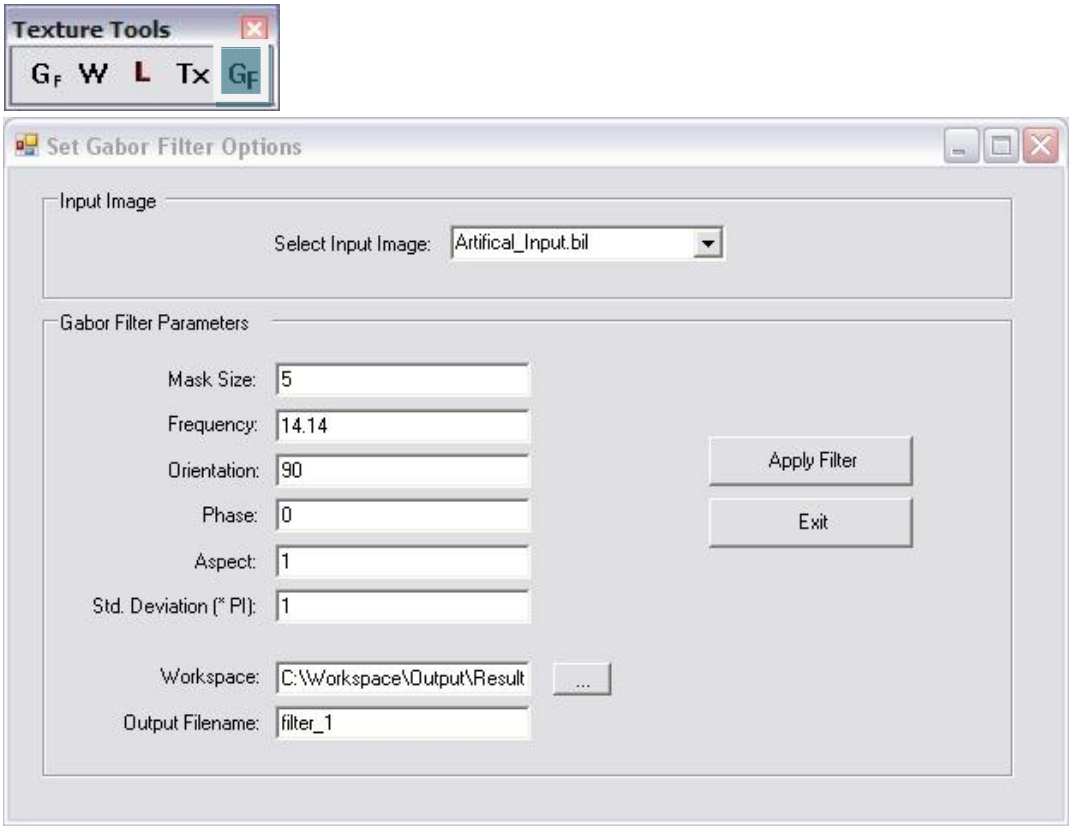


Figure 4.7 – User interface for the selection of Gabor filter parameters

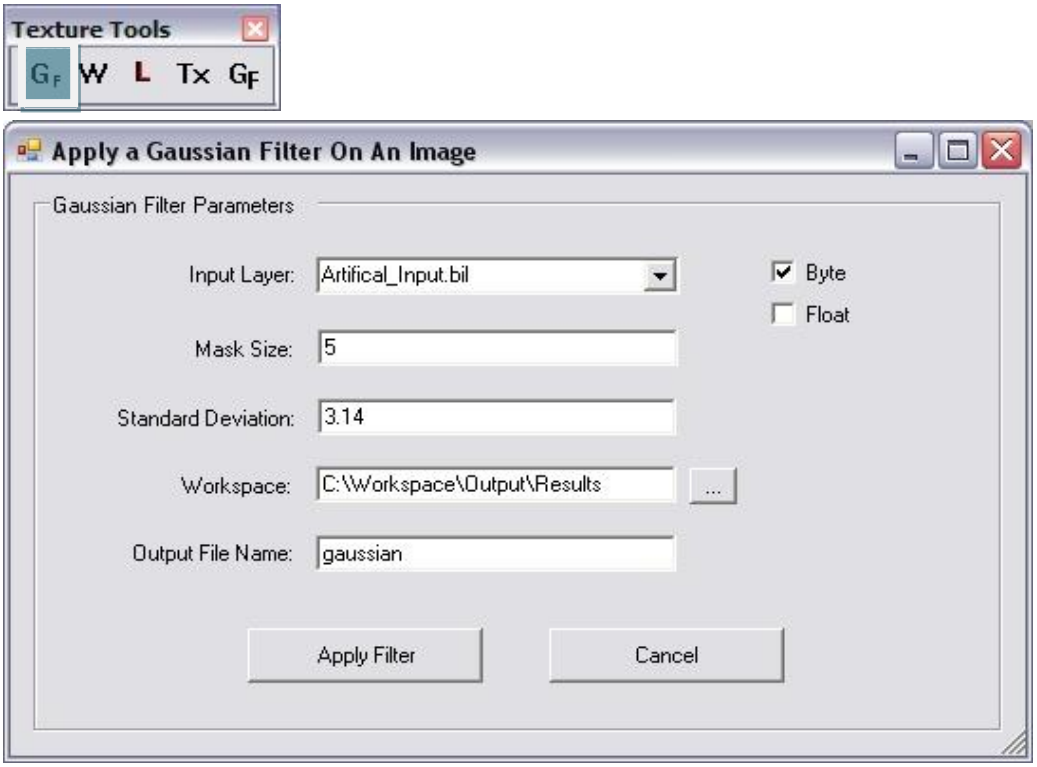


Figure 4.8 – User interface for the selection of Gaussian low pass filter parameters

The implementation strategy has been explained in detail in Fig. 4.9.

```
filter(I, mask_size, f, angle, p, aspect, stddev){
       image = read_image(I)
       //Get Gabor Real
       g_real = generate_gabor_filter_r(_
                mask_size, f, angle, p, aspect, stddev)
       //Get Gabor Imaginary
       g_real = generate_gabor_filter_r(_
                mask_size, f, angle, p, aspect, stddev)
       //2-D Discrete Convolution
       real = convolute_2d(image, g_real)
       imag = convolute_2d(image, g_imag);
       //Also generate Magnitute response
       //nrows, ncols being no. of rows and columns in the image
       for(i=0;i<nrows;i++){
              for(j=0;j<ncols;j++){
                     magnitude(i,j) = _
                     sqrt(pow(real,2), pow(imag,2))
              }
       }
       //Output Image
}
```

Figure 4.9 – Core implementation of the 2-D Gabor filter

The support functions used in the above implementation strategy are explained in Fig. 4.10.

```
//Generates the Real part of the Gabor Filter
generate_gabor_filter_r(mask_size, freq, orient, phase, aspect, stddev){
	//Determine mask origin
	mask_x_orig = floor(mask_size / 2)
	mask_y_orig = floor(mask_size / 2)
	//#Define mask
	for(i=0; i< mask_size;i++)
	{
		for(j=0; j< mask_size; j++)
		{
			//Evaluate every mask_coefficient
			x_offset = i - mask_x_orig
			y_offset = j _ mask_y_orig
			//Evaluate Rotated Co-ordinates
			x_dash = x_offset * cos(angle)
					+ y_offset * sin(angle)
			y_dash = -1 * x_offset * sin(angle)
					+ y_offset * cos(angle)
			//Get Gaussian Coeffient
			g_coeff = get_gaussian_coeff(_
					x_dash, y_dash)
			temp = 2 * pi * x_dash / freq
			mask(i,j) = _//Mask value
			coefficient * cos((pi * temp) + _
					(pi * phase) / 180.0)
		}
	}
```

Figure 4.10 – Support functions used to generate the Gabor filter

```
//Generates the Imaginary part of Gabor Filter
generate_gabor_filter_i(mask_size, freq, orient, phase, aspect, stddev){
	//Determine mask origin
	mask_x_orig = floor(mask_size / 2)
	mask_y_orig = floor(mask_size / 2)
	#Define mask
	for(i=0; i< mask_size;i++)
	{
		for(j=0; j< mask_size; j++)
		{
			//Evaluate every mask_coefficient
			x_offset = i - mask_x_orig
			y_offset = j _ mask_y_orig
			//Evaluate Rotated Co-ordinates
			x_dash = x_offset * cos(angle)
					+ y_offset * sin(angle)
			y_dash = -1 * x_offset * sin(angle)
					+ y_offset * cos(angle)
			//Get Gaussian Coeffient
			g_coeff = get_gaussian_coeff(_
					x_dash, y_dash)
			temp = 2 * pi * x_dash / freq
			mask(i,j) = _//Mask value
			coefficient * sin((pi * temp) + _
					(pi * phase) / 180.0)
		}
	}

//This function is used to generate the 2-D Sinusoidal Grating Coefficient
for a given (x, y)
get_gaussian_coeff(x,y){
	aspect_constant = 1 / (2 * pi * pow(stddev, 2) * aspect)
	std_term = 1 / (2 * pi * pow(stddev, 2))
	//Exponential term
	exp_term =
	aspect_constant * _
	exp(-1 * (pow(x/aspect, 2) + (pow(y/aspect),2))/std_term)
	//Final Coefficient
	return exp_term
}
```

Figure 4.10 (Continued)

4.6 Preliminary Results with Artificial Textures

To test the Gabor filter, an artificial image consisting of three dominant textured regions derived from the Brodatz texture album is used. The input image is a combination of textures *D16, D49* and *D53***.** The input image used is shown in Fig. 4.11. The input image consists of a high spatial frequency component in the centre and two low spatial frequency components with orientations at $0^o$and 90^orespectivily. The Gabor channel responses for this image for each of the frequency and orientation combinations are given in Figures 4.12 – 4.14.

The Gabor channel response obtained in Fig. 4.12 is obtained from choosing a high frequency Gabor filter attuned to the dominant spatial frequency of the textural component. The input parameter for the filter used in generating Fig. 4.12 are (frequency = $1/8\sqrt{2}$, orientation =90^o, phase = 0, aspect = 1, and standard deviation of Gaussian = π).

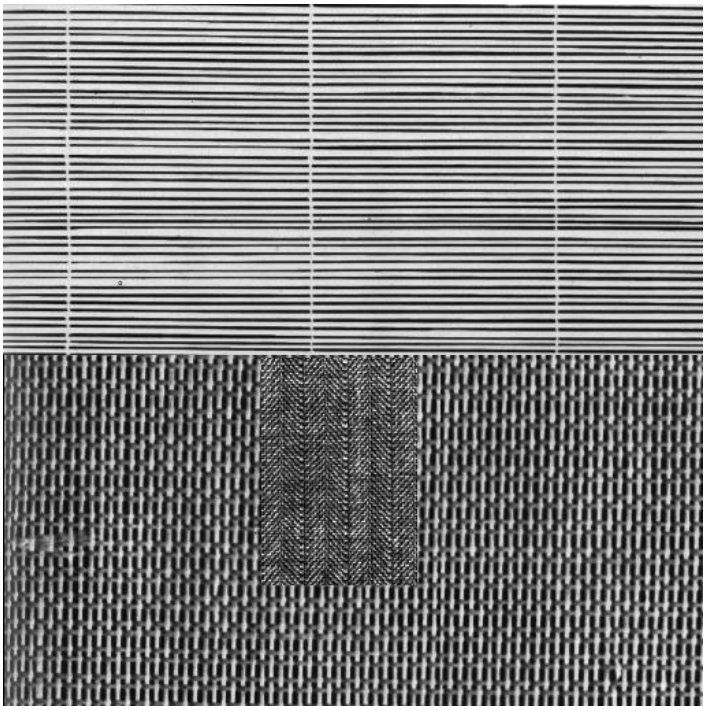


Figure 4.11 – Input image to test the Gabor filter bank

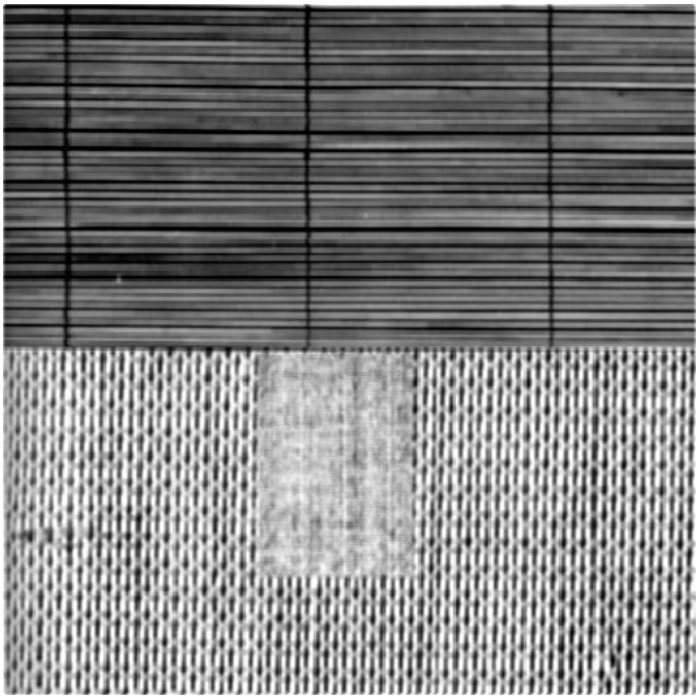


Figure 4.12 – Gabor response of the high spatial frequency component in the center of the image

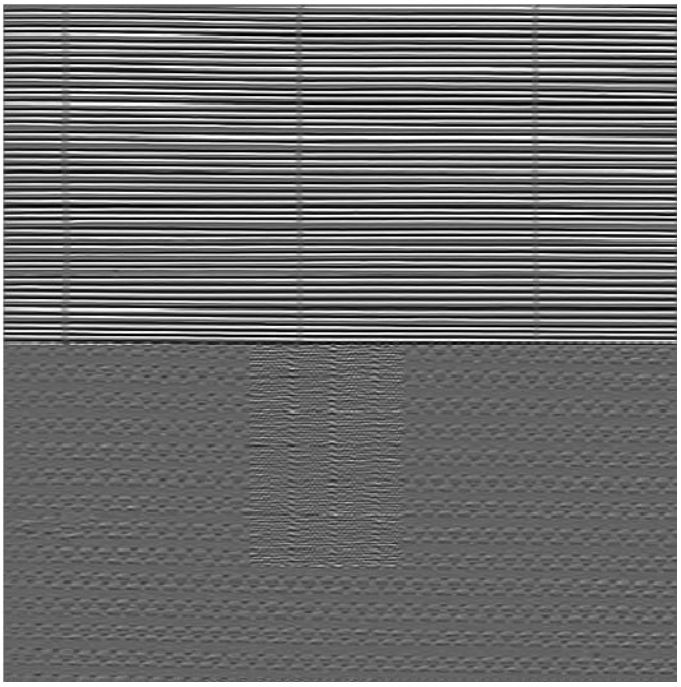


Figure 4.13 – Gabor response of lower spatial frequency component oriented at 0^o

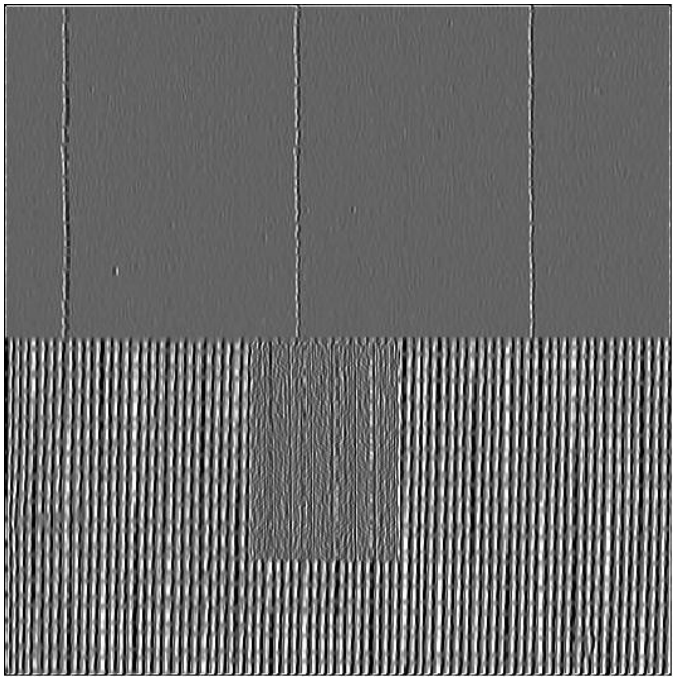


Figure 4.14 – Gabor response of the lower spatial frequency component oriented at 90^o

Similarly the lower spatial frequency textural components in the image (Fig. 4.11) oriented at $0^o$and 90^o respectively are filtered using a set of Gabor filters with input parameters (frequency = $1/20\sqrt{2}$, orientation =90^o, phase = 0, aspect = 1, and standard deviation of Gaussian = π) and (frequency = $1/22\sqrt{2}$, orientation = 0^o, phase = 0, aspect = 1, and standard deviation of Gaussian = π) respectively to generate Fig. 4.13 and Fig. 4.14 respectively. The filtered images are then filtered with an appropriate Gaussian low pass filter so as to reduce the noise in the filtered output.

The final classification result of the input image, using the Gabor filtered channels' is shown in Fig. 4.15. This classified result is clearly a great improvement when compared to the result derived from conventional gray scale thresholding methods (Fig. 4.16).

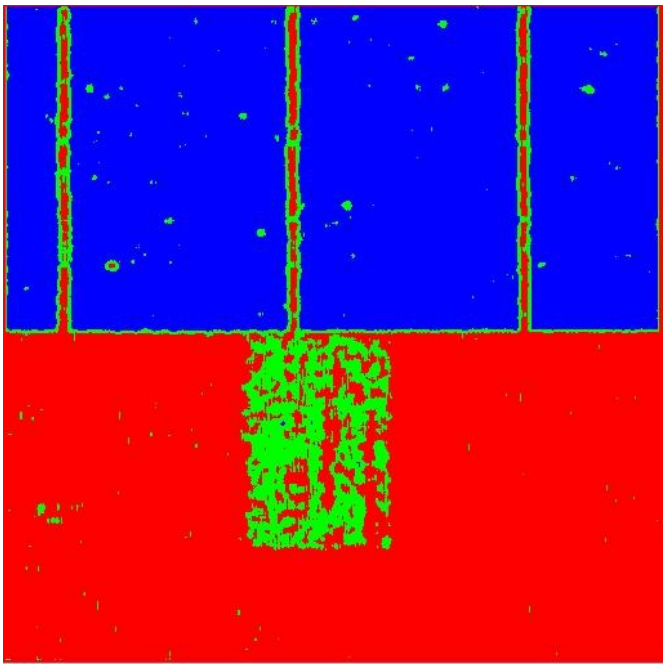


Figure 4.15 – Classification result derived from using Gabor filtered channels

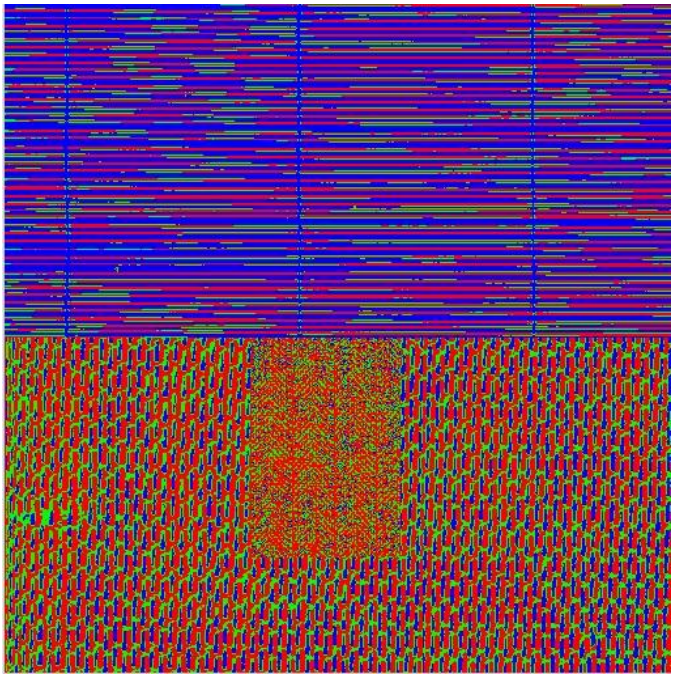


Figure 4.16 – Classification result derived from gray scale thresholding approach

CHAPTER V

WAVELET TRANSFORM METHODS FOR TEXTURE ANALYSIS

5.1 Need for a Multi-scale Approach

The use of statistical indicators has been one of the most popular methods of texture analysis (Tuceryan and Jain 1993). However statistical methods of deriving texture indicators suffer a major drawback as they do not address the problem of spatial scale. In order to derive texture indicators from Gray level co-occurrence matrices, decisions about the size of the neighborhood, the inter-pixel distance and the orientation angle need to be made. If the texture is coarse-grained, then the size of the neighborhood has to be large, so would be the inter-pixel distance on which the measurements are made. However, if the texture is fine-grained, then the window size and the inter-pixel distance would have to be small.

The spatial scale at which texture is analyzed defines the *coarseness* or *fineness* of texture. In order to analyze image texture, the size of the texture processing unit or *texel* has to be determined. Thus there is a need for a multi-scale approach to texture analysis.

5.2 Wavelet Theory

A wavelet transform decomposes an image into a set of images, based on an *ortho-normal basis* (Livens et al. 1997). These transformations are calculated by translating and scaling a single mother wavelet which is localized in both the spatial and

the frequency domains (Livens et al. 1997). When this transformation is applied in discrete steps, it becomes a discrete wavelet transformation. Every wavelet implementation consists of a low pass and a high pass filter and the image is dilated by a factor of two after applying the transformation. This scheme of implementation is called a *dyadic* wavelet transformation.

A wavelet decomposition can be represented mathematically as a convolution of the input image *f(x, y)* with a family of functions (Lu, Chung, and Chen 1997):

$$\left[f(x,y), \varphi_{2^s,t}(x,y)\right] = \int\int f(x,y) * \varphi_{2^s,t}(x,y)dx\,dy \tag{5.1}$$

The basic functions $\varphi_{2^s,t}(x)$ are obtained by translating and scaling a single kernel function or a mother wavelet Ж(x).

$$\varphi_{2^s,t}(x) = \varphi_{2^s}(x - 2^{-s}t) = 2^s * Ж(2^s x - t) \tag{5.2}$$

where *s*, *t* are the scaling and translation parameters. For Ж(x) to be characterized as a mother wavelet it must satisfy the admissibility condition:

$$\int_{-\infty}^{\infty} \frac{|Ж_f(w)^2|}{|w|}\, dw < \infty \tag{5.3}$$

or equivalently,

$$Ж_f(0) = \int_{-\infty}^{\infty} Ж(t)dt = 0 \tag{5.4}$$

where $Ж_f$ is the Fourier transform of Ж.

In multi-resolution analysis framework wavelet decomposition is achieved through the use of *scaling* and *fluctuation* functions which are essentially low pass and high pass quadrature mirror filters.

A general Gaussian and Laplacian pyramidal decomposition of an image is represented in Fig. 5.1. At every level in a Gaussian pyramidal decomposition, the input image is subject to a Gaussian low pass filter whose cut-off frequency can be controlled by varying the standard deviation of the filter. This filtered image is subsequently sub-sampled to get a representation of the image at a lower spatial scale. The Laplacian of the filtered image can be computed by calculating the difference between the original image and the filtered image. This process is repeated to get more images at lower scales thus creating an image pyramid.

In contrast, a wavelet decomposition of an image is represented schematically in Fig. 5.2A. Low pass and high pass filters are subsequently applied on the input image. Low pass filters result in an average or a *trend* sub image and the high pass filters result in a difference or a *fluctuation* sub-image. This is done in both the horizontal and the vertical directions, resulting in a set of four images. Every sub image is sub-sampled by a factor of two. This leads to representation with equal number of pixels as that of the input image with the possibility of *complete lossless reconstruction* to that of the original resolution. The same process is repeated iteratively for a multi-level decomposition resulting in an image pyramid. If the original image consists of 2^n rows and columns, the decomposition can be carried *n* times.

5.3 Haar Wavelet Transform

To explain the implementation of a Haar wavelet, let f = (f_1, f_2, $f_{3,\ldots}$ f_N) be a discrete vector of length *N*. The values of this discrete vector are real numbers and are typically measured values of an analog function *g(t)* at times (t_1, t_2, $t_{3,\ldots}$ t_N). Like all wavelet transforms the Haar wavelet transforms the original input vector *f* into vectors half its original length. The *average* vector is obtained by taking an average of the discrete input vector in the following manner.

$$t_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \tag{5.5}$$

where, m =1, 2…, N/2.

If we considered a one dimensional input vector defined by eight values: f = (4,6,10,10,12,8,5,5).

The *trend* vector t^1 for the above input vector is $(5\sqrt{2}, 10\sqrt{2}, 10\sqrt{2}, 5\sqrt{2})$. The other *fluctuation* vector which can be represented mathematically as,

$$diff_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \tag{5.6}$$

The *fluctuation* vector $diff^1$ for the input vector is $(-\sqrt{2}, 0, 2\sqrt{2}, 0)$.

A Haar transformation can be performed in multiple stages. The first level of the transformation can be represented as

$$f_{H1} = (t^1 | \, diff^1) \tag{5.7}$$

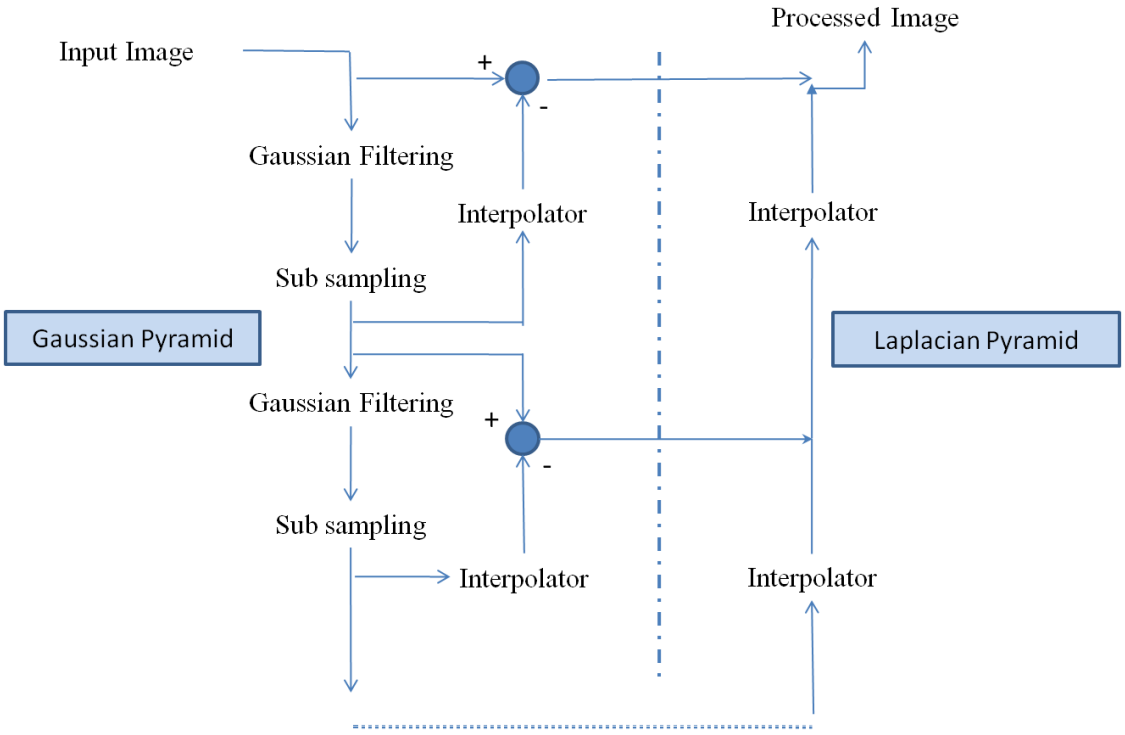


Figure 5.1 – Traditional Gaussian and Laplacian pyramids

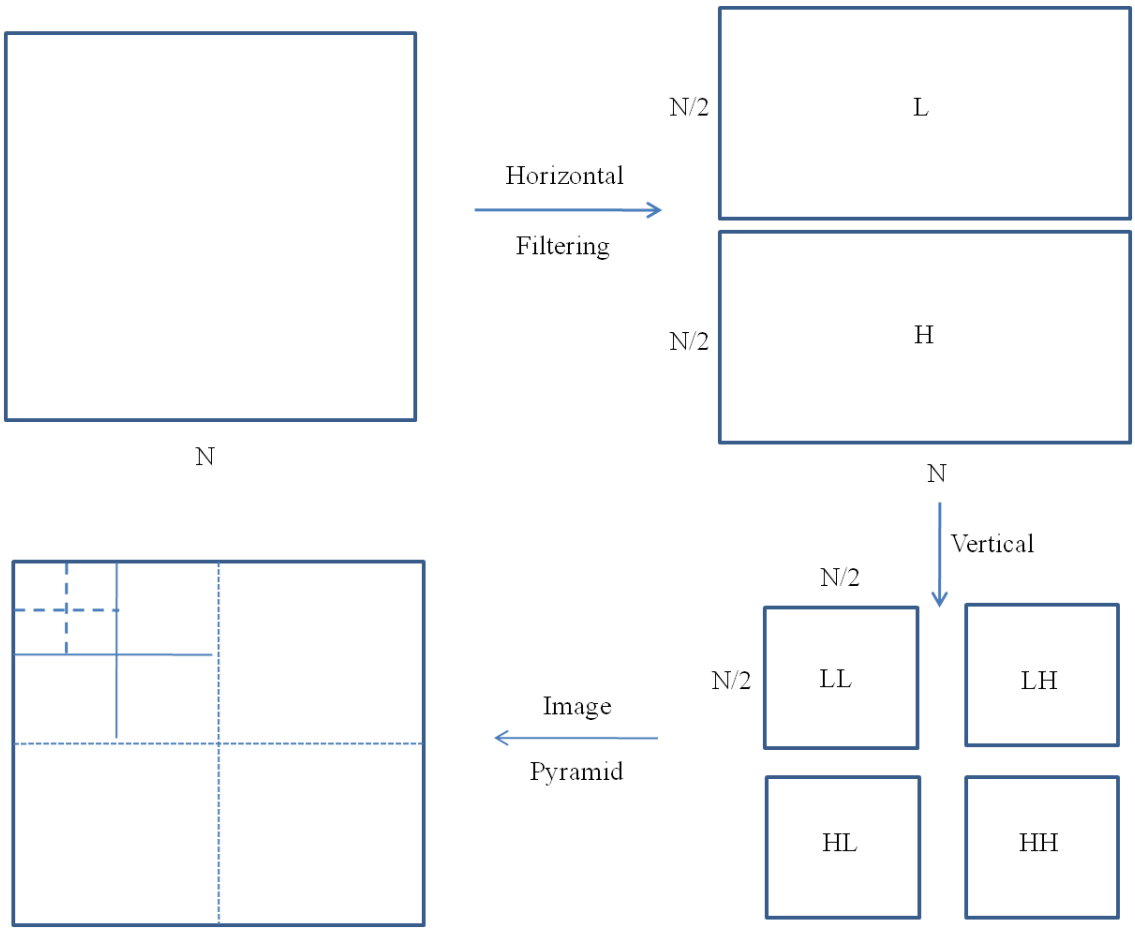


Figure 5.2A – Wavelet decomposition and image pyramids

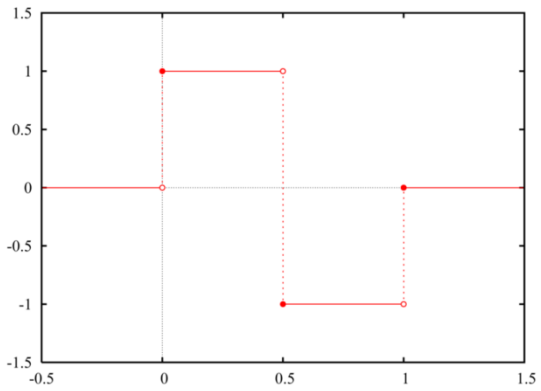


Figure 5.2B – Haar wavelet

In the example, for the input vector *f* the first level Haar transformation can be represented as $(5\sqrt{2}, 10\sqrt{2}, 10\sqrt{2}, 5\sqrt{2} \mid (-\sqrt{2}, 0, 2\sqrt{2}, 0)$. The Haar wavelet transformation also has an inverse. Given *t* and *diff* vectors, the inverse Haar transformation can be represented as

$$f = \left\{ \frac{(t_1 + diff_1)}{\sqrt{2}}, \frac{(t_1 - diff_1)}{\sqrt{2}}, \ldots, \frac{\left(t_{\frac{N}{2}} + diff_{\frac{N}{2}}\right)}{\sqrt{2}}, \frac{\left(t_{\frac{N}{2}} - diff_{\frac{N}{2}}\right)}{\sqrt{2}} \right\} \tag{5.8}$$

The advantages of transforming the original input signal *f* into a trend and fluctuations vectors are, that the magnitudes of the transformed *fluctuations* vector is often significantly smaller than the magnitude values of the original input vector (Walker 1999). This property of a wavelet transformation is known as the *small fluctuations feature*. The reason why this *small fluctuations feature* is generally true is that, the input vector that we deal with are samples of a continuous analog signal *g* measured in between very short time intervals between the samples (Walker 1999). If the time increment between subsequent samples is small enough, then the sampled values of the input signal *f* will be close to each other due to the continuity of *g.* This property of a wavelet transformation has applications in image compression and image representation. Another important property of a wavelet transformation is the conservation and compaction of *energy*. The energy of an input vector ε_r is defined as

$$\varepsilon_r = {f_1}^2 + {f_2}^2 + {f_3}^2 + \ldots + {f_N}^2 \tag{5.9}$$

A wavelet transformation conserves energy of an input vector and compresses most of the energy of the vector into the *trend* vector. For the above example the energy

of the trend vector accounts for 98.7% of the energy of the vector. The property of *energy compaction* of the input signal means that the properties of the input vector have not altered although the length of the signal has been reduced to half of its original size.

The idea of a Haar wavelet transformation can be extended for multi-resolution analysis. Given an input vector f then the first level Haar transformation can be represented as:

$$f = T_1 + Diff_1 \tag{5.10}$$

To compute the second level transformed function, the first order *trend* vector is taken into account,

$$T_1 = T_2 + Diff_2 \tag{5.11}$$

Thus the original input vector can be represented as

$$f = T_2 + Diff_2 + Diff_1 \tag{5.12}$$

In general for vector of length N (2^k), the k^{th} level multi resolution analysis can be represented as

$$f = T_k + Diff_k + \ldots + Diff_2 + Diff_1 \tag{5.13}$$

In the above example, inputs are transformed into trend component and fluctuation components taking into account a one dimensional input vector. However images are two dimensional in nature and therefore need to be subject to transformations along the horizontal and the vertical directions thereby resulting in four sub-images (Fig. 5.2A).

A Haar wavelet (Fig. 5.2B) is somewhat irregular and discontinuous (Kumar and Georgiou 1997) but provides a good choice for images that consist of sharp variations of pixel intensities. It's also simple to understand and implement.

5.4 Daubechies Wavelet Transform

The implementation of the Daubechies wavelet is very similar to the implementation of the Haar wavelet with a few important differences. The first major difference in the implementation of a Daubechies wavelet transformation is in the wavelet coefficients used for transformation of the input signal. The trend coefficients for the *Daub4* wavelet transformation $\propto_1, \propto_2, \propto_3, \propto_4$ are

$$\propto_1 = \frac{(1+\sqrt{3})}{4\sqrt{2}}, \propto_2 = \frac{(3+\sqrt{3})}{4\sqrt{2}}, \propto_3 = \frac{(3-\sqrt{3})}{4\sqrt{2}}, \propto_4 = \frac{(1-\sqrt{3})}{4\sqrt{2}} \tag{5.14}$$

The fluctuation coefficients for the *Daub4* wavelet transformation are

$$\beta_1 = \frac{(1-\sqrt{3})}{4\sqrt{2}}, \beta_2 = \frac{(\sqrt{3}-3)}{4\sqrt{2}}, \beta_3 = \frac{(3+\sqrt{3})}{4\sqrt{2}}, \beta_4 = \frac{(-1-\sqrt{3})}{4\sqrt{2}} \tag{5.15}$$

The trend and fluctuation coefficients of the *Daub4* wavelet have been represented in Fig. 5.3.

The second major difference in the implementation of a *Daub4* wavelet transformation is in the handling of the wavelet *wrap-around-effect*. The Haar transformation coefficients depend on the support of two adjacent non-zero values of the discrete input vector and therefore there is no wavelet *wrap around effect*. The *Daub4* wavelet coefficients depend on the support of four non- zero adjacent values in the discrete input vector and therefore a wrap around needs to be defined as

$$f_{N+n} = f_n \tag{5.16}$$

assuming that the discrete input vector *f* is periodic.

The major advantage of the Daubechies wavelet is that they are able to detect very short lived transient fluctuations, thereby exhibiting greater sensitivity to the inputs. (Walker 1999). Daubechies class of wavelets are also smooth in comparison to the Haar wavelet. For image texture analysis, strong localization properties in the Fourier domain are desired for applications in image filtering (Kumar and Georgiou 1997). Although a Haar wavelet is conceptually simple and easy to implement, its spectrum is not well localized (Kumar and Georgiou 1997). However, the Daubechies class of wavelets have excellent localization properties in the Fourier domain and are therefore very popular in the areas of image texture analysis and image compression (Walker 1999; Choi and Baraniuk 2001; Abasolo and Perales 2003; Arivazhagan and Ganesan 2003; Daugman 2003; Bartels, Wei, and Mason 2005; Clausi and Deng 2005; Kandaswamy, Adjeroh, and Lee 2005; Arivazhagan, Ganesan, and Priyal 2006; Wei and Bartels 2006).

The general algorithm for the implementation of a Haar and Daubechies wavelet forward transformation is given below:

1. Read the input image *I(x, y)*.
2. For every row *R* in the image *I(x, y)* do:
    a) Apply the wavelet transformation (trend and fluctuations) on *R* to get trend and fluctuation coefficients.
    b) Add all trend coefficients to *L* and all the fluctuation coefficients to *H* (low pass and high pass filters).

3. For every column in the trend matrix *L* do:
    a) Apply the wavelet transformation (trend and fluctuations) on *L* to get trend and fluctuation coefficients.
    b) Add all trend coefficients to *LL* and all the fluctuation coefficients to *LH* (low pass and high pass filters).
4. For every column in the fluctuation matrix *H* do:
    a) Apply the wavelet transformation (trend and fluctuations) on *H* to get trend and fluctuation coefficients.
    b) Add all trend coefficients to *HL* and all the fluctuation coefficients to *HH* (low pass and high pass filters).
5. End

The general algorithm for the implementation of a Haar and Daubechies inverse wavelet transformation is given below:

1. Read the LL and LH images.
2. For every row R_i and R_j in LL and LH respectively do:
    a) Apply reverse transformation using respectively R_i and R_j to get row $L_{i.}$
3. For every row R_i and R_j in HL and HH respectively do:
    a) Apply reverse transformation using respectively R_i and R_j to get row $H_{i.}$
4. For every row R_i and R_j in L and H respectively do:

a) Apply reverse transformation using respectively R_i and R_j to get image row $I_{i.}$

5. End

The software tool implemented using Microsoft Visual C# and ESRI ArcObjects are in Figures 5.4 - 5.5. The detailed implementation for the 2-D *Haar* and *Daub4* wavelet forward and inverse transformations are explained in Fig. 5.6.

5.5. Multi-scale Classification Model

The input image is transformed by using a suitable wavelet transform without any sub-sampling, to generate a *set of wavelet decomposed images called wavelet frames* which have the same dimensions of that of the input image (Unser 1995). Statistical texture indicators like gray level co-occurrence matrices and non-linear operators like texture energy are used to derive *image texture* indicators using the trend and fluctuation images.

The non-linear operator for deriving texture energy can be mathematically represented as,

$$e(x,y) = \frac{1}{M^2} * \sum_{a,b \,\epsilon\, W} |\gamma(I(a,b))| \tag{5.17}$$

where $\gamma\,(.)$ is a non-linear function defined in equation 5.18, and W is a M*M window, centered at a pixel with co-ordinates x, y.

$$\gamma\,(t) = \tanh(\alpha t) = \frac{1-e^{-2\alpha t}}{1+e^{-2\alpha t}} \tag{5.18}$$

where α is a constant and is chosen to be 0.25 as suggested by Farrokhia and Jain (1991).

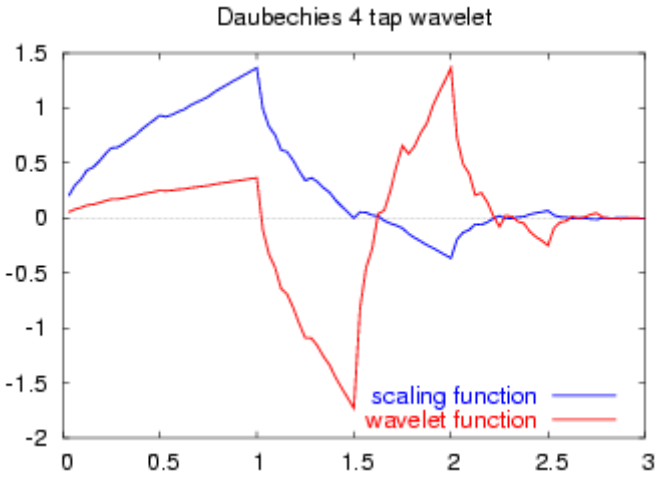


Figure 5.3 – Scaling (trend) and Wavelet (fluctuation) functions of *Daub4* wavelet

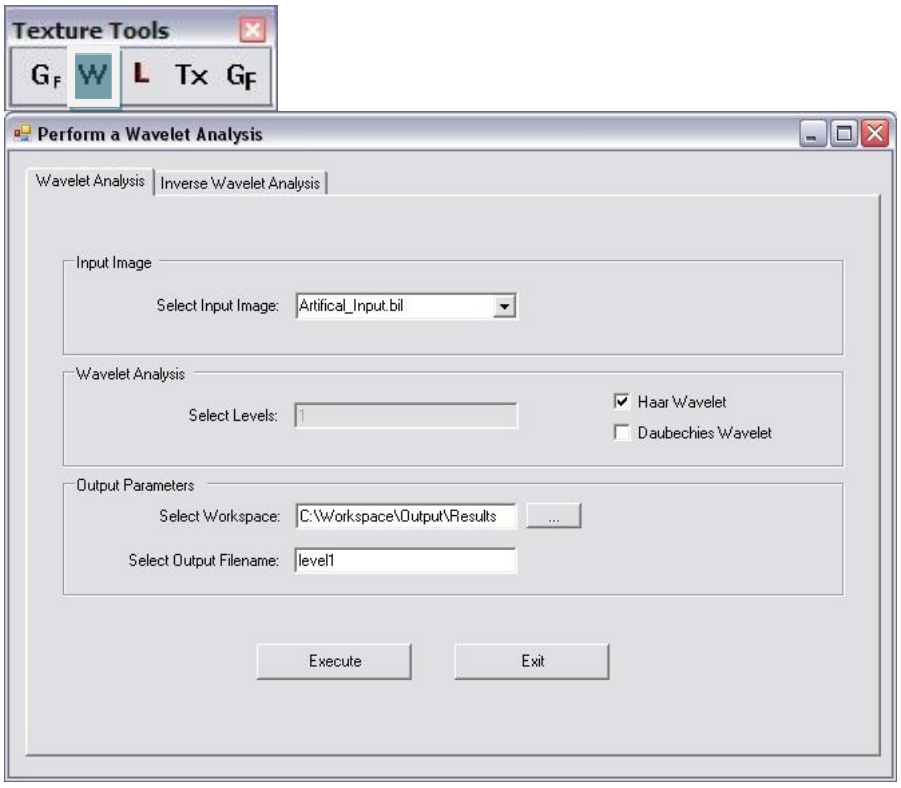


Figure 5.4 – User interface for generating a forward wavelet transformed image

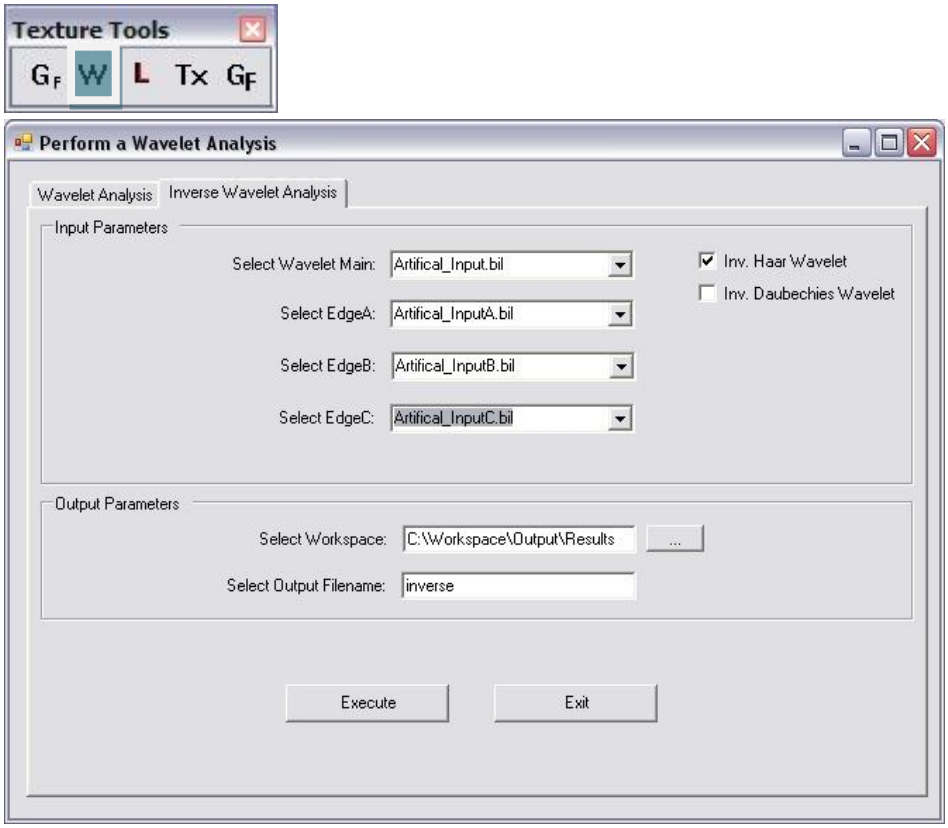


Figure 5.5 – User interface for generating an inverse wavelet transformed image

```
//Performs the 2D Wavelet Tranformation for the Image
perform_wavelet2D(image, nrows, ncols){

      //Perform a 1-D Wavelet Transformation in the horizontal_
      //direction

      //Steps : Read Column By Column
      //      : Perform Wavelet Transformation 1D
      //      : Split and Insert As Column in 2 float[nrows,halfNCols]'s
     //Creating Intermediate Matrices

   float [,] A = new float[halfNRows,ncols]
   float[,] B = new float[halfNRows, ncols]

      for(i=0; i<ncols; i++)
      {
            //Get Image Column
            float [] vector = get_image_column(i)

            //Compute wavelet 1D
            compute_wavelet_1D(vector, trend, fluctuation);

            //Save in intermediate matrices
            set_as_column(A, trend, i)
            set_as_column(B, fluctuation, i)
      }

   //Perform a 1-D wavelet Transformation in the vertical_
      //direction

      //Creating Final Matrices
      waveletResult = new float[halfNRows, halfNCols]
      edgeA = new float[halfNRows, halfNCols]
      edgeB = new float[halfNRows, halfNCols]
      edgeC = new float[halfNRows, halfNCols]

      int k = 0;
      for (k = 0; k < halfNRows; k++)
      {
            //Getting Row Vector for Intermediate Matrix
            float [] vector = get_image_row(A, k)

            compute_wavelet_1D(vector, trend, fluctuation);

            //Set As Rows
            set_as_row(waveletResult, trend, k)
            set_as_row(edgeA, fluctuation, k)
      }
```

Figure 5.6 – Core implementation of the wavelet transform

```
int l = 0;
	for (l = 0; l < halfNRows; l++)
	{
		//Getting Row Vector for Intermediate Matrix
		float [] vector = get_image_row(B, l)

		compute_wavelet_1D(vector, trend, fluctuation);

		//Set As Rows
		set_as_row(edgeB, trend, k)
		set_as_row(edgeC, fluctuation, k)
	}

	//Completed 2-D wavelet Transformation
}

//Performs the inverse 2-D wavelet Transformation
//waveletMain (LL)
//edgeA (LH)
//edgeB (HL)
//edgeC (HH)
perform_inverse_wavelet2D(waveletMain, edgeA, edgeB, edgeC){

	length = get_length(waveletMain)

	 //Create Intermediate Matrices twice the size of the sub-signals
	float[,] A = new float[nrows, 2 * ncols]
	float[,] B = new float[nrows, 2 * ncols]

	//Combine waveletMain and edgeA
	//Combine edgeB and edgeC

	for (i = 0; i < nrows; i++)
	{
		//Getting Row By Row
		float[] waveletRow = get_image_row(waveletMain, i)
		float[] egdeARow = get_image_row(edgeA, i)

		float[] edgeBRow = get_image_row(edgeB, i)
		float[] edgeCRow = get_image_row(edgeC, i)

		compute_inverse_wavelet_1D(waveletRow, edgeARow, resultRowA)
		compute_inverse_wavelet_1D(edgeBRow, edgeCRow, resultRowB)

		//Set Results to Intermediates
		set_as_row(A, resultRowA, i)
		set_as_row(B, resultRowB, i)
	}
```

Figure 5.6 (Continued)

```
//Combine the intermediate matrices to get the result

      float[,] result = new float[nrows * 2, ncols * 2]
       for (nk = 0; nk < ncols * 2; nk++)
            {
                  float[] AColumn = get_image_column(A, nk)
                  float[] BColumn = get_image_column(B, nk)

                  compute_inverse_wavelet_1D(AColumn, BColumn, resultColumn)

                  set_image_column(result, resultColumn, nk);
            }

      //Completed Inverse wavelet2D
      return result
}
```

Figure 5.6 (Continued)

5.7 Preliminary Results with Artificial Textures

To test the statistical indicators derived from a wavelet transforms, an artificial image consisting of three dominant textured regions derived from the Brodatz texture album is used. The input image is a combination of textures *D16, D49* and *D53***.** The input image used is shown in Fig. 5.7. The input image consists of a high spatial frequency component in the centre and two low spatial frequency components with orientations at $0^o$and 90^orespectivily.

A first level Daubechies decomposition of the input image is shown in Figures 5.8-5.11.

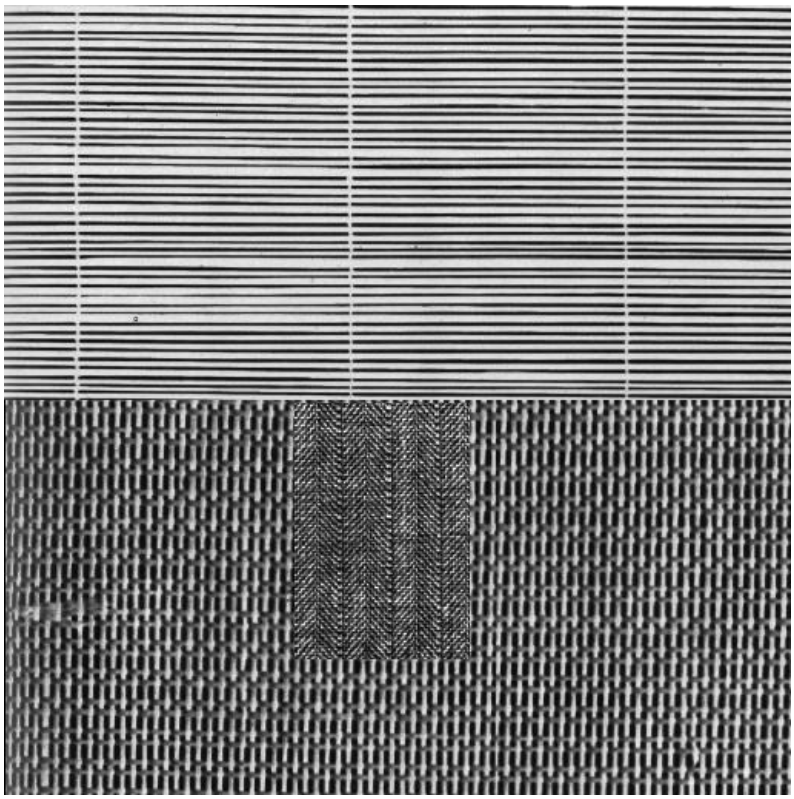


Figure 5.7 – Input image to test wavelet based indicators

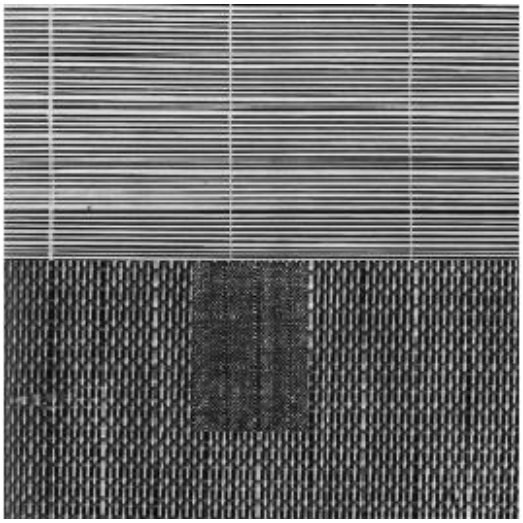


Figure 5.8 – Wavelet decomposition (Average)

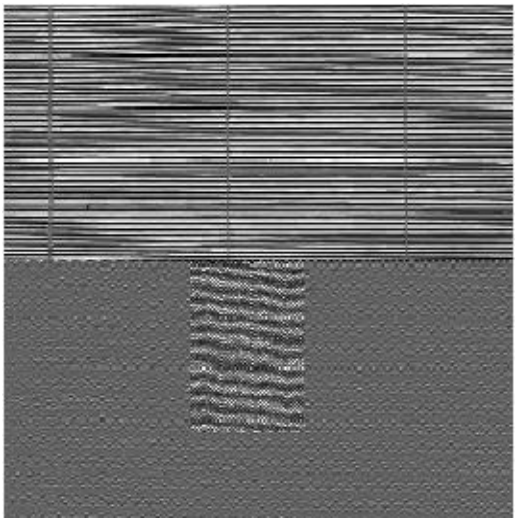


Figure 5.9 - Wavelet fluctuation (Horizontal)

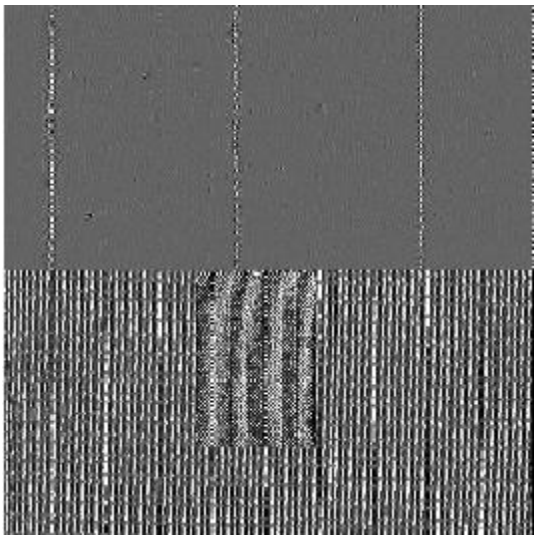


Figure 5.10 – Wavelet fluctuation (Vertical)

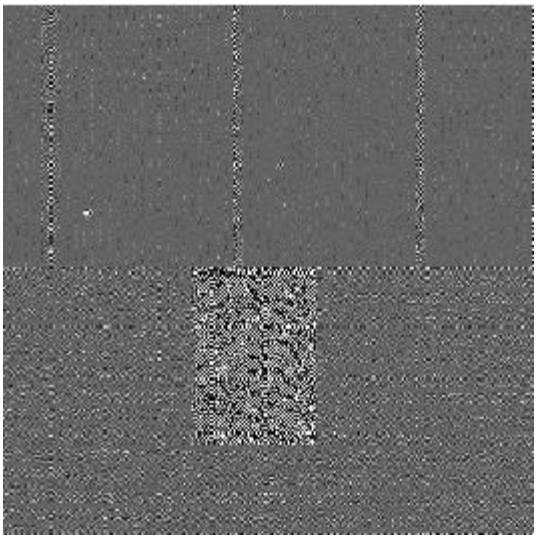


Figure 5.11 - Wavelet fluctuation (Diagonal)

The classification results using the wavelet based classification model and traditional gray level thresholding are shown in Figures 5.12 and 5.13. Clearly the results using the wavelet based classification model are far superior to that of traditional methods.

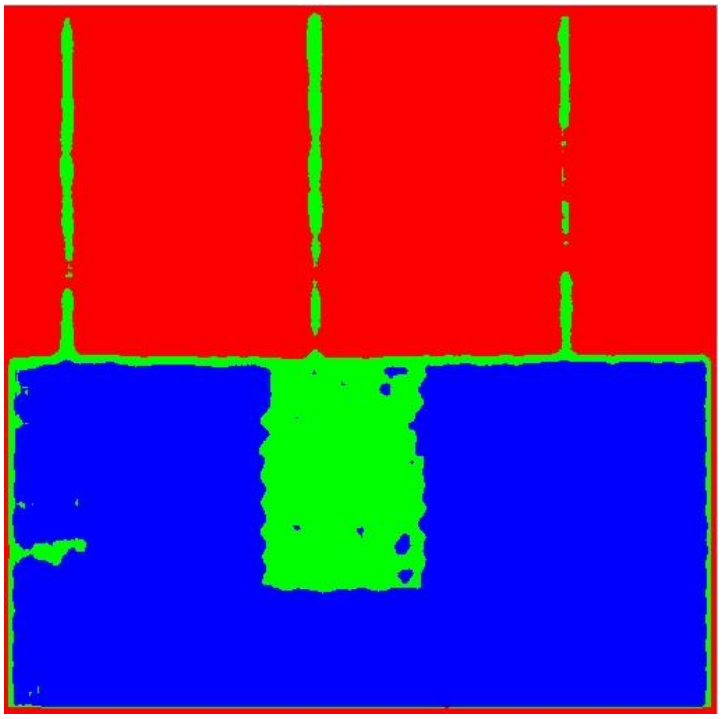


Figure 5.12 – Classification results (wavelet based classification model)

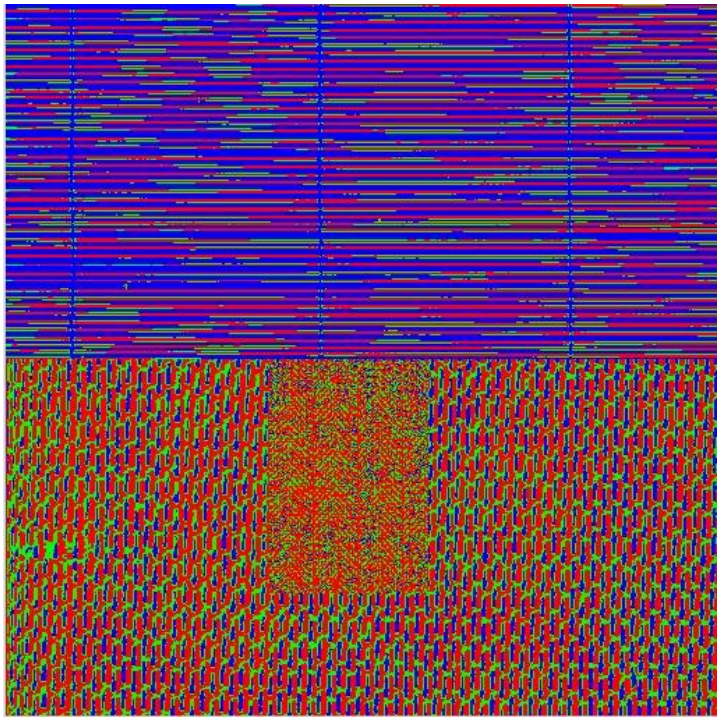


Figure 5.13 – Classification result derived from gray scale thresholding approach

CHAPTER VI

EXPERIMENTS AND COMPARISON RESULTS

A comprehensive evaluation of various texture processing techniques is performed using natural images and the results of applying these techniques are presented. In the final section a comparison between color-only classification and texture aided classification is presented.

6.1 Texture Processing Techniques on Natural Images

The input remotely sensed image (natural) exhibiting strong textural characteristics is given in Fig. 6.1. This image represents a small region in Texas and is acquired by Landsat at a spatial resolution of 30 m. Image spectral information is restricted to the visible region (R, G, and B).

This region is predominantly covered by dense and sparse vegetation, which exhibit interesting forms of texture at different spatial frequencies and orientations. There are also some linear features in the form of roads.

Arid and fallow lands are present in this input image. Fig. 6.2 represents the ideal classification result for the given input image.

The ISODATA classification result using spectral information alone is given in Fig. 6.3. The classification parameters used are (No of classes = 6-8, Iterations = 3, minimum number of pixels per class = 50 and max. standard deviation in class = 2). This result is with a lot of salt and pepper noise, due to the confusion between various classes arising due to textural characteristics.

The classification result using GLCM indicators with parameters (mask size 7*7, d = 1, α = 0, -45) is given in Fig. 6.4. From Fig. 6.4, it is clear that a lot of the error in classification has been resolved by using texture indicators.

The classified result using Laws texture indicators on the input image is shown in Fig. 6.5. The *L5E5, L5S5* and *L5R5* convolution masks were used. Although Laws serves as an excellent textural edge detector its results are not very suitable for ISODATA classification.

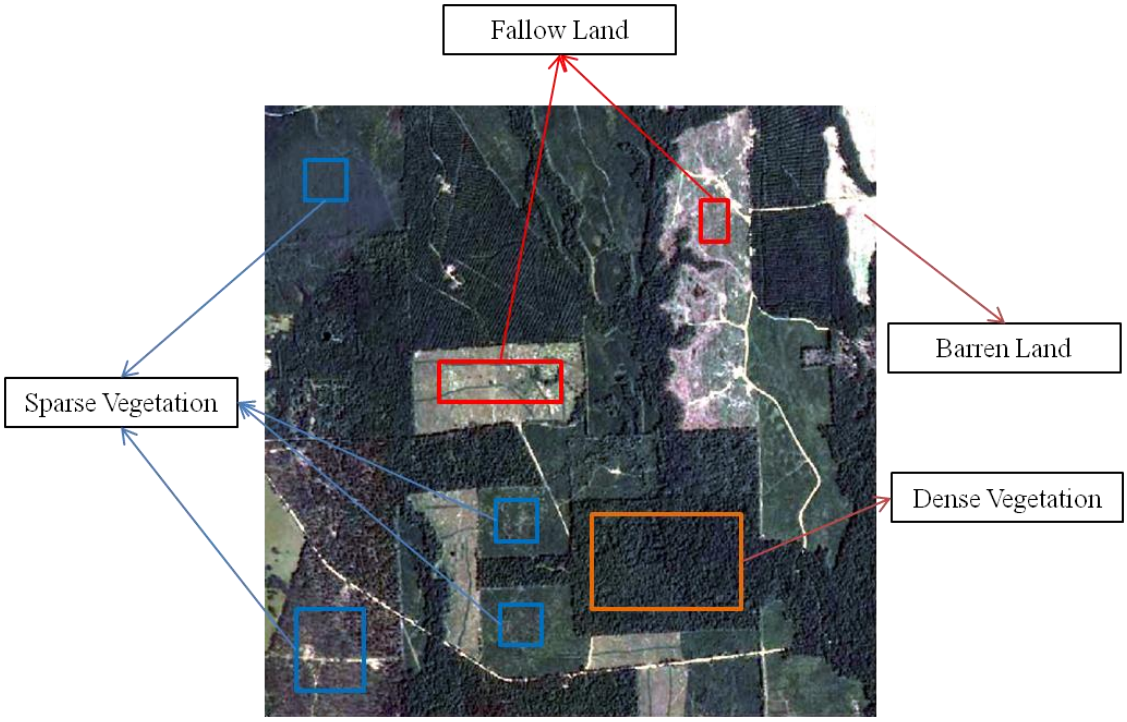


Figure 6.1 – Input Landsat image

Figure 6.2 – Classification result (manual)

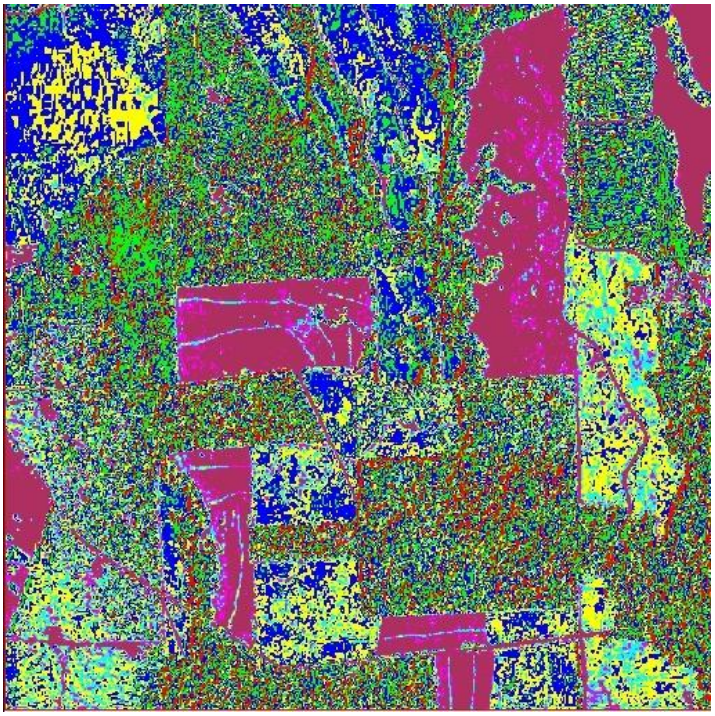


Figure 6.3 – Classification result (color only)

Figure 6.4 – Classification result (GLCM)

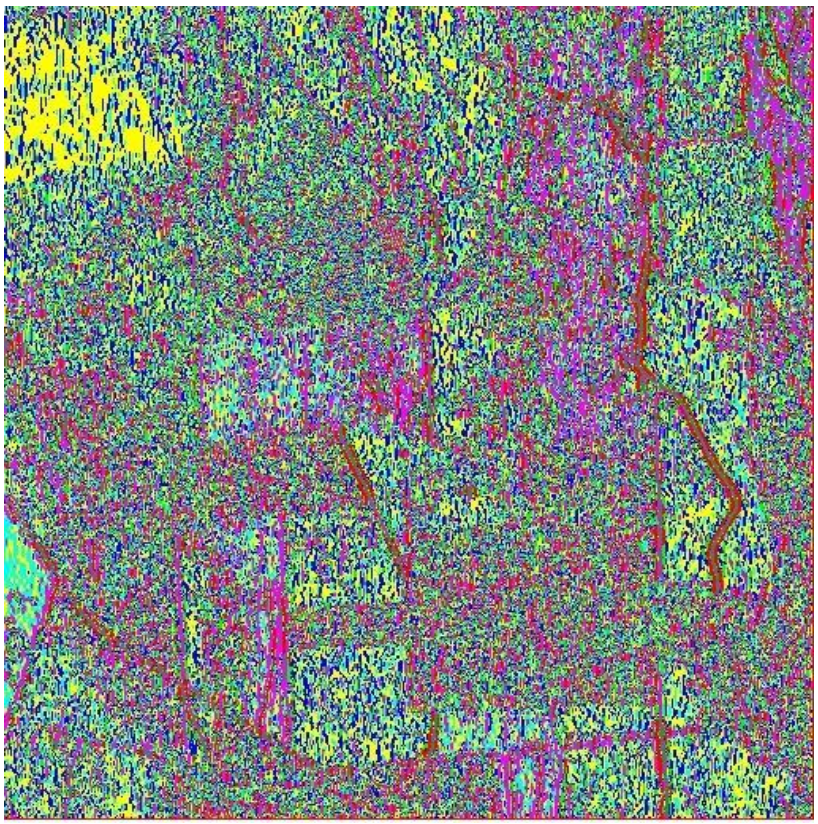


Figure 6.5 – Classification result (Laws masks)

A series of Gabor filtered image channels are shown in Figures 6.6 (a) – (d). The Gabor filter parameters for Fig. 6.6 (a) are (frequency = $1/8\sqrt{2}$, angle = 0, phase = 0, standard deviation of the filter = π). The Gabor filter parameters for Fig. 6.6 (b) are (frequency = $1/10\sqrt{2}$, angle = 90, phase = 0, standard deviation of the filter = π). The Gabor filter parameters for Fig. 6.6 (c) are (frequency = $1/15\sqrt{2}$, angle = 45, phase = 0, standard deviation of the filter = π). The Gabor filter parameters for Fig. 6.6 (d) are (frequency = $1/20\sqrt{2}$, angle = 0, phase = 0, standard deviation of the filter = π).

The classification output using a Gabor filtered images with parameters (frequency = $1/8\sqrt{2}$, and $1/20\sqrt{2}$, orientation = 0, 45, phase = 0, standard deviation of the filter = π) is shown in Fig. 6.7. The classification result derived using GLCM indicators (mean and angular second moment) on a second level Daubechies wavelet decomposition is shown in Fig. 6.8.

A natural image chip corresponding to Antigua Island is shown in Fig. 6.9. This image is an IKONOS image has a spatial resolution of 4 m. Image spectral information is available in visible bands and NIR bands.

This image consists of high spatial frequency texture components corresponding to coral reefs, dense vegetation and has a medium-low spatial frequency component corresponding to sparse vegetation. Some urban features in the form of an airport are present in the image.

The results of a supervised classification on the original IKONOS image corresponding to Antigua Island are shown in Fig. 6.10.

The classified result using spectral information of the image is represented in Fig. 6.11. The ISODATA classification parameters used are (no. of classes = 5-6, iterations = 3, min. no. of pixels per class = 100 and std. deviation within a class = 2).

The classified result using GCLM indicators (angular second moment and mean) on Band 3 of the IKONOS image is shown in Fig. 6.12. This result helps us differentiate between shallow and deep water bodies, urban areas and fallow land. However there is some amount of confusion between vegetation and water bodies.

The classification result using Laws texture indicators (L5E5, L5S5) on the input IKONOS image is shown in Fig 6.13. Although Laws masks are excellent for identifying textural discontinuities, the classification results are inferior compared to other texture indicators used.

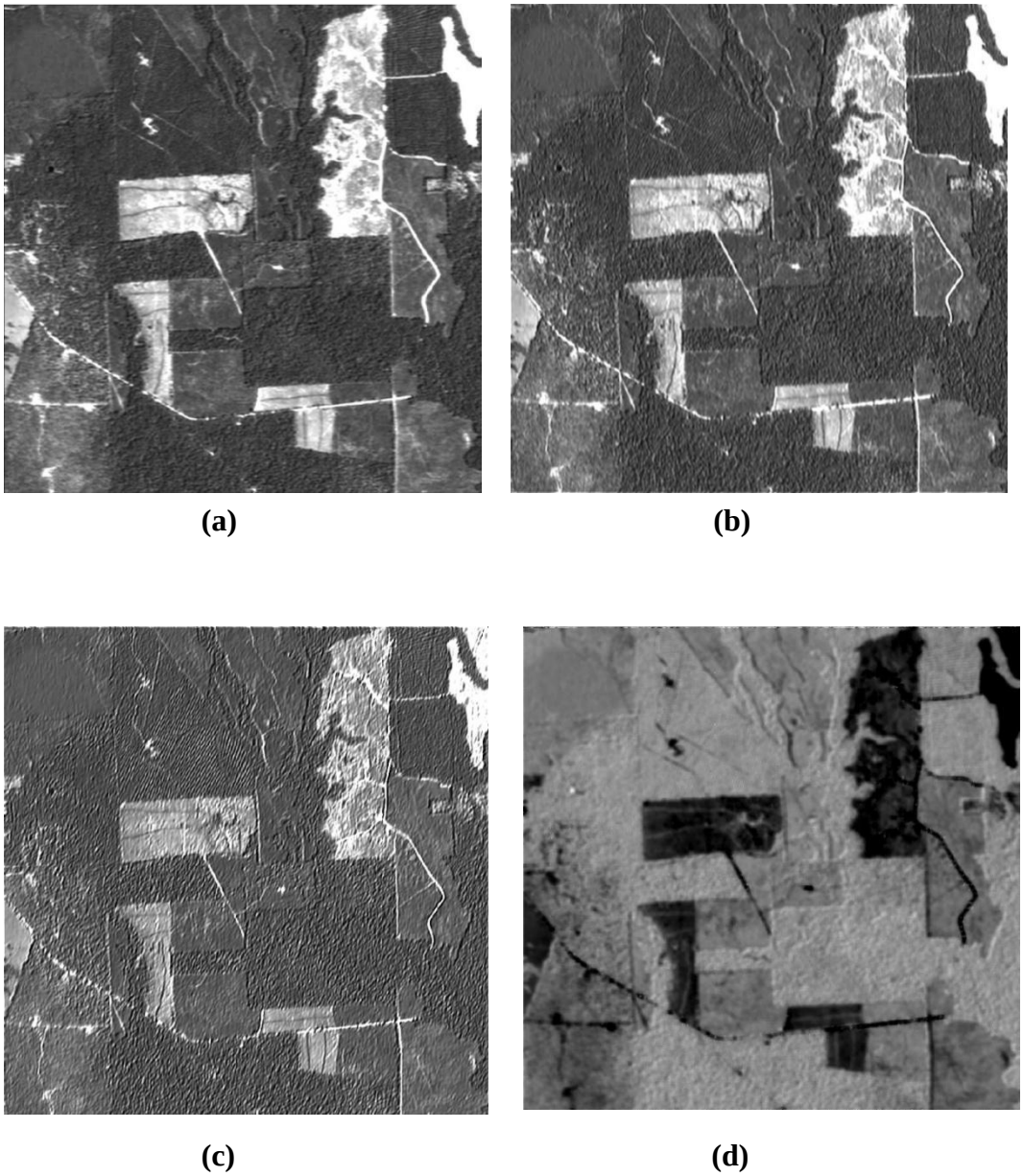
The classification result using Gabor filtered images are represented in Fig. 6.14. The filter parameters used are (frequency = $1/15\sqrt{2}$, orientation = 0, 90 and phase = 0, standard deviation of the filter = π).

The classified result using GLCM indicators (mean and angular second moment) on a first level Daubechies decomposition on the input Antigua image is shown in Fig. 6.15.

A natural image chip, corresponding to an urban scene in St. Johns, Antigua has been used test texture analysis of urban features. This image is shown in Fig. 6.16. The aim of the classification process in this image is to capture the macro scale urban texture. When this image chip is classified using color information alone, there is a *salt and peppery* nature to the classified output (Fig. 6.17).

The classification result using GLCM indicators mean, dissimilarity and angular second moment (window size = 11, d = 1, and α = 0, 90) is given in Fig. 6.18. Laws texture indicators are not used for the analysis of this image, as Laws masks are not suitable for analysis of macro level textures; given that they are specific to a given scale and orientation.

The classification result using Gabor filter (frequency = $1/25\sqrt{2}$, orientation = 0 and phase = 0, standard deviation of the filter = π) is given in Fig. 6.19. The classification result derived from using 2 level Daubechies wavelet transform and post processed with non-linear texture energy function (M = 5, α = 0.25) is given in Fig. 6.20.



(a) **(b)**

(c) **(d)**

Figure 6.6 –Gabor filter bank

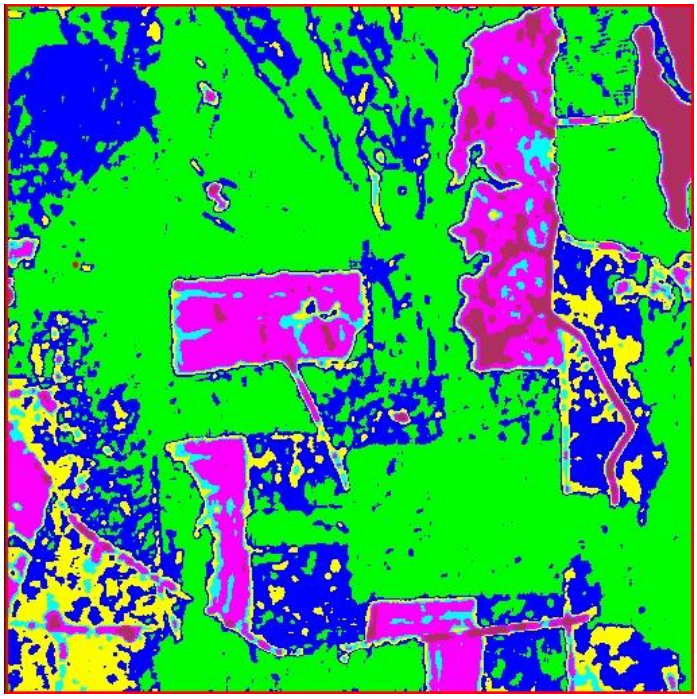


Figure 6.7 – Classification result (Gabor)

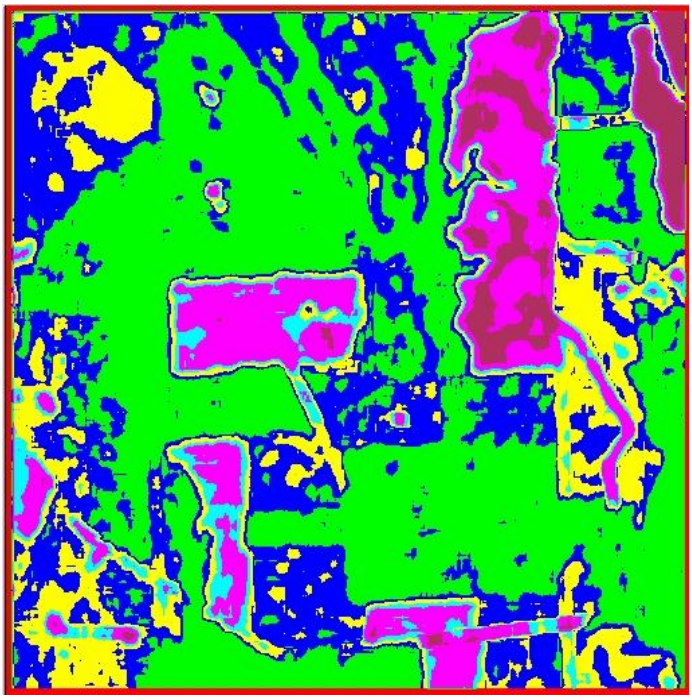


Figure 6.8 – Classification result (wavelet)

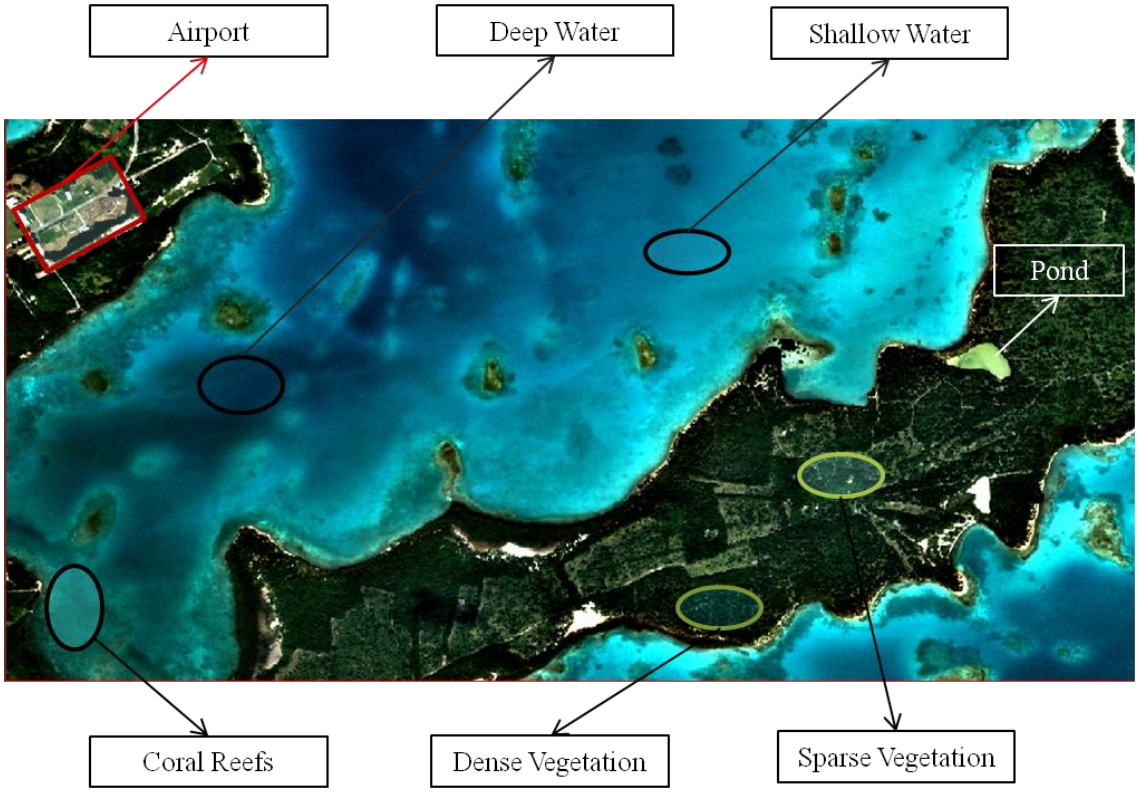


Figure 6.9 – Original image (Antigua) [Quickbird]

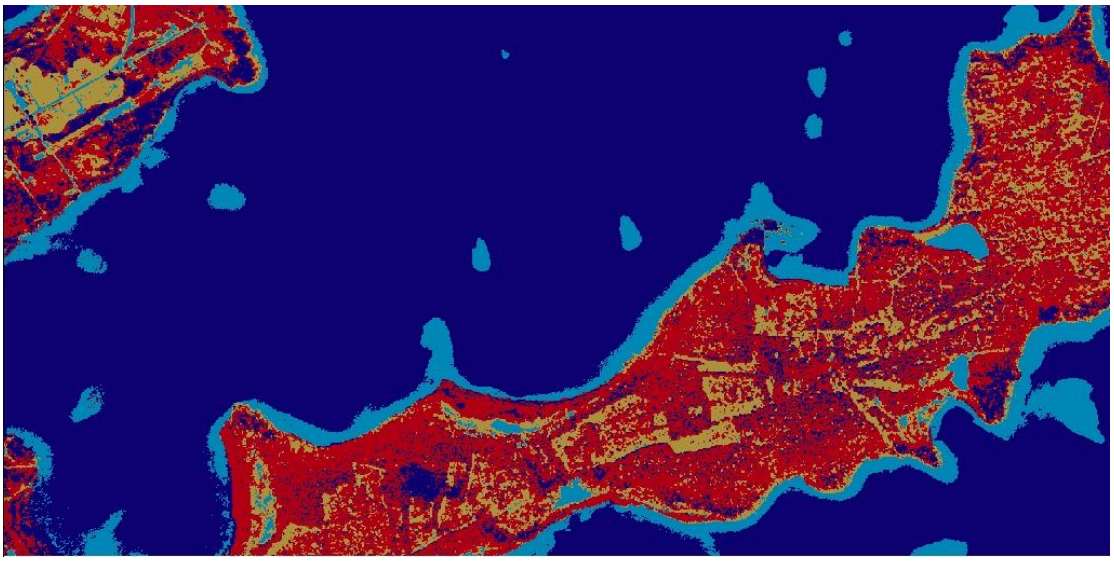


Figure 6.10 – Classification result (supervised)

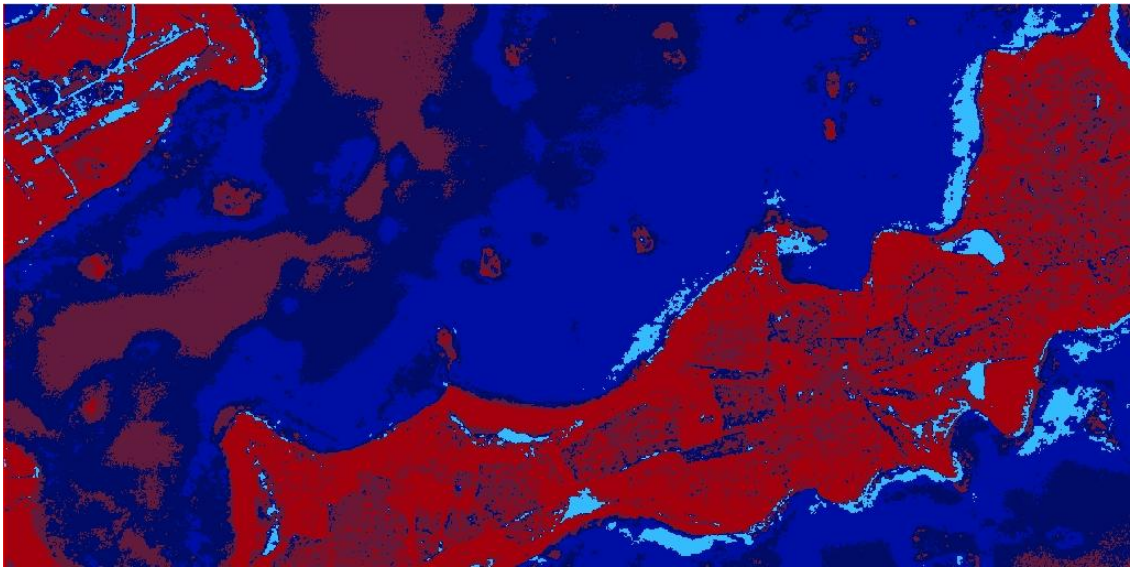


Figure 6.11 – Classification result (color only)

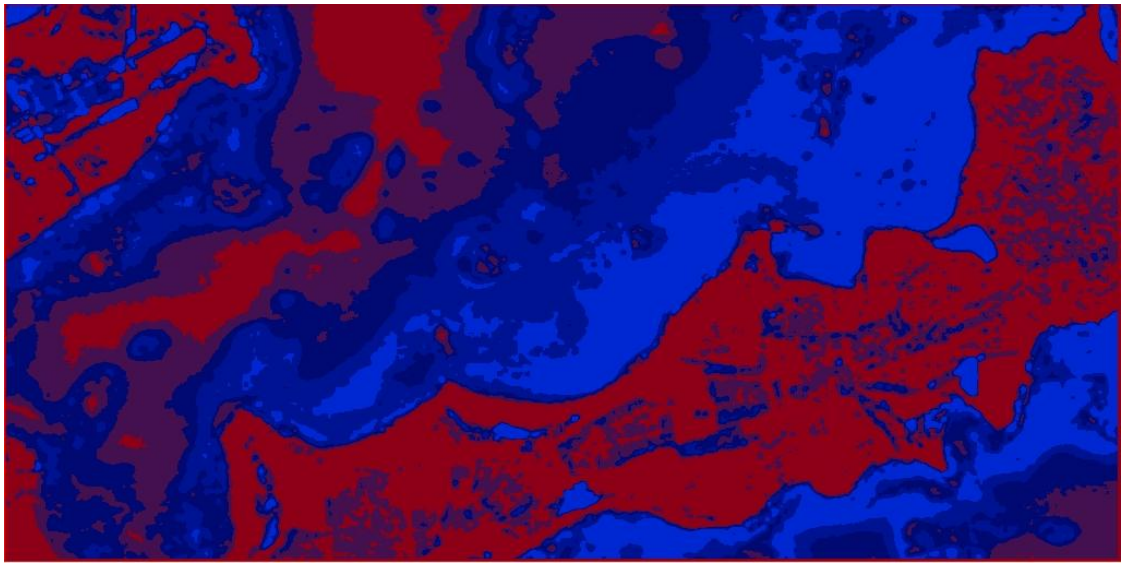


Figure 6.12 – Classification result (GLCM)

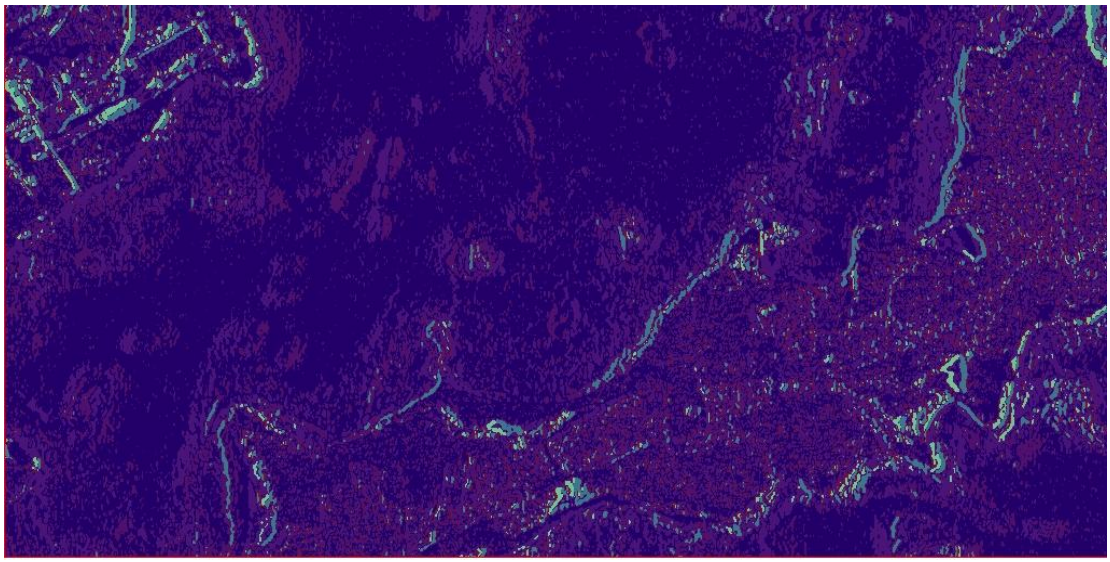


Figure 6.13 – Classification result (Laws masks)

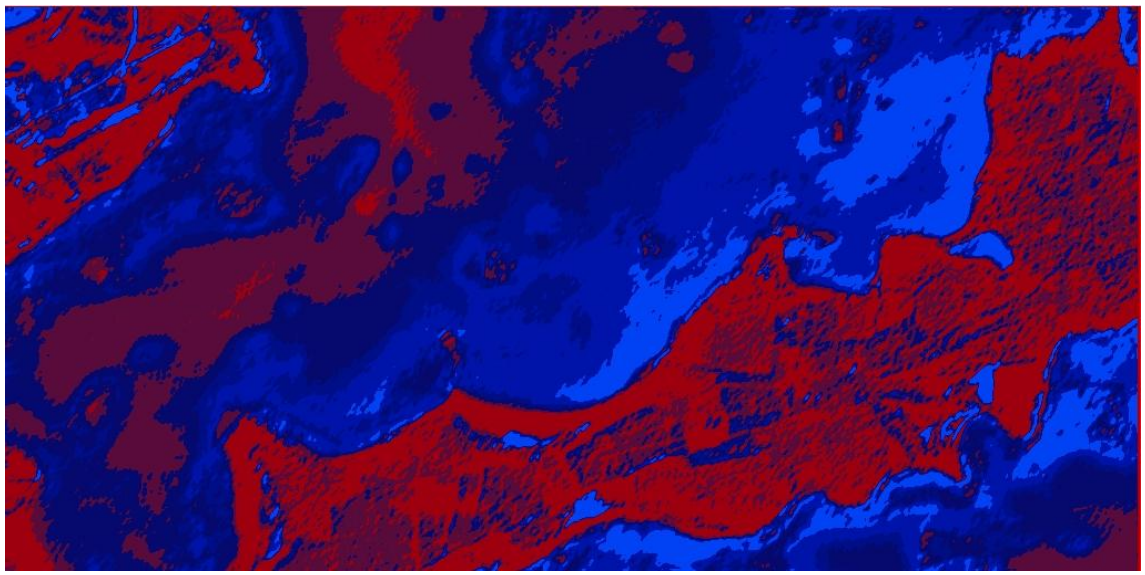


Figure 6.14 – Classification result (Gabor)

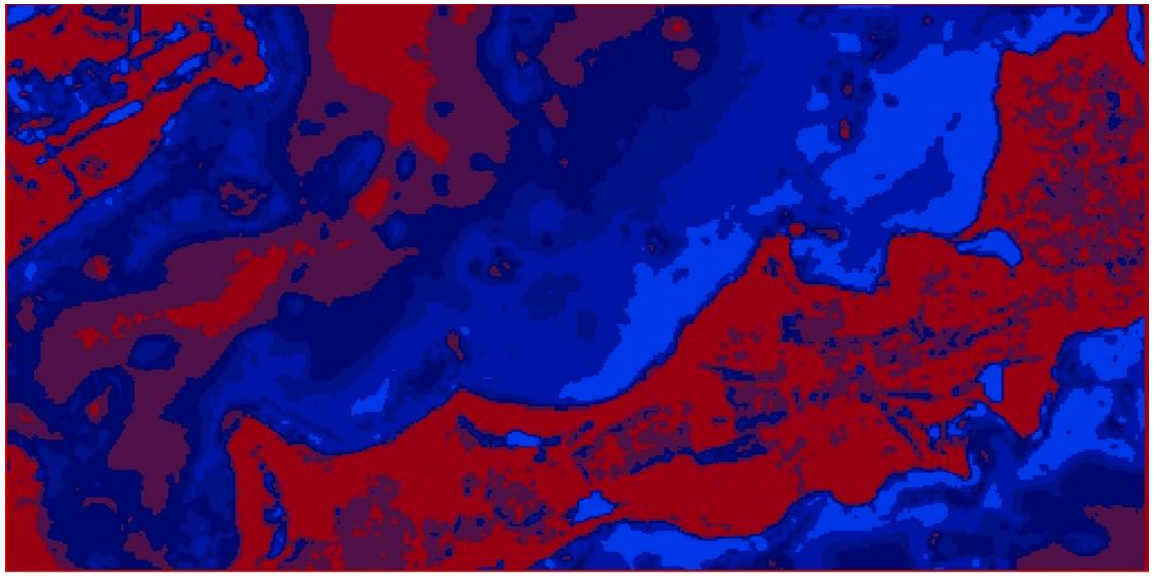


Figure 6.15 – Classification result (wavelet)

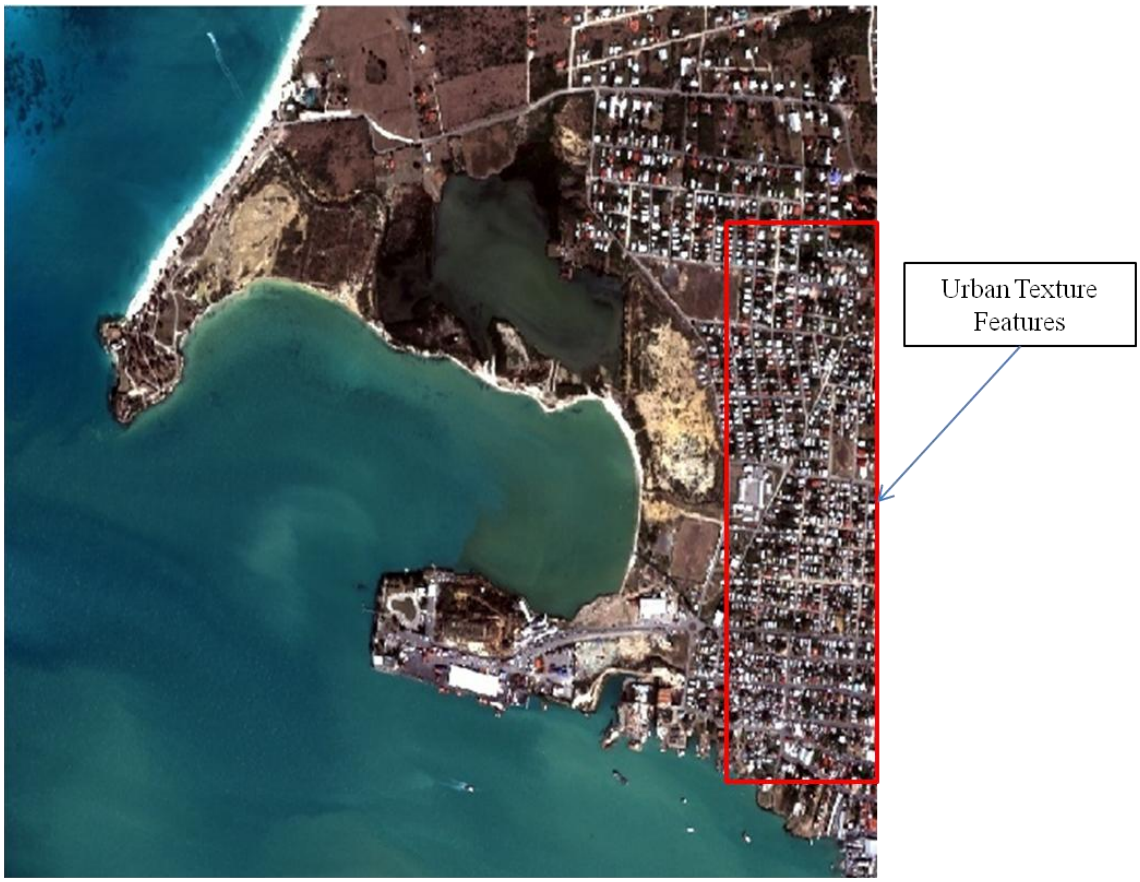


Figure 6.16 – Urban texture features (St. Johns, Antigua)

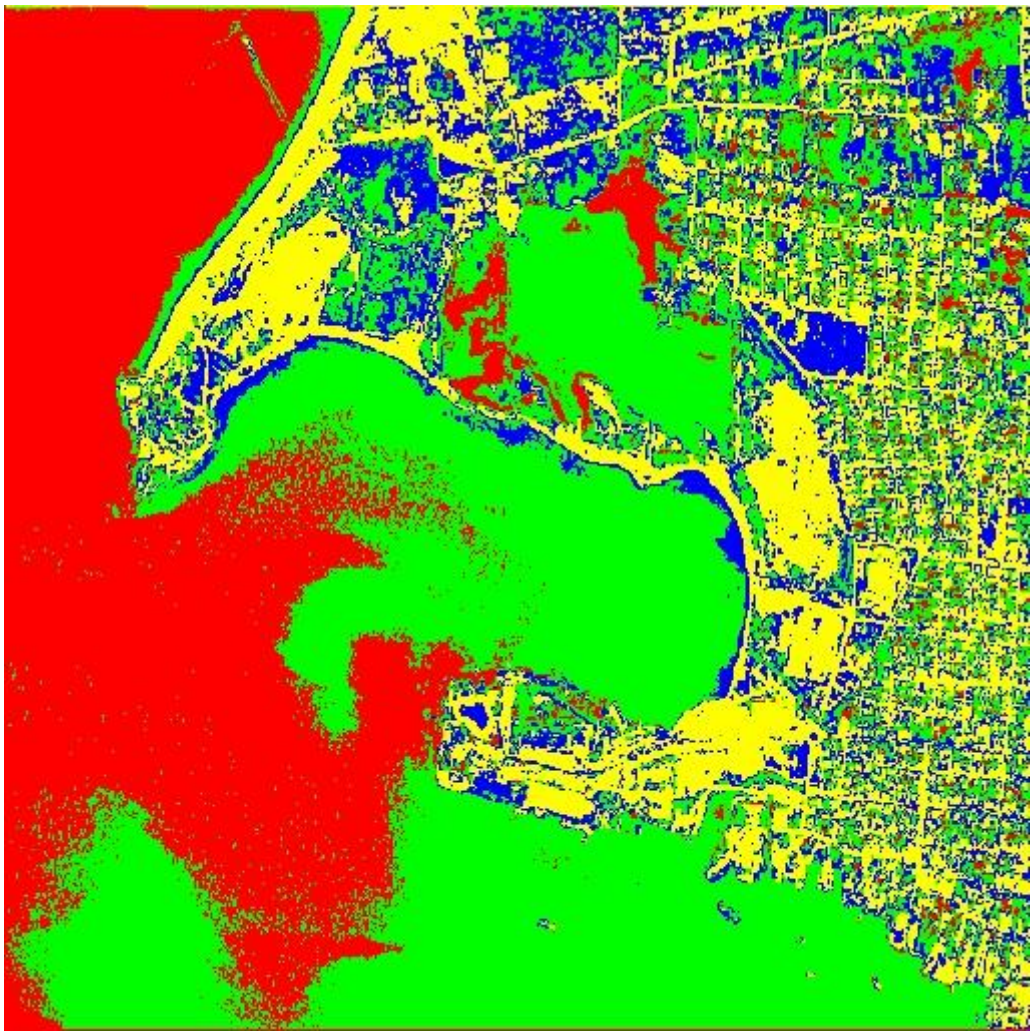


Figure 6.17 – Classification result (color only)

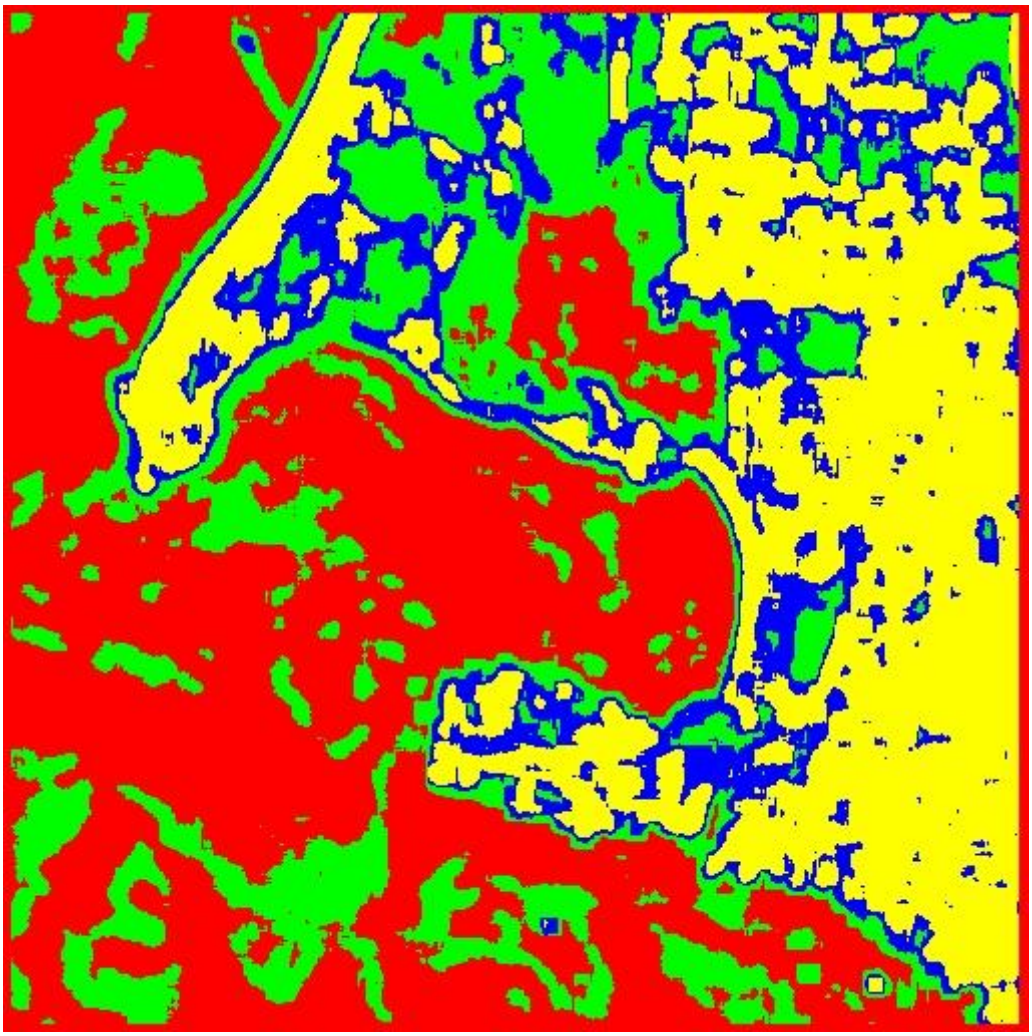


Figure 6.18 – Classification result (GLCM)

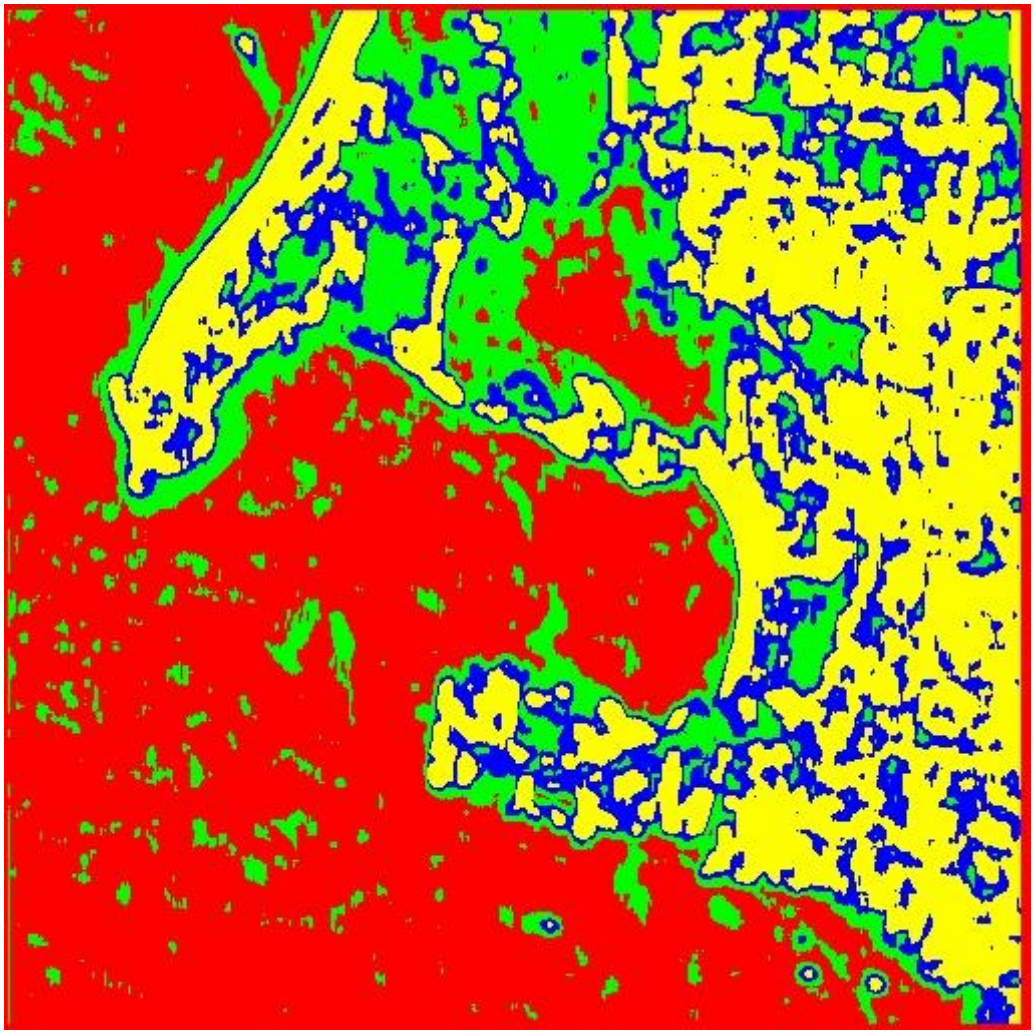


Figure 6.19 – Classification result (Gabor filters)

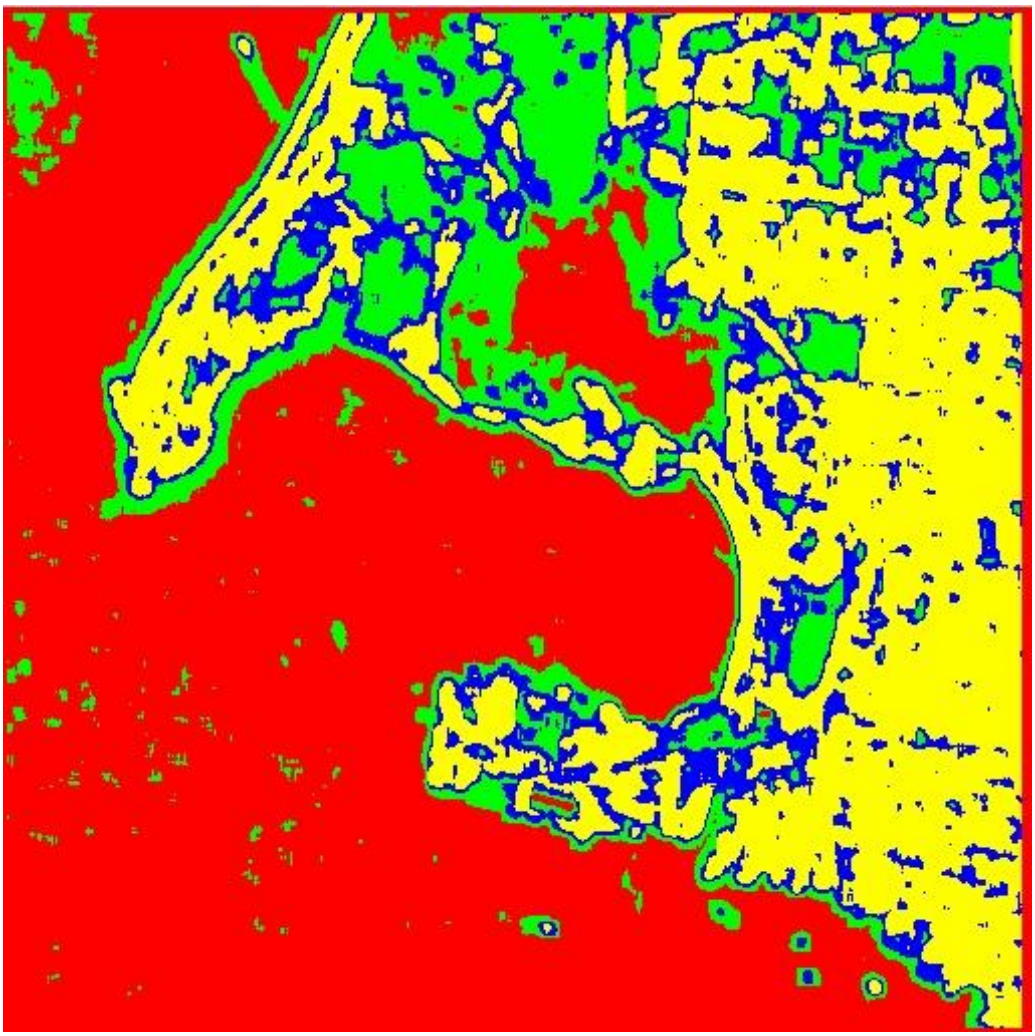


Figure 6.20 – Classification result (wavelets)

6.2 Comparison of Results

6.2.1 Landsat Image

For the natural Landsat image, the results of the classification are summarized in Fig. 6.21. In Fig. 6.21 (c), where only color information is used for the classification of the image, there considerable confusion in classes where texture information is prominent especially in the case of dense and sparsely vegetated lands. As a result, there is a *salt and pepper effect* in the classification result. However, when texture indicators derived from gray level co-occurrence matrices, Gabor filter banks and wavelet transforms are used, classification is improved. The overall classification result is superior when multi-scale image texture analysis techniques (Gabor filter banks and wavelets) are used (Table 6.1).

6.2.2 Antigua (Quickbird image)

The classification results for the image corresponding to Antigua Island are shown in Figures 6.22 and Fig. 6.23. This image consist of dense, sparse vegetation, urban features in the form of an airport and a road network, ponds, coral reefs and variable reflectance characteristics of both deep and shallow water. With conventional classification results, the differences in the electromagnetic reflectance due to *variable depths* of water are not fully captured, but using texture indicators this difference is highlighted in the classification result.

6.2.3 St. Johns (Quickbird Image)

All classification results for the Quickbird image corresponding to St. Johns, Antigua are show in Fig. 6.24. With conventional classification, macro texture features are not captured at a macro scale resulting in a salt and peppery classification result. However when texture indicators are used, these macro textures are captured. The classification result using GLCM indicators suffer from an edge effect due to the increase in the window size used for the estimation of the GLCM indicators. On the other hand, these artifacts are not apparent in multi-texture texture analysis techniques.

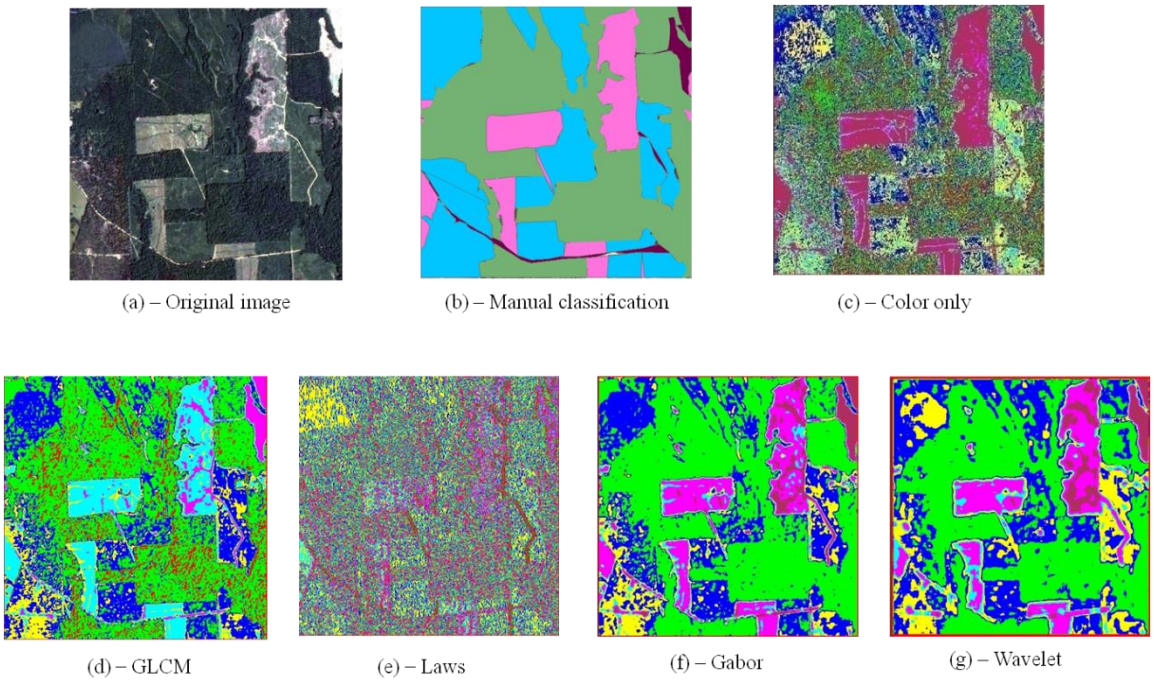
6.3 Overall Classification Accuracy

The overall accuracy of classification is estimated at a *macro* scale or at a *neighborhood* level using manually classified *reference images* thereby differentiating between major land cover classes in the image, while ignoring minor details. Classes are masked out of the manually classified image and used as *ground truth* images to estimate classification accuracies. The classification accuracy is measured as the ratio of pixels correctly classified by the ISODATA classifier vs. the pixels for the same class in the ground truth image.

The overall classification accuracies for the classified results have been summarized in Table 6.1. It is clear from the table that classified results from multi-scale texture analysis techniques have an edge over statistical indicators and conventional methods of classification.

Table 6.1 – Overall classification accuracies

Image Used	Overall Classification Accuracy (%)			
	Color	GLCM	Gabor	Wavelet
Landsat(Texas)	47.9	74.1	80.1	79.2
Antigua (IKONOS)	56.3	66.9	80.2	79.5
Urban (Quickbird)	62.6	78.9	87.4	85.9

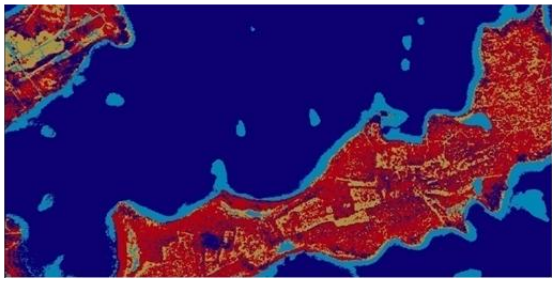


(a) – Original image (b) – Manual classification (c) – Color only

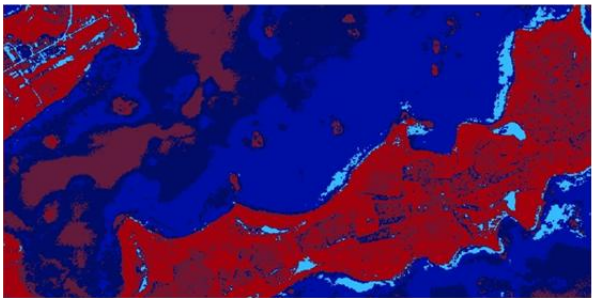
(d) – GLCM (e) – Laws (f) – Gabor (g) – Wavelet

Figure 6.21 – Comparison of classification results – Landsat image

(a) – Original image

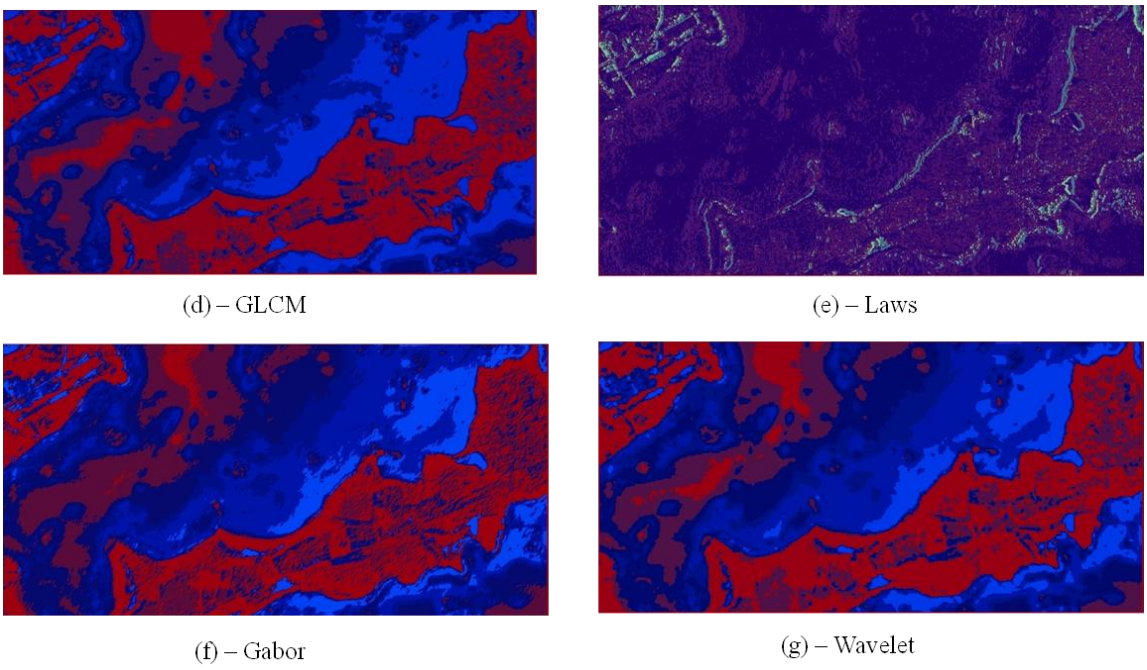


Supervised Classification



(c) – Color

Figure 6.22 – Comparison of classification results



(d) – GLCM (e) – Laws

(f) – Gabor (g) – Wavelet

Fig. 6.23 – Comparison of classification results (texture indicators)

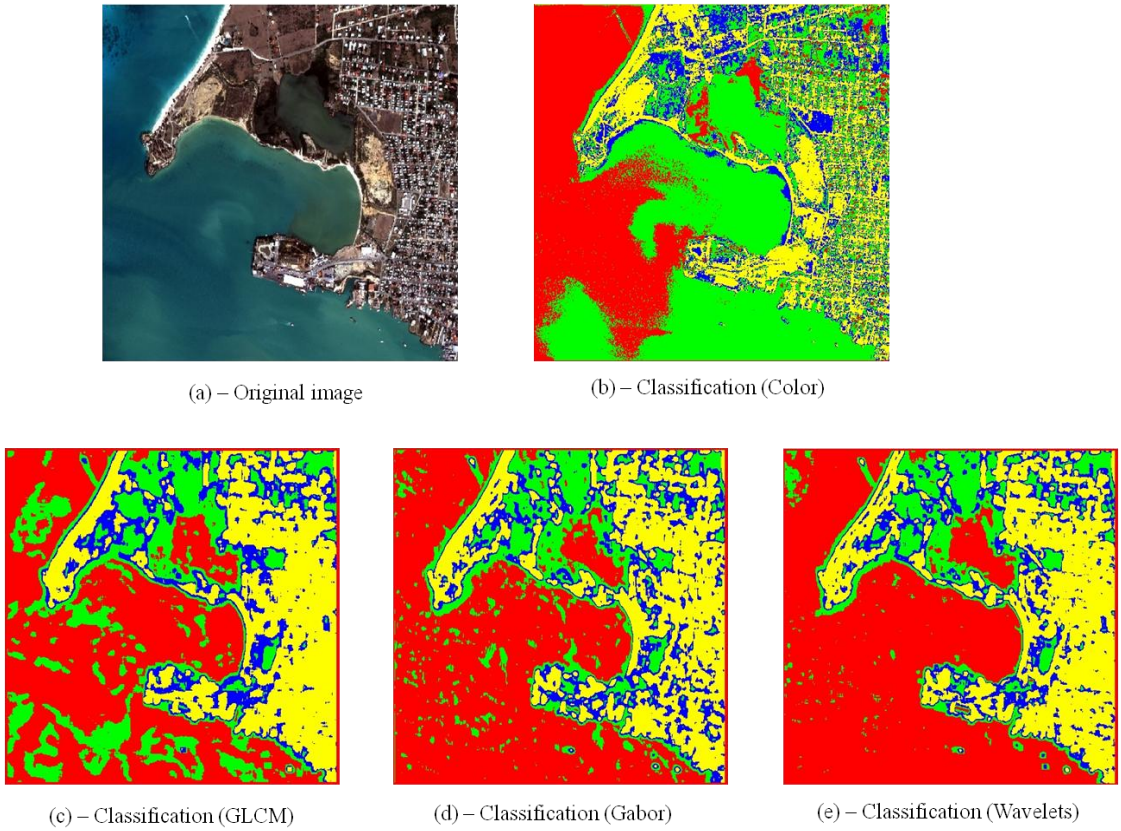


Figure 6.24 – Comparison of classification results (St. Johns, Antigua - Quickbird)

CHAPTER VII

SUMMARY AND CONCLUSIONS

This research has shown that texture can be extracted and incorporated in conventional classification algorithms to improve the accuracy of classified results. Texture indicators from Gray level co-occurrence matrices, Laws masks and multi-scale texture analysis techniques including Gabor filter banks and wavelet transformations have been derived and used to classify both *artificial* and *natural* images.

Although statistical texture indicators derived from Gray level co-occurrence matrices are very suitable for analysis of image texture, they suffer from an *edge* broadening effect as the size of the window over which the co-occurrences are measured, increases. Also the computation time in the case of statistical texture indicators increases *exponentially* with the increase of window size and the size of the image being analyzed.

Laws masks can be used to analyze and classify artificial image textures to an extent, as they are based on a specific scale and orientation set. They are useful to identify textural edges or discontinuities in spatial frequencies. However Laws masks are not very suitable when analyzing textures from natural images or remotely sensed data as natural textures are a lot more complex when compared to artificial textures.

A qualitative and quantitative comparison between statistical texture indicators and multi-scale texture indicators has been performed. Multi-scale texture indicators derived from Gabor filter banks have been found to be very effective due to the nature of

their configurability to target specific textural frequencies and orientations in an image. One of the major difficulties in using Gabor filter banks for image texture analysis is determining the dominant spatial frequency for a given texture. In the experiments, the dominant frequencies were *estimated* manually by measuring the rate of change of pixel intensities. This needs to be a subject of further research, as the determination of an algorithm to determine the dominant spatial frequency for a given image texture, will help in effective use of Gabor filter banks and help reduce the *curse of dimensionality*. The major difference between Gabor filters and wavelet transforms are that Gabor filters are based on a non orthogonal basis due to which filtered images have a lot of *overlapping information* and therefore there is no subsequent *inverse* Gabor filter.

Wavelet transformations are effective tools in image texture analysis as they help identify the ideal scale at which texture indicators need to be measured and reduce the computation time taken to derive statistical texture indicators. Haar wavelets are more effective when there are sudden changes in texture. Daubechies wavelets have been found to be more useful where textural change in gradual. Further research needs to address the effectiveness of other wavelets in image texture analysis.

Possible avenues for further research lie in integrating texture indicators as *additional channels* along with spectral information for classification, segmentation and feature extraction. Also, the use of texture indicators with segmentation algorithms like the Watershed Segmentation algorithm and the Region Growing algorithm needs to be explored. These algorithms should give a better result when compared to conventional unsupervised classification techniques. This is due to the fact that, while classification

takes place at a pixel level, segmentation takes into account the pixel and its local neighborhood, and therefore there is a *spatial* component in the segmentation result.

Multi-channel filtering techniques like Gabor filters and Wavelet transforms are sophisticated techniques that have been developed by the Computer Vision, Signal Processing and the Pattern Recognition community. A lot of work has been done to *translate* and *interpret* their algorithms and parameter settings so as to make it more understandable to the Remote Sensing community.

There is no *single method* can effectively help in the analysis and discrimination between *all* image textures. The method of analysis of image texture is based on the type of image texture being analyzed.

Finally, a robust set of software tools that are capable of aiding *further adoption* of image texture analysis by the remote sensing community has been developed using the popular .NET and ArcObjects. ArcObjects has been chosen as the API of choice, as these tools can be *seamlessly integrated* into ArcGIS.

REFERENCES

Abasolo, M.J., & Perales, F.J. (2003). Wavelet analysis for a new multiresolution model for large-scale textured terrains. *Proceedings of Computer Graphics, Visualization and Computer Vision (WSCG), 3-7 February 2003, Plzen, Czech Republic.*

Angelo, N.P., & Haertel, V. (2003). On the application of Gabor filtering in supervised image classification. *International Journal of Remote Sensing, 24*, 2167-2189.

Arivazhagan, S., & Ganesan, L. (2003). Texture classification using wavelet transform. *Pattern Recognition Letters, 24*, 1513-1521.

Arivazhagan, S., Ganesan, L., & Priyal, S.P. (2006). Texture classification using Gabor wavelets based rotation invariant features. *Pattern Recognition Letters, 27*, 1976-1982.

Baraldi, A., Parmiggiani, F., & Imga-Cnr, M. (1995). An investigation of the textural characteristics associated with gray level co-occurrence matrix statistical parameters. *IEEE Transactions on Geoscience and Remote Sensing, 33*, 293-304.

Bartels, M., Wei, H., & Mason, D.C. (2005). Wavelet packets and co-occurrence matrices for texture-based image segmentation. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, 428-433.

Bovik, A.C., Clark, M., & Geisler, W.S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*, 55-73.

Chen, Y.Q., Nixon, M.S., & Thomas, D.W. (1995). Statistical geometrical features for texture classification. *Pattern Recognition, 28*, 537-552.

Chica-Olmo, M., & Abarca-Hernández, F. (2000). Computing geostatistical image texture for remotely sensed data classification. *Computers and Geosciences, 26*, 373-383.

Choi, H., & Baraniuk, R.G. (2001). Multiscale image segmentation using wavelet-domain hidden Markov models. *IEEE Transactions on Image Processing, 10*, 1309-1321.

Clark, M., Bovik, A.C., & Geisler, W.S. (1987). Texture segmentation using Gabor modulation/demodulation. *Pattern Recognition Letters, 6*, 261-267.

Clausi, D.A., & Ed Jernigan, M. (2000). Designing Gabor filters for optimal texture separability. *Pattern Recognition, 33*, 1835-1849.

Daugman, J.G. (1988). Complete discrete 2-D Gabor transforms by neural networks for imageanalysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing]*, *36*, 1169-1179.

Duin, A., Mao, J., & Jain, A.K. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 22*, 4–37.

Dunn, D., & Higgins, W.E. (1995). Optimal Gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, *4*, 947-964.

Dunn, D., Higgins, W.E., & Wakeley, J. (1994). Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*, 130-149.

Farrokhnia, F., & Jain, A.K. (1991). A multi-channel filtering approach to texture segmentation. *Proceedings of Computer Vision and Pattern Recognition, 1991. CVPR'91., IEEE Computer Society Conference on*, 364-370, *3-6 June 1991, Maui, HI, USA.*

Haralick, R.M., Dinstein, I., & Shanmugam, K. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics, 3*, 610-621

He, D.C., & Wang, L. (1991). Texture features based on texture spectrum. *Pattern Recognition, 24*, 391-399.

Idrissa, M., & Acheroy, M. (2002). Texture classification using Gabor filters. *Pattern Recognition Letters, 23*, 1095-1102.

Jain, A.K., & Farrokhnia, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern Recognition, 24*, 1167-1186.

Jain, A.K., Ratha, N.K., & Lakshmanan, S. (1997). Object detection using Gabor filters. *Pattern Recognition, 30*, 295-309.

Kandaswamy, U., Adjeroh, D.A., & Lee, M.C. (2005). Efficient texture analysis of SAR imagery. *IEEE Trans. on Geoscience and Remote Sensing, 43*.

Kumar, P., & Foufoula-Georgiou, E. (1997). Wavelet analysis for geophysical applications. *Rev. Geophys, 35*, 385–412.

Kyrki, V., Kamarainen, J.K., & Kälviäinen, H. (2004). Simple Gabor feature space for invariant object recognition. *Pattern Recognition Letters, 25,* 311-318.

Lam, W.K., & Li, C.K. (1995). Classification of rotated and scaled textures by local linear operators. *IEEE International Symposium on Circuits and Systems, 1995, ISCAS '95, 241,* 243-246.

Livens, S., Scheunders, P., van de Wouwer, G., & Van Dyck, D. (1997). Wavelets for texture analysis, an overview. *Sixth International Conference on Image Processing and Its Applications, 1997, 2, Dublin, Ireland.*

Lu, C.S., Chung, P.C., & Chen, C.F. (1997). Unsupervised texture segmentation via wavelet transform. *Pattern Recognition, 30,* 729-742.

Myint, S.W. (2001). A robust texture analysis and classification approach for urban land-use and land-cover feature discrimination. *Geocarto International, 16,* 29 – 40.

Randen, T., & Husoy, J.H. (1994). Multichannel filtering for image texture segmentation. *Optical Engineering, 33,* 2617-2625.

Randen, T., & Husoy, J.H. (1999). Filtering for texture classification: a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*, 291-310.

Randen, T., Husoy, J.H., & Stavanger, H. (1995). Optimal texture filtering.. *Proceedings of International Conference on Image Processing, 1995, 1, 23-26 October 1995, Washington DC, MD, USA.*

Reed, T.R., & du Buf, J.M.H. (1993). A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding, 57*, 359-372.

Tuceryan, M., & Jain, A.K. (1993). Texture analysis. *Handbook of Pattern Recognition and Computer Vision*, 235–276.

Turner, M.R. (1986). Texture discrimination by Gabor functions. *Biological Cybernetics, 55*, 71-82.

Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing, 4*, 1549-1560.

Van de Wouwer, G., Scheunders, P., & Van Dyck, D. (1999). Statistical texture characterization from discrete wavelet representations. *IEEE Transactions on Image Processing, 8*, 592-598.

Walker, J.S. (1999). *Wavelets and Their Scientific Applications* New York: CRC Press.

Wechsler, H. (1980). Texture analysis - A survey. *Signal Processing, 2*, 271-282.

Wei, H., & Bartels, M. (2006). Unsupervised Segmentation Using Gabor Wavelets and Statistical Features in LIDAR Data Analysis. *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01*, 667-670.

Weldon, T., & Higgins, W.E. (1999). Designing multiple Gabor filters for multitexture image segmentation. *Optical Engineering, 38*, 1478.

VITA

Name: Rahul Ravikumar

Address: c/o Department of Geography, Eller O&M Building,
College Station, TX 77843-3148

Email Address: rahul@geog.tamu.edu

Education: B.E., Geoinformatics, College of Engineering, Guindy,
Anna University – India, 2005
B.Tech., Information Technology, College of Engineering, Guindy,
Anna University – India, 2006
M.S., Geography, Texas A&M University, 2008