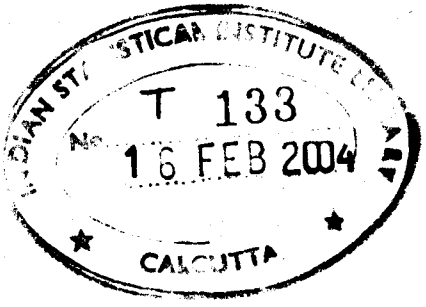
ON TEXTURED IMAGE ANALYSIS USING WAVELETS

Mausumi Acharyya
Machine Intelligence Unit
Indian Statistical Institute
203 B. T. Road
Kolkata - 700108
India.



A thesis submitted to the *Indian Statistical Institute*
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
2002

To my Parents to whom I owe my existence

ACKNOWLEDGEMENTS

I would like to extend my most sincere thanks to my advisor, Professor Malay K. Kundu for his guidance, encouragement and all the things that he had done in helping me develop my professional and personal skills. I am certain that I will benefit from his rigorous scientific approach, and the way of critical thinking throughout my future career. I want to thank the PhD council members of the Machine Intelligence Unit: Professor Sankar K. Pal, Professor C. A. Murthy, and Dr. Susmita Mitra for their constructive suggestions and comments. Also, my thanks go to Dr. Sambhunath Biswas, Dr. Ashish Ghosh, Dr. Debaprasad Mandal and Mr. B. Uma Sankar. I thank Dr. Sanghamitra Bandyopadhyay for her valuable suggestions and encouragement. I am also grateful to Dr. Rajat K. De for his all round help and support with conceptual problems at all times. Some of my works have been jointly carried out with him. He has been in close association with me in preparing this manuscript and has been an excellent friend all through my research period. I am eternally indebted to my colleague and junior Mr. Pabitra Mitra for all the help and expertise provided by him throughout this study. I really appreciate all the valuable time he has spent in helping me with any problem in programming or handling the software. I have been very fortunate to have him as my junior and it would take more than a few words to express my sincere gratitude. Many thanks to B. L. Narayan and Sarif K. Naik for the excellent companionship provided by them. Thanks to Dr. Swati Chaudhuri, Ms. Minakshi Banerjee and Ms. Tandra Pal for their care and friendship. Thanks are also due to all the office staffs of MIU.

Most of all, my deepest thanks go to my family. I can not thank them enough for their love, support, sacrifice and their belief in me. I would like to show my deep respect and sincere gratitude towards my parents who gave me the most valuable assets in life: industry, perseverance and tenacity. I shall forever remain indebted to them for being there at all phases of my life and for their encouragement in my research. It is beyond my capacity to express the amount of support and inspiration I have received from them. A deep sense of gratitude goes to my brothers Ashish and Tapas for their moral support, encouragement and inspiration and their never ending trust on me. I take this opportunity to thank my sisters-in-law Deepabali and Snigdha. Thanks to Deepabali for her encouragement, friendship, humor and company. Thanks to Snigdha for being my best friend and a wonderful companion and for reminding me that life

has more to it than textures. Thank you all!

I would also like to thank the Indian Statistical Institute, Kolkata, India and the Council for Scientific and Industrial Research, New Delhi, India for their generous financial support.

Mausumi Acharyya.

(Mausumi Acharyya)
ISI, Kolkata
October 2002.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Scope of the Thesis

1.1 Introduction

The most beautiful thing we can experience is the mysterious. It is the source of all true art and science. He to whom this emotion is a stranger, who can no longer pause to wonder and stand rapt in awe, is as good as dead: his eyes are closed.

(Albert Einstein)

In image processing and computer vision research, we aim to derive better tools that give us different perspectives on the same image, allowing us to understand not only its content, but also its meaning and significance. Image processing can not compete with the human eye in terms of accuracy but it can outperform the latter easily on observational consistency, and ability to carry out detailed mathematical estimations. With time, image processing research has broadened from the basic pixel-based low-level operations to high-level analysis, that now includes the use of artificially intelligent techniques for image interpretation and understanding. These new technologies are being developed to gain a better semantic understanding of an image based on the relationship between its components and its context.

What comprises an image must be first identified before we can analyze the image any further. Texture is a concept used to indicate some spatial properties of image regions. Thus, for an example, we can think of the above adage, where we can identify the printed text portion from the handwritten portion (signature) as two different textures.

In the present thesis, we are primarily interested in texture analysis and application of texture analysis to real life data, like document and remotely sensed images and some natural scenes. The recognition of textures in an image is not at all a trivial task. So, this area has been addressed by a number of researchers. Textures have considerable variability in terms of translation, rotation, gray-scale transform etc., and presence of noise, which make their identification difficult.

Almost all naturally occurring patterns and natural surfaces exhibit texture. Texture is a fundamental characteristic of an image, and plays an important cue to the human visual system for recognition and interpretation of images. Despite its pivotal role in

the analysis of image data, there exists neither a formal/precise definition of texture nor an obvious qualitative measure to characterize it. Image texture can be quantitatively expressed in terms of *coarseness, fineness, granularity, lineation, randomness* and *smoothness.* The analysis of image texture is extremely important. It requires the understanding of how humans discriminate between different textures and how to model our algorithms to do a predefined task in the best way. Undoubtedly, texture analysis has a wide range of applications. These include:

- classifying images based on texture;
- segmenting an input image into regions of homogeneous texture;
- determining surface-shape on the basis of texture gradient;
- synthesizing natural looking textures for graphics applications; and
- image retrieval from a database based on texture similarity.

Most standard definitions of texture treat it as a measure of coarseness in an image. A simple measure of texture coarseness is based on, first computing the local extremum of an image function along rows and columns. The density of this extremum can then be used as a measure of coarseness of texture. The coarseness at a pixel location can be determined by doing this computation within a small neighborhood.

The basis of a texture description is usually based on some mathematical model. Many different models for modeling texture using different mathematical tools have been proposed. These include: *co-occurrence matrices, fractals, Markov random fields (MRFs), Gabor filters, neural networks, wavelets* etc. Furthermore, each model usually comes with its own set of scalar measures, which try to describe certain texture features, like, *smoothness, coarseness* or *regularity* of the texture.

In general, texture analysis task can be divided into two major phases. The first phase is a critical one and involves feature extraction, in which the image information is reduced to a small set of descriptive features. The second step can be regarded as a standard pattern recognition procedure, and deals with classifying the obtained features.

Texture classification aims at assigning texture labels to unknown textures. This can be accomplished by having an *a priori* knowledge of the classes to be recognized. As an

example, texture classification has been appropriately applied for the classification of regions in satellite images into categories of *land use* [80]. *Segmentation* is a process of partitioning an image space into some non-overlapping meaningful homogeneous (property) regions. The quality of segmentation depends on the term 'meaningful'; which means how we define the homogeneity of a region. Segmenting an image into meaningful regions based on textural cues is referred to as *texture segmentation.* Application areas of texture segmentation are discussed in the later part of this chapter.

An important aspect of texture is scale. The importance of scale in texture descriptions is clear from the change in appearance of most textures when viewed at different resolutions. A texture cannot be analyzed without mentioning the frame of reference in which the gray level primitives are considered, *e.g.*, for an image texture there exists a scale at which if it is examined it would appear smooth. As the resolution increases, the image may exhibit a fine texture. *Texture can be defined as a local statistical distribution of pixel pattern (micro-region) in an observer's domain of interest.* Psychovisual studies reveal that the human visual system processes an image in multiple scales [30]. The visual cortex has separate cells that decompose an image into various frequency bands and orientations. Texture is especially suited for this type of analysis because of its properties. Multiresolution techniques give image representations, in which both frequency and spatial information are present. During the past two decades, wavelet analysis has become a paradigm for multiresolution analysis, and has found important applications in various fields, ranging from seismology to image compression. So, multiscale wavelet analysis is an ideal approach to describe texture, because it is well recognized that *scale* is one of the most important aspects of texture information.

A wavelet system is a set of building blocks to construct or represent a signal or function. It is a two-dimensional expansion set for some class of one (or higher) dimensional signals. The wavelet expansion gives a time-frequency[1] localization of a signal. That is, most of the energy of a signal is well represented by a few expansion coefficients. Wavelet system also satisfies the multiresolution conditions. This means that, the lower resolution representation of a signal can be obtained from its higher resolution information.

Typically, the wavelet transform maps a signal onto a low resolution signal and a

[1]Note that in case of a two-dimensional signal (or image), one talks of the *spatial domain* and the *spatial-frequency.*

series of detail (higher resolution) signals. The low resolution signal is obtained by iteratively blurring the original signal; the detail signals contain information lost during this operation. The energies or the mean deviations of the detail signals are most commonly used features for texture analysis.

The present thesis demonstrates the application of this multiresolution aspect of wavelet representation, and the development of several algorithms for feature extraction of digital images with textural content. We will demonstrate from the texture characterization perspective how the proposed decomposition schemes perform more efficient energy compaction of an image, and the first-order statistics computed on the decomposed image can be used as reliable texture descriptors for classification and segmentation. In addition to this, a neuro-fuzzy algorithm is employed to evaluate the extracted features, so as to reduce the feature dimensionality and increase fidelity. Finally, the developed texture segmentation algorithms have successfully been applied to real life data like document and remotely sensed images and some natural scenes.

The chapter is organized as follows. Section 1.2 gives a formal definition of texture, discusses some applications of texture analysis, and reviews the various methods for texture analysis. Section 1.3.1 deals with the basics of wavelet, and furnishes a survey of wavelet-based texture analysis methods. An overview of neuro-fuzzy methodologies for feature selection is discussed in Section 1.4.1 and also presents the basic concept, and features of classification/clustering). Finally, Section 1.5 summarizes the scope of the thesis.

1.2 Texture Analysis

Texture is an important cue for the analysis of many types of images. It has always posed a challenge to the image analysis and computer vision community. The ability to effectively classify and segment images based on textural features is therefore an important research topic. Before we go into the details of several methodologies for texture analysis, we try to broadly define what texture means in the following section.

1.2.1 Definition of texture

Texture refers to properties that represent the surface or structure of an object (in reflective and transmissive images, respectively); it is widely used, and perhaps intuitively obvious, but has no precise definition due to its wide variability. It may be defined as *something consisting of mutually related elements*; which means that a group of pixels (a **texture primitive** or **texture element**) are considered. According to such a definition of texture, it is highly dependent on the number of these texture elements (texture scale) [78]. Examples of textures can be found in [27].

Texture is usually described as **fine**, **coarse**, **grained**, **smooth**, etc., implying that some more precise features must be defined to make machine recognition possible. Such features can be found in the **tone** and **structure** of a texture [78]. Tone is based mostly on pixel intensity properties in the primitive, while structure is the spatial relationship between primitives.

Each pixel can be characterized by its location and tonal properties. A texture primitive is a contiguous set of pixels with some tonal and/or regional property, and can be described by its average intensity, maximum or minimum intensity, size, shape, etc. The spatial relationship of primitives can be random, or they may be pair-wise dependent, or some number of primitives can be mutually dependent. Image texture is then described by the number and types of primitives and by their spatial relationship.

The same number and same type of primitives does not necessarily give the same texture. Similarly, the same spatial relationship of primitives does not guarantee texture uniqueness, and therefore is not sufficient for texture description. Texture tone and structure are not independent, textures always display both tone and structure even though one or the other usually dominates. Tone can be understood as tonal properties of primitives, taking primitive spatial relationships into consideration. Structure refers to spatial relationships of primitives considering their tonal properties as well.

If the texture primitives in the image are small and if the tonal differences between neighboring primitives are large, a **fine** texture results. If the texture primitives are larger and consist of several pixels, a **coarse** texture results. Again, this is a reason for using both tonal and structural properties in texture description. Note that the **fine**/coarse texture characteristic depends on *scale*.

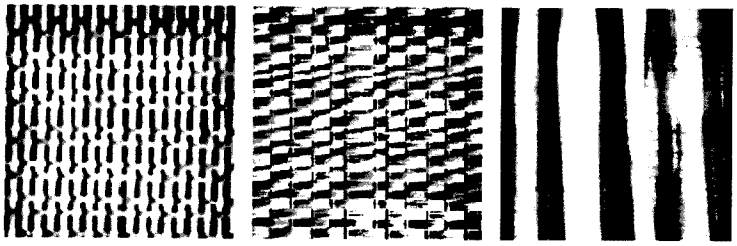


Figure 1.1: Examples of structured textures

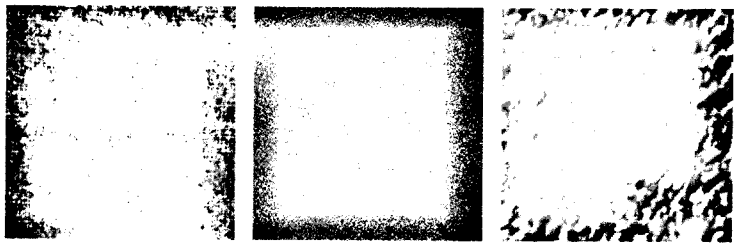


Figure 1.2: Examples of stochastic textures

Further, texture can be classified according to their strength. **Weak** textures have small spatial interactions between primitives, and can be adequately described by frequencies of primitive types appearing in some neighborhood. In **strong** textures, the spatial interactions between primitives are somewhat regular.

Based on the above definitions of texture, it can be most popularly classified into two categories, namely, stochastic and structured texture [78, 71].

The **Structured** texture is decomposable into two dimensions. The first dimension describes the primitives of the texture and the second dimension accounts for the description of the spatial dependence or interaction between primitives (placement rules). A given texture image may have several types of primitives. These primitives may vary from one position to another within the texture image. Moreover, the placement rule may be deterministic or stochastic. The structured textures are sometimes also called as macro-textures. A few samples of structured texture are shown in Fig. 1.1.

The **Stochastic** textures are those that can not be decomposed into two dimensions as in the case of structured textures. They form an irregular but visually homogeneous structure. Sometimes the term micro-texture is also used to represent a stochastic texture. Few examples of stochastic texture are shown in Fig. 1.2.

Rosenfeld and Lipkin [140] have stated that texture can be studied on at least two

levels, statistical and structural. On the statistical level, the texture of an image is defined by a set of statistics extracted from the entire textural region. On the structural level, a texture is considered to be composed of sub-patterns, called *primitives*. A variety of statistical methods such as auto-correlation [78], co-occurrence approach [80], edge frequency method [78], Law's method [101], etc, have been proposed for texture analysis which are based on the variability in gray-scale in images, which indicates the inherent coarseness or fineness of texture. The textural property of an image also depends on the spatial size of texture primitives. Large primitives give rise to coarse texture (rock surface) and small primitive gives fine texture (silk surface).

1.2.2 Applications of texture analysis

There exists a wide range of successful applications of texture recognition systems. Some of the examples are briefly discussed in this section:

Remote sensing: Among the different remote sensing techniques, satellite imagery plays an important role so far as different practical applications are concerned. Numerous approaches of texture analysis in remote sensing images have been reported in the literature. Application areas like, terrain classification, cloud classification, sea floor mapping, seismic pattern recognition [54, 112, 148] etc. have been studied. In this thesis we have studied such an application and further details are furnished in Chapter 7 (Section 7.2).

Document segmentation: When viewed from a distance, the text regions and graphics portions of a printed document are perceived as different textures. For a particular language, font, etc., the text appears as a homogeneous texture. By the use of texture analysis, it is possible to segment regions of text from regions of graphics in scanned documents. Such segmentation is important for successful document analysis by computer [34, 102, 154, 104]. The present thesis deals with such an application and is described in Chapter 7 (Section 7.1).

Content based image search: Different types of digital image data in volumes are being used for archiving and transmission over the internet. Locating some particular image with specific content, in a database of images is not easy. Therefore, designing of search tool for content-based image retrieval requires automatic labeling of an image

based on its contents (*e.g., number of different regions, shape, texture color* etc.), and is a difficult task. Texture analysis plays an important role in this process [62].

Medical imaging: Medical images are in general captured using non-intrusive techniques, such as photography, x-ray, ultrasound, tomography, etc. But also there are certain types of images, which are acquired by, intrusive techniques *e.g.*, microscopic images of biopsies. In these images, textural properties are important for diagnosis, *e.g.*, cancer is often detected by texture characteristics of images obtained by different medical imaging techniques. Texture analysis techniques have been applied successfully for the early detection of lesions in mammogram image [75, 155] and also in electron microscopic image of thin slices of liver cells for cancer detection [16].

Shape from texture: Texture analysis has also found its application in reconstructing three-dimensional shape from two-dimensional images. By analyzing how the texture pattern changes, the shape of an object may in principle be constructed from the texture [172].

Industrial inspection: In industrial processes, the detection of defects in manufactured products or in the raw material may be of crucial importance. In many cases, the quality of a surface is well characterized by its texture. Hence, texture analysis plays an important role in inspection. Numerous approaches to inspection by texture techniques have been proposed. Examples of such inspection are quality inspection of painted metallic surfaces [60], lumber defect detection [133] etc.

1.2.3 Different approaches to texture analysis

A wide variety of approaches for texture feature extraction and analysis exists. Making an extensive survey of different approaches is a giant work. Several books and articles that provide overviews of the existing methodologies [78, 134] are available. In this section we will describe briefly four major approaches to texture analysis including the signal processing approach which motivates the research work in this thesis.

Geometric and structural approaches

In structural models of texture, it is defined as a *structure, which is made up of a large ensemble of elements that resemble each other to a great extent and which have some kind of "order" in their locations, with no one element attracting the viewer's eye* [64].

These elements are called texture *primitives*. Hence, the texture is characterized by a description of its primitives and their placement according to some placement rules. The placement rules may be deterministic [111] or stochastic [78] for more random structure. The former implies that either the texture image is highly periodic or the reconstructed image will look synthetic. Matsuyama *et al.* [111] for example, have used the Fourier domain as the basis of a top-down analysis. They have first identified the two-dimensional (2D) fundamental spatial frequency of the texture by locating the maxima of the spectral density function. This information have been used to measure the texture placement rule and to isolate the texture primitives. This method fails if the texture does not have a complete periodic structure.

Kundu and Chaudhuri [96] have proposed the use of fuzzy geometric features for texture classification. In their approach, first a set of 2D local membership value extremum is detected for the image. These are used as seed regions and grown till they do not touch other seed regions. A number of features from these regions are calculated, such as, fuzzy area, perimeter, compactness, height, and width, which form the basis of texture classification. Chen *et al.* [40] and Garcia *et al.* [70] have investigated the use of binary stacks for characterizing texture. Chou [43] has given a simple technique for classifying image pixels in one of the three categories: *shaped feature point, smooth feature point*, and *textured feature point*. The decomposition is based on a set of simple fuzzy rules operating on the edge strength information.

The main strength of these structural models is their conformity to the physical reality of natural textures. But the major drawback is that, for a given texture, we do not know *a priori* the most appropriate primitive's description. Obviously, these models work well only when we have some, *a priori* information on the type of textures being analyzed. The other drawback of these models is that they work only with periodic and quasi-periodic textures.

Statistical approaches

Statistical models are those, for which there is no explicit assumption on the geometric structure of the texture. These models are mainly designed for stochastic textures but it can also analyze structured textures with some success.

Haralick *et al.* [80] have introduced the concept of co-occurrence matrices for classifying texture images. Co-occurrence matrix gives information on how different combination

These elements are called texture *primitives.* Hence, the texture is characterized by a description of its primitives and their placement according to some placement rules. The placement rules may be deterministic [111] or stochastic [78] for more random structure. The former implies that either the texture image is highly periodic or the reconstructed image will look synthetic. Matsuyama *et al.* [111] for example, have used the Fourier domain as the basis of a top-down analysis. They have first identified the two-dimensional (2D) fundamental spatial frequency of the texture by locating the maxima of the spectral density function. This information have been used to measure the texture placement rule and to isolate the texture primitives. This method fails if the texture does not have a complete periodic structure.

Kundu and Chaudhuri [96] have proposed the use of fuzzy geometric features for texture classification. In their approach, first a set of 2D local membership value extremum is detected for the image. These are used as seed regions and grown till they do not touch other seed regions. A number of features from these regions are calculated, such as, fuzzy area, perimeter, compactness, height, and width, which form the basis of texture classification. Chen *et al.* [40] and Garcia *et al.* [70] have investigated the use of binary stacks for characterizing texture. Chou [43] has given a simple technique for classifying image pixels in one of the three categories: *shaped feature point, smooth feature point*, and *textured feature point.* The decomposition is based on a set of simple fuzzy rules operating on the edge strength information.

The main strength of these structural models is their conformity to the physical reality of natural textures. But the major drawback is that, for a given texture, we do not know *a priori* the most appropriate primitive's description. Obviously, these models work well only when we have some, *a priori* information on the type of textures being analyzed. The other drawback of these models is that they work only with periodic and quasi-periodic textures.

Statistical approaches

Statistical models are those, for which there is no explicit assumption on the geometric structure of the texture. These models are mainly designed for stochastic textures but it can also analyze structured textures with some success.

Haralick *et al.* [80] have introduced the concept of co-occurrence matrices for classifying texture images. Co-occurrence matrix gives information on how different combination

of pixel gray levels occur in an image. Gray level co-occurrence matrices are referred to as GLCM, it is a second-order statistical measurement because it measures the relationship between two pixels. Extraction of a set of 14 textural features, from each of these matrices, is suggested. Some of these textural feature measures obtained from the matrices relate to specific textural characteristics of the image such as homogeneity, contrast and the presence of organized structure within the image. Other measures characterize the complexity of the gray-tone transitions, which occur in the image. Since this seminal study, the features suggested by Haralick *et al.* [80] have been used in most of the texture studies. Dyer *et al.* [57] have used a gray level co-occurrence matrix based on edge maxima for texture classification.

In 1981, Julesz [89] wrote a seminal paper on textons. He proposed the theory of textons to explain the pre-attentive discrimination of texture pairs. Textons are visual events (such as *collinearity, terminations, closure*, etc.), whose presence is detected and used in texture discrimination. The textons are invariant to positional and scaling transformations. These symmetries of textons can not be broken by pre-attentive and peripheral vision. The analysis of textons is now a well developed area in texture analysis and several researchers have used them for classifying and modeling texture. Gagalowicz [69] has investigated the models for texture fields for modeling human vision. The author has designed a new tool to synthesize stochastic texture fields and has proposed a new conjecture for the visual discrimination of texture fields. The idea of second-order statistics, proposed by Julesz, has been replaced by the ideas of second-order spatial averages and the concept of locality. Some recent studies on statistical approaches include works by Bonet and Viola [24] and Mandhuchi and Portilla [110].

Statistics of order greater than two have also been applied for texture analysis. Murino *et al.* [116] have proposed a novel texture classifications scheme which is based on using higher order statistics (HOS) for defining discriminatory features. One of the advantages of using these parameters is that, they are insensitive to additive Gaussian noise, especially third-order statistical parameters. Al-Janobi [15] has proposed a cross-diagonal texture matrix (CDTM) technique for texture analysis. This method incorporates the properties of gray level co-occurrence matrices and texture spectrum.

Model based approaches

Model based texture analysis methods are based on the construction of an image model

that can be used to describe texture. The model parameters capture the essential perceived qualities of texture. Ahuja and Rosenfeld [14] have proposed a random mosaic model for modeling natural textures. Markov random fields (MRFs) have been popular for modeling images. They are able to capture the local (spatial) contextual information in an image. These models assume that the intensity at each pixel in the image depends on the intensities of only the neighboring pixels. MRF models have been applied for texture classification tasks by Chellapa [35] and Khotanzad and Kashyap [92].

Many natural surfaces have statistical qualities of roughness and self-similarity at different scales. Fractals are very useful in modeling these properties in texture images. Kaplan [90] has proposed an extended fractal analysis approach to texture classification and segmentation. The author has shown that, features obtained from methods such as Gabor transform provide a compact description of harmonics in a texture using local linear transforms. These techniques succeed in classifying a variety of textures but fails to distinguish some other types of natural textures that do not show any periodic nature.

Signal processing approaches

This section reviews the various techniques of texture analysis that relies on signal processing methods. Most of these methods try to compute some features from the filtered images, which are then either used in classification or segmentation tasks. The feature detection stage in many of these approaches is based on Julesz's findings that the human visual system uses only local changes in texton density and therefore the first-order local statistics are the best texture features [89].

Spatial domain filtering is the most direct way to capture image texture properties. Earlier attempts using such filtering methods was mainly based on the edge density measurement. Fine textures tend to have a higher density of edges per unit area than coarser ones. Simple edge operators (such as the Robert's or the Laplacian operator) usually computes the measurement of edginess. Malik and Perona [106] have used spatial filtering to model the pre-attentive texture perception in human visual system. They have worked with even-symmetric filters that consist of differences of offset Gaussian (DOOG) functions. This method is able to discriminate various types of textures that are natural as well as synthetic.

The analysis of a textured image is best done in the frequency domain. The psychophysical results indicate that, the human visual system analyzes a textured image by decomposing it into multiple channels corresponding to its frequency and orientation components [51]. The multiple channels tuned to different frequencies are also referred as multichannel filtering. Many signal processing methods use a multichannel filtering scheme. In present days, these approaches are becoming more important and inexpensive hardware solutions are also readily available. Daugman [51] has used Gabor filters [68] in the modeling of the receptive fields of simple cells in the visual cortex of humans. Gabor functions are the Gaussian modulated complex exponential in the spatial domain and shifted Gaussian functions in the frequency domain and thus have the optimum resolution of both frequency and time. The proposal to use the Gabor filters in texture analysis was made by Turner [163] and Clark and Bovik [44]. The Gabor-based decomposition, has been extensively used for texture discrimination and segmentation in [60, 56, 103]. Dunn *et al.* [56] have designed the bandpass Gabor-based filters and have shown that if Gabor filters are carefully designed the segmentation results would improve. Bovik *et al.* [26] have modeled texture as irradiance patterns that are distinguished on the basis of high concentration of localized spatial frequencies.

He and Wang [82] have proposed a new set of texture measures derived from the texture spectrum of an image. The authors have suggested that, a texture image be composed of a collection of small units, known as texture units, which characterize the local texture information of a pixel and its neighborhood. *Texture spectrum is termed as the frequency distribution of all the texture units.* Based on texture spectrum, various features are extracted from a texture image. Chen and Chen [36] have studied in detail the traditional Gabor filter and have proposed a modified version that works at different image resolutions. The procedure involves convolution of the Fourier transform of the image and the Gabor function, and then taking an inverse transform. By this process, different frequency subbands are highlighted depending on the Gabor filter parameters. The resultant image can be analyzed for detecting features that describe texture properties corresponding to different frequencies.

Gabor filters have some major drawbacks in practical applications. The outputs of Gabor filter banks are not mutually orthogonal, which demand higher computational complexities and may result in significant correlation between texture features. In

addition, the selection of filters is dependent on the image frequency characteristics. The accurate implementation of a complete Gabor expansion would necessitate an impractical number of filters. Also the evaluation of low frequency components are difficult.

Most of these problems can be avoided by using wavelet transform. Wavelet theory provides a more formal, precise and unified framework for the analysis and characterization of a signal at different scales (resolutions). The present thesis focuses on the *signal processing approaches* to texture analysis, using wavelets.

1.3 Wavelet-based Texture Analysis

This section furnishes a brief exposure on some of the basic concepts of wavelet analysis (detailed description of it would be presented in the next chapter). It also discusses about the relevance of wavelets for texture analysis and its advantages over Gabor functions. This is followed by a survey on wavelet-based texture analysis methods.

1.3.1 Fundamentals of Wavelets

The wavelet transform decomposes a signal into its "wavelets", which are the scaled and shifted versions of the "mother wavelet". Fig. 1.3 shows Daubechies-8 wavelet, which is a so-called wavelet from the Daubechies family of wavelet functions. The wavelet in general is irregular in shape and compactly supported. Their irregular shape lends them to analyzing signals with discontinuities or sharp changes, while their compactly supported nature enables temporal (spatial) localization of a signal's features.

The wavelet transform allows localization in both the time (space) domain via translations of the mother wavelet, and in the scale/frequency (spatial-frequency) domain via dilations [49]. The translation and dilation operations applied to the mother wavelet are performed to calculate the wavelet coefficients, which represent the correlation between the wavelet and a localized section of the signal. The wavelet coefficients are calculated for each wavelet segment, giving a time-scale function relating the wavelets correlation to the signal. This process of translation and dilation of the mother wavelet is illustrated in Fig. 1.4. A fully scalable modulated window (mother wavelet, $\phi(t)$) is

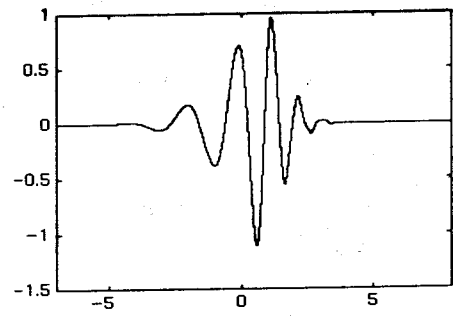


Figure 1.3: The Daubechies-8 wavelet

shifted along the signal and for every position the spectrum is calculated (Fig. 1.4). Then this process is repeated many times with a slightly shorter (or longer) window for every new cycle. At the end, the result is a collection of time-frequency representations of the signal, all with different resolutions. Because of this collection of representations, we can speak of a multiresolution analysis. Large scale (low resolution) gives an overall information, while the small scales (higher resolutions) show the details of a signal. Thus, in this context, going from small scale to large scale is equivalent to zooming out a signal. Therefore, a multiresolution representation provides a simple hierarchical framework for interpreting the signal information.

1.3.2 Relevance of wavelets

Recent models for human vision suggest the existence of an internal spatial/frequency representation that processes images by decomposing them into filtered images of various frequencies and orientation at different scales and is capable of preserving both local and global information [89]. It has been observed that the response is similar to Gabor-like function.

The basic weakness of most of the texture analysis schemes discussed previously in Section 1.2.3 is that, the image is analyzed at one scale: a limitation, which can be overcome by employing multiscale representations using Gabor filters. Although Gabor filters are easy to design and have desirable properties including orientation selectivity and filter bandwidth, but they are computationally inefficient. In addition, a large number of channels are required to approximately cover the complete frequency plane.

In the last decade, wavelet theory has emerged, as a powerful tool and the contribu-

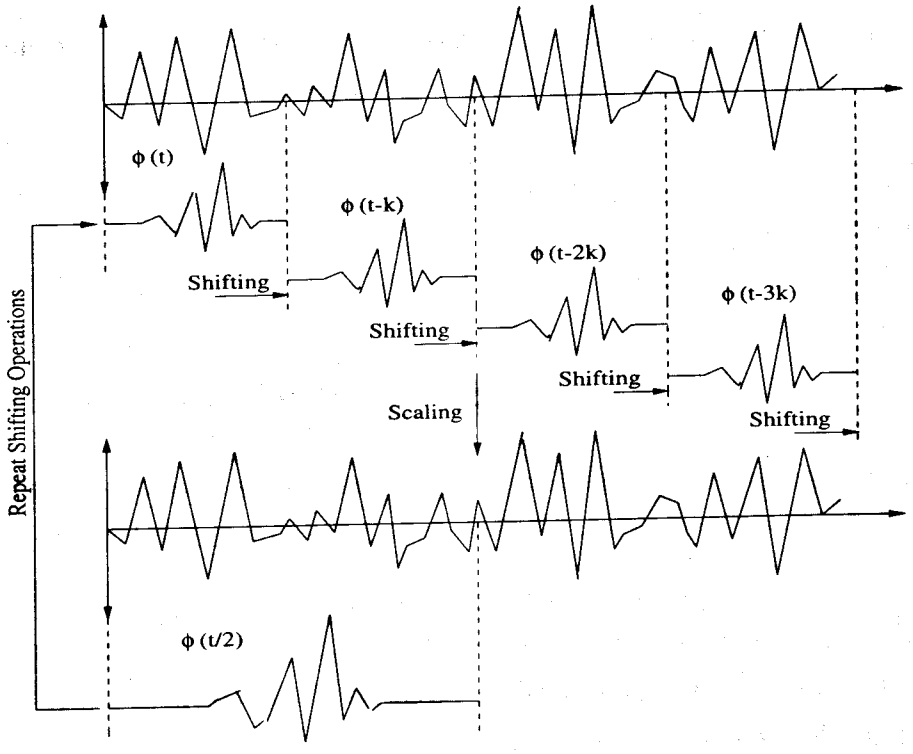


Figure 1.4: The scaling and shifting process of the WT

tions by several researchers [50, 107] have provided a mathematical framework which gives a more unified and formal paradigm for multiscale image analysis. It has provided a promising alternative through multichannel filter banks that have several useful advantages:

- Wavelet filters cover exactly the complete frequency domain.
- The wavelet expansion allows a more accurate local description and separation of signal characteristics. A wavelet expansion coefficient represents a component that is itself local and is easier to interpret. The wavelet expansion may allow a separation of components of a signal that overlap in both time (space) and frequency (spatial-frequency).
- Wavelets are adjustable and adaptable. Because there is not just one wavelet, they can be designed to fit individual applications. They are ideal for adaptive systems that adjust themselves to suit the signal.
- Fast algorithms are readily available to facilitate computation. The defining

equation of a wavelet does not use calculus *i.e.*, there are no derivatives or integrals, only algebraic operations like multiplication and addition are used that are basic to a digital computer and thus facilitates computing.

The multiscale processing which humans obviously apply successfully to texture perception, is a strong motivation for texture analysis methods based on wavelets.

1.3.3 Survey of wavelet-based texture analysis

Since this thesis pertains to some texture classification and segmentation tasks using wavelets, we present some of the most relevant and past research works done in this area.

Studies on successful application of wavelet theory for texture analysis have been reported using the multiresolution signal decomposition developed by Mallat [107]. He used quadrature mirror filters to relate information at different scales of decomposition of the embedded subspace representation. Recent advances on wavelet have motivated several important studies on texture analysis [135, 22]. M. Unser [165] has introduced local linear transforms for classification and segmentation of textures using orthogonal and compactly supported wavelets. Unser [164] has also used overcomplete wavelet decomposition of the standard wavelet and characterized the texture by a set of channel variances estimated at the output of the filter bank. Carter [31] has reported texture classification results using Mexican hat wavelets.

The standard dyadic wavelet transforms perform a frequency splitting in octave bands. which is sufficient for many problems but sometimes is too coarse for detailed texture analysis. Wavelet packet transform remedies this coarseness to a certain extent, at the expense of a large feature set and worse spatial localization. Chang and Kuo [32] have shown that texture features are most prevalent in intermediate frequency bands. This proposal has been studied with a particular attention to the use of wavelet packets [98]. Laine and Fan [99] have also carried out studies on texture analysis using multichannel wavelet frames for feature extraction, where representations obtained from both standard wavelets and wavelet packets are evaluated. Fatemi *et al.* [61] have made a study on the evaluation of performance of texture segmentation algorithms based on wavelets, which agrees with the findings of Chang and Kuo [32]. Several other

attempts on texture discrimination using wavelets have been reported in the literature [73, 103, 127].

Pichler *et al.* [126] compare wavelet transforms with adaptive Gabor filtering feature extraction and report superior results using Gabor technique. However, the computational requirements are much higher for these than needed for wavelet transform, and in certain applications accuracy may be compromised for a faster algorithm. Four filtering methods of texture discrimination has been compared by Chen and Chen [37]. These methods include Fourier transform, spatial filter, Gabor filter and wavelet transform. They find that wavelet and Gabor features perform equally well but the wavelet method is computationally less intensive. These two performances are better than the two other feature extraction methods.

Randen and Husøy [132] provide a detailed comparative study of various filtering approaches to texture extraction. The basic assumption for most filtering approaches is that the energy distribution in the frequency domain defines a texture. Hence, if the frequency spectrum is decomposed into sufficient number of subbands, the spectral energy signatures for these are very different for different textures. The approaches compared in this study include: Law's masks, ring/wedge filters, dyadic Gabor filter banks, wavelet transforms, wavelet packets and frames, quadrature mirror filters (QMF), discrete cosine transform (DCT), eigen filters, optimized Gabor filters, linear predictors, and optimized finite impulse response (FIR) filters. A summary of other studies comparing texture algorithms is also provided. The experiments have been conducted on composite texture images. The authors find that no clear hierarchy of classification performances are observed. Different methods perform better on different images. The traditional Law's method and ring/wedge filters are never the winners. The poor performances are also observed for Gabor filters and DCT. However, DCT has the least computational complexity of all methods. The QMF and wavelet frame approaches are among the best for most images. Co-occurrence matrix method is also compared and found to be the worst in all experiments.

It is to be noted that, we have done comparison of the results of our investigation with the results reported in [132] to establish the effectiveness and superiority of our proposed algorithms.

1.4 Pattern Recognition

After the extraction features, these are passed on to a feature selection phase, where the dimensionality of the feature space is reduced by retaining only some characteristic features. Finally, the selected features are passed on to a classifying/clustering system that evaluates the incoming information and makes a final decision.

1.4.1 Feature Selection

Feature selection is a process of selecting a map of the form $X = f(Y)$, by which a sample $\mathbf{y}$ $(=[y_1, y_2, \ldots, y_n])$ in an n-dimensional measurement space Ω_Y is transformed into a point $\mathbf{x}$ $(=[x_1, x_2, \ldots, x_{n'}])$ in an n'-dimensional feature space Ω_X, where $n' < n$. The main objective of this task or feature evaluation [20, 53], is to retain/develop the optimum salient characteristics necessary for the recognition process and to reduce the dimensionality of the measurement space Ω_Y so that effective and easily computable algorithms can be devised for efficient classification. The problem of feature selection has two aspects – formulation of a suitable criterion to evaluate the goodness of a feature, and deciding on its retention in the optimal set. In general, those features are considered to have optimal saliency for which inter-class/intra-class distances are maximized/minimized. The criterion of a good feature is that it should be unchanging with any other possible variation within a class, while emphasizing differences that are important in discriminating between patterns of different types.

We would discuss a neuro-fuzzy methodology of feature selection that we have worked with, in Chapter 6. Before that we present a brief overview of neuro-fuzzy methodologies for feature selection in the following section.

Overview of neuro-fuzzy feature selection

Researchers all over the world [123] have been trying to combine the merits of fuzzy set theoretic and connectionist approaches under the heading *neuro-fuzzy computing* for building more intelligent decision making systems than either of the above paradigms individually. The uncertainties involved in the input description and output decision are taken care of by the concept of fuzzy sets while the neural net theory helps in generating the required (linearly non-separable) decision regions. A survey on the

existing literature reveals that, research work on feature selection in the neuro-fuzzy hybridized paradigm, have been done on a very limited scale, which we would mention in this section. Before that we mention some of the works carried out in the individual domains of fuzzy sets and artificial neural networks (ANNs).

Bezdek and Castalez [23] have shown an application of the fuzzy c-means clustering algorithm to select an optimum feature subset from the set of available features. Pal and Chakraborty [121] have used fuzziness measures (the index of fuzziness, entropy, and π-ness) of a set in selecting features without going through classification. Some of the recent attempts made for feature selection in the framework of artificial neural networks (ANN) include [151, 52, 128]. The methods based on multilayer feedforward networks include, among others, determination of saliency (usefulness) of input features. Setino *et al.* [147] have demonstrated how a three layer feedforward neural network is used to select a subset of features. Battiti [20] has investigated the application of *mutual information* criterion to evaluate and select a set of candidate features.

Depending on whether the class information of the samples is known or not, feature selection methods are classified under supervised or unsupervised modes. Some attempts have been made by Pal *et al.* [120] to integrate the above two theories for performing the task of neuro-fuzzy feature selection in a supervised mode. Pal *et al.* [122] have studied the problem of neuro-fuzzy feature selection under unsupervised mode of training. We have worked on a neuro-fuzzy feature selection algorithm, which is primarily based in this method (deatils of which would be given in Chapter 6).

1.4.2 Classification/Clustering

In the classification/clustering phase, the selected features are passed on to the classifying/clustering system that evaluates the incoming information and makes a final decision. This phase basically establishes a transformation between the features and the classes/clusters. Different forms of transformation can be a Bayesian rule of computing *a posterior* class probabilities, nearest neighbor rule, linear discriminant functions, perceptron rule, nearest prototype rule, k-means clustering algorithm etc. [55, 53, 65, 161]

Classification techniques can be categorized into two major approaches: *parametric* and *non-parametric* according to the assumption that can be made about probability

distributions of the patterns in each of the classes. Parametric classifiers exploit the assumption that the class conditional probability densities have a known parametric form [53]. The knowledge of the form can be used to derive a parametric decision rule. Once the parameters of the decision rule are estimated from the training data, the classifier can be applied to classify unknown patterns. Non-parametric methods estimate the unknown class densities or *a posteriori* probabilities at a point using an available training set. They do not require any *a priori* mathematical model for the underlying patterns. A group of non-parametric methods derive the local statistics based on distances between the point and the patterns in the training set. The most commonly used classifier within this group is the *k-nearest neighbor (k_NN)* classifier [67]. The principle of this method is based on a voting mechanism. To find the affinity of a pattern to a class, we first find the separation of this pattern from every pattern in all classes. The pattern under study belongs to the class which contains K patterns with the closest distance to it. Another group of methods that can compute the relevant functions are Artificial Neural Networks (ANN's). This group can be considered as adaptive classifiers that learn through examples in the training set.

Segmentation algorithm accept as input a set of features and put a consistent labeling for each pixel. Fundamentally this can be considered a multidimensional data-clustering problem. Having obtained the features, the main task is to integrate these features to produce segmentation. Let us assume that there are K classes, $C_1, \ldots, C_K$. present in a data set. If the features are capable of discriminating these classes then the patterns belonging to each category will form a cluster in the feature space, which is compact and isolated from, clusters corresponding to other categories. Clustering algorithms are ideal modes for forming such clusters in the feature space. Throughout the dissertation, we have used the traditional *k-means* clustering algorithm [161].

An overview of the unsupervised *k-means* clustering algorithm is given below.

$k - means\ (x[1:B, 1:S], K)$

$x[1:B, 1:S]$: array of structure containing vectors

B $\rightarrow$ Number of feature elements in a feature vector

S $\rightarrow$ Data size (number of pixels in the image)

K $\rightarrow$ Total number of classes

begin

begin (Initialization)

Select K number of vectors arbitrarily from the array $x[1:B, 1:S]$ and then each of these are assigned a class, these form the initial class centers C_k's.

end

begin

Euclidean distance between each of the S vectors and the selected K vectors are found out taking one out of S vectors at a time. A vector is assigned to the class k if it is closest to C_k. Recompute the class centers C_k by taking mean of the vectors assigned to class k. Repeat until there is no change in the class centers.

end

end

1.5 Scope of the Thesis

The objective of this thesis is to present some results of investigations, that demonstrate the advantage of the multiresolution processing ability of wavelets for texture analysis, by developing some new algorithms for texture feature extraction. The features are extracted by linear filtering and estimation of local energy of the filter response.

The basic procedure for filtering is as follows. First, the input image is subjected to filtering that allows certain frequencies to pass through and blocks the remaining frequencies. A non-linear function is applied to the filtered image which rectifies the filter response followed by smoothing. Some of the commonly used non-linearities are magnitude, squaring and logarithm. Smoothing functions include rectangular and Gaussian functions.

Initially, the commonly used 2-band wavelet is used to extract texture features, which are invariant to rotation, translation, gray-scale variations and tolerant to noise corruption. These features are then used for texture classification and segmentation. We have extended our work further, by employing M-band, $(M > 2)$ wavelet for texture feature extraction, which gives an improvement in the quality of segmentation. The superiority of these algorithms over some related ones is exhaustively demonstrated on various benchmark images as well as real life data like, document image (both structured and unstructured), remotely sensed image and natural scene. The organization

of the thesis is depicted in the flow diagram outlining the problems addressed and is shown in Fig. 1.5. The results of the investigation are summarized below under different chapter headings.

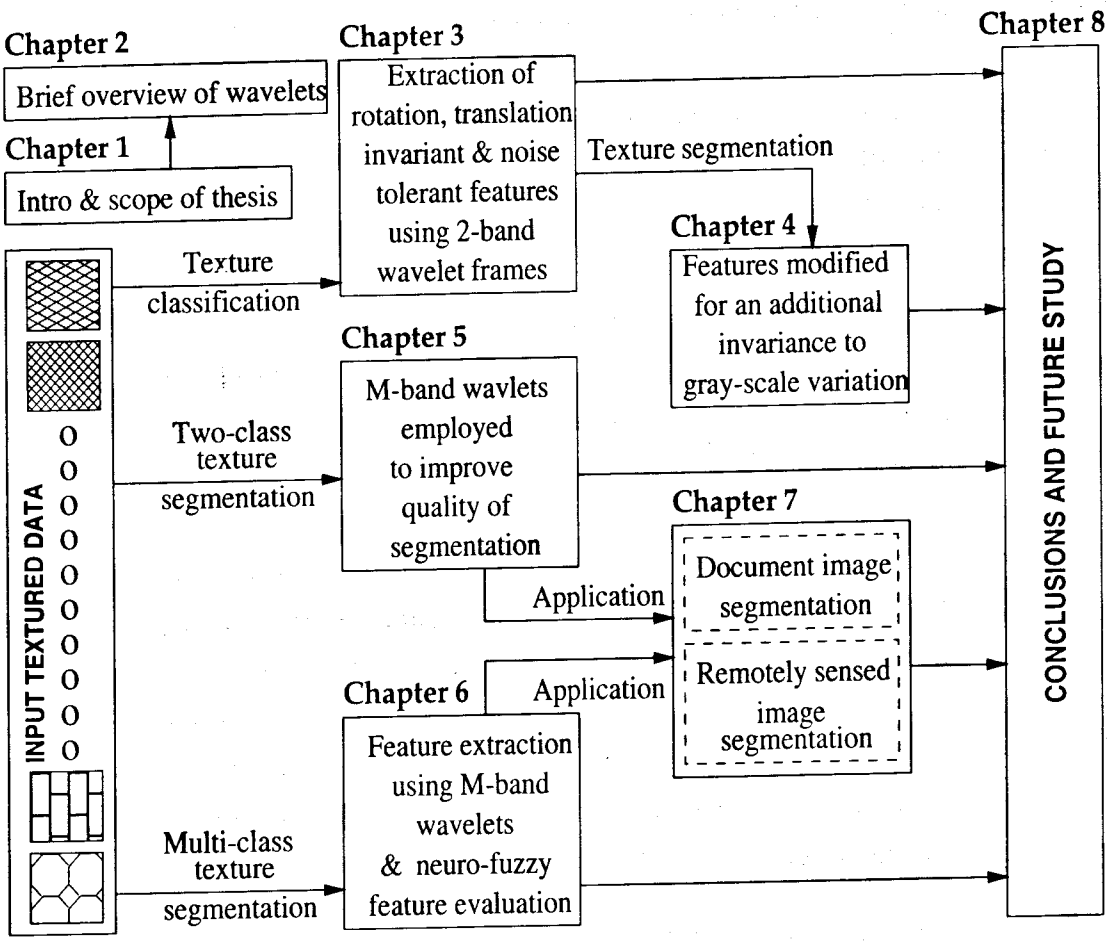


Figure 1.5: Flow diagram depicting the works addressed in this thesis

1.5.1 Multiresolution analysis and wavelet

In Chapter 2 we give a brief introduction about a general concept of multiresolution representation of a signal. We then discuss about wavelet, which is an efficient tool for multiresolution representation of signals. We furnish a brief discussion about the different types of wavelet transforms which includes wavelet packets and M-band wavelets. Although a detail discussion on wavelets is beyond the scope of the present thesis, but we will discuss only those topics, which will be required in the rest part of the thesis.

1.5.2 Robust texture classification using wavelet frames [1, 2]

Chapter 3 describes a theoretically very simple, yet efficient, multiresolution approach for extraction of features invariant to rotation and translation and tolerant to noise for texture identification. Here, we assume that, the images to be recognized may have arbitrary orientation and subjected to translational shifts and corrupted by noise. Let there be a finite number of texture classes K and a number of training images belonging to each class is available. Then, based on the information extracted from these images, a given test image of unknown class can be classified to one of the K classes.

To achieve this, we characterize textures at multiple scales using 2-band wavelet frames [1, 2]. A non-separable wavelet transform which gives an overcomplete/redundant wavelet decomposition (leading to wavelet frames) is employed. Theoretical arguments are given to explain why they are to be preferred to separable transforms. An obvious advantage of using non-separable transform is a more flexible filter design. This is exploited properly to build rotation as well as translation invariant and noise tolerant features using a circularly symmetric filter. Here, a texture is characterized by a set of values of variance of the different frequency bands in which it is decomposed. These values serve to provide the texture features that are used to perform classification of the textured images. In an extensive experimental study, the performance of the algorithm is demonstrated for classification over 24 independent texture images (Brodazt and MeasTex). Classification performance of the proposed methodology is compared with wavelet packet features used in [98].

The proposed approach is very robust in terms of rotational variations since by definition, the operator is invariant against any rotation (orientation) of the imaging sensor. Another advantage is its computational simplicity. The experimental results are obtained to show the effectiveness of the proposed technique for rotation invariance, where the classifier is trained for one particular rotation angle and tested with samples obtained with other rotation angles. Our study indicates that texture signatures based on wavelet frame analysis are beneficial for accomplishing subtle discrimination of textures and robust classification against rotation, translation and noise.

1.5.3 Wavelet frames for rotation and gray-scale transform invariant texture segmentation [3, 10]

In Chapter 4, we continue our discussion on the issue of extraction of texture features that are invariant to gray-scale transformations. These are also rotation invariant and noise tolerant. The filtering approach described in Chapter 3, for the extraction of rotation-invariant texture features is modified for any gray-scale variations. These features are then employed for texture segmentation tasks [3, 10]. To our knowledge, only a few significant attempts have been made to study gray-scale transform invariant texture analysis.

Spatial filtering is the key idea for the different texture segmentation schemes presented in this thesis. Chapter 4 briefly discusses the basic feature extraction scheme for texture segmentation used in the rest part of the thesis. The texture is decomposed into a set of bandpass channels by a circularly symmetric wavelet filter, which in essence gives a measure of edge magnitudes of the texture at different scales. Then it is characterized by local energies over small overlapping windows around each pixel at different scales. The (*k-means* clustering algorithm is then applied to this signature to achieve the desired segmentation ($k - means$ algorithm is applied for clustering in the rest of the thesis). Performance of the segmentation algorithm is evaluated through extensive testing over various types of test images.

1.5.4 M-band wavelet for two-class texture segmentation [4, 5]

Chapter 5 begins with the discussion on the role and effectiveness of M-band wavelet as a tool for image segmentation. We investigate the problem of segmentation of two-class texture images by using a generalization of the standard 2-band wavelet transform to the M-band case [4, 5]. It is logical to assume that the M-band wavelet transform has the efficiency to perform multiscale, multidirectional filtering of the images, since it is a tool for viewing signals at different directions and decomposes a signal by projecting it onto a family of functions generated from a single wavelet basis via its dilations and translations.

In texture segmentation a local neighborhood is investigated. The goal is to describe a group of pixels in some intelligent manner that lets similar and dissimilar groups

to be recognized. The feature extraction scheme consists of a filtering stage followed by a local energy estimator. The objective of the filtering is to transform the edges between textures into detectable discontinuities (*i.e.*, to extract local frequencies within a texture). While the purpose of the local energy estimator, is to estimate the energy of the filter output within a local region.

In the filtering stage we make use of an M-band wavelet system. The wavelet filter decomposes an image into a number of frequency subbands corresponding to different resolutions. The image is first transformed into $M \times M$ bandpass frequency channels by applying the M-band wavelet filters in a separable manner, without downsampling. This gives an overcomplete representation of an image (wavelet filter coefficients) along different orientations in the frequency plane at different scales. In the next step, a local energy estimator (consisting of a nonlinear operator and a smoothing filter) is applied to these bandpass sections. Size of the smoothing filter is to be chosen very carefully. For estimating the local energy of an image with low spatial frequency the smoothing filter must have a wide impulse response and vice versa. We have proposed an adaptive method to determine the size of the smoothing filter based on the spectral frequency content of the images. These operations result in a set of feature images (derived) from which a set of feature vectors are obtained. These feature vectors effectively characterize a texture at different scales.

Feature set may have redundancy and reduce the recognition score and increase computational cost. So, we have proposed a methodology, to rank the extracted features according to some feature quality index, and then select them by using a simple cluster quality index, which gives a sub-optimal set of features. Statistical tests are used to evaluate the average performance of extracted features. Superior classification performance of our algorithm over several other existing methods is established.

1.5.5 Wavelet feature extraction and their neuro-fuzzy evaluation for multi-class texture segmentation [6, 12]

In Chapter 6 a scheme for segmentation of multi-class texture image is discussed. The methodology involves extraction of texture features using an overcomplete wavelet decomposition scheme called discrete M-band wavelet packet frame (DMbWPF). Fi-

nally, this is followed by a section describing selection of important features using a neuro-fuzzy algorithm under unsupervised learning [6, 12].

It is previously shown in Chapter 5 that, the decomposition scheme based on M-band ($M > 2$) wavelets, gives a better representation of an image (signal) in terms of its constituent energies. In this work we use *discrete M-band wavelet packet frame* (DMbWPF). This gives an overcomplete/redundant representation of the image to accomplish translation invariance and accurate edge localization between different classes. This transform corresponds to a general tree-structured filter bank and generates a large number of independent bases (subbands). It is quite evident that an exhaustive search to determine an optimal basis from this large set is computationally expensive. In order to overcome this problem we propose an adaptive decomposition algorithm using a criterion based on maximum of textural measures extracted from each of the subbands. Then the most significant subbands are identified, and it is decided that whether further decomposition of the particular subband would generate more information or not. This search enables one to zoom into any particular desired frequency channel for further decomposition. Thus a full decomposition of the subband tree is avoided.

The filtering technique for feature extraction described above gives rise to a large number of features. As mentioned earlier that a large feature set may decrease the recognition quality, so our efforts have been to characterize a texture most efficiently by finding out a low dimensional and best set of features from the extracted feature set. Therefore, after the extraction of the wavelet features, these are evaluated and then selected using a computationally efficient neuro-fuzzy algorithm. This involves minimization of a fuzzy feature evaluation index in a connectionist framework. It is to be mentioned here that, both the feature extraction and neuro-fuzzy feature evaluation methods are unsupervised, and do not require the knowledge of number and distribution of classes.

The superior discriminating capability of the extracted features over those obtained by several other existing techniques is demonstrated for segmentation of various types of texture images containing 5 to 16 overlapping texture classes and some natural scenes.

1.5.6 Segmentation of document and remotely sensed images [7, 8, 9, 11, 13]

Chapter 7 gives two examples of real life applications of the texture segmentation algorithms developed in Chapters 5 and 6. In the first part of the chapter, an efficient and computationally fast method for segmenting text and graphics part of document images based on textural cues is presented [7, 8, 11]. It is logical to assume that the graphics part has different textural properties than the non-graphics (text) part. We have used M-band wavelet feature extraction scheme developed in Chapter 5 for this purpose. It is to be mentioned that no *a priori* information regarding the font size, scanning resolution, type layout etc. of the document is assumed in our scheme

The later part of the chapter describes how the features extracted in Chapter 6 above can be used for segmenting real life data such as remotely sensed images. Different quasi-homogeneous regions in the image can be treated to have different texture properties. Based on this assumption the multi-class texture segmentation scheme is applied for this purpose. The effectiveness of the methodology is demonstrated on two 4-band IRS-1A images containing 5 to 6 overlapping classes and a 3-band SPOT image containing 7 overlapping classes [9, 13].

1.5.7 Conclusion

The concluding remarks along with scope for further research are presented in Chapter 8.

♣

Operation of the neural network used in Chapter 6 is given in the Appendix.

Chapter 2

Multiresolution Analysis and Wavelets

2.1 Introduction

This chapter focuses on the part of wavelet theory which is of our interest mainly for texture classification/segmentation problems. The philosophy behind the use of wavelet packet and M-band wavelet is described. We will not go into all the details, but we will restrict ourselves to the topics that are important for the rest of this thesis. We begin this chapter defining the concept of multiresolution analysis in general, and extend our discussions on wavelet, which is an efficient tool for multiresolution analysis.

2.2 Multiresolution Analysis

In computer graphics applications, a continuous function is represented using a set of discrete samples. So, a computer with finite storage capacity, can represent an image by a finite collection of samples, and will in general, be an approximation of the original. The samples may be used to reconstruct the function that is defined over a particular domain by treating them as coefficients for a corresponding set of basis functions defined over the same domain. Thus, given the discrete samples, one can evaluate the continuous function everywhere, not just at the locations of the samples. The size of the intervals between samples dictates the resolution of the representation and the degree to which it is able to describe (resolve) fine details in the function it represents. Tiny details of frequencies higher than half the sampling rate (the Nyquist limit) cannot be resolved. When we zoom in to a signal/image there is no more fine detail to be seen; the signal/image usually looks blurry/blocky, depending on the nature of basis functions used. In the case of an image, for example, the pixels are usually organized in a regular rectangular grid. We call this a uniresolution representation, which is a finite, discrete set of samples spaced uniformly over the domain of the function they represent.

Uniresolution representations are advantageous for a number of reasons. They are simple, easy to understand, and easy to implement. Furthermore, they often make use of implicit data structures such as arrays, so there is no extra storage overhead beyond the actual data being represented. Finally, these representations usually have a predictable size, which is advantageous for storage and transmission. However, uniresolution representations suffer from a number of limitations, which are as follows (since

we are dealing with image in this thesis so, henceforth we would be mentioning image instead of function):

- An *a priori* decision about the resolution of an image is required. It is difficult to know in advance the extent of finer detail information present in the image to be represented. An incorrect guess of sampling frequency (resolution) may either result in loss of information or redundancy.
- Once a specific resolution is decided, no finer detail information higher than this can be achieved.
- Different parts of an image may differ in information content. That is, different parts of the image must be represented by different number of samples according to their region wise information content.

These limitations can be alleviated by using a multiresolution representation by representing different parts of an image with different resolution based on the requirement. In addition, multiresolution representations often yield following advantages over uniresolution representations:

- **Control**. Multiresolution representations facilitate manipulation of data at different levels of detail.
- **Feature detection**. Multiresolution representations can facilitate identification of salient features in the images they represent.
- **Compression**. Multiresolution representations lead naturally to compression schemes that maintain only salient features and ignore areas of little or no detail.
- **Refinement**. Multiresolution representations permit addition of fine detail to an image without converting the image to a higher resolution.

From the very early days of signal and image processing, it has been recognized that multiresolution methods are important for following reasons:

- There is strong evidence that the human visual system processes information in a *multiresolution* fashion and try to fuse the information available at different scale;

- Signals usually consist of features of physically significant structure at different resolutions;
- Sensors may provide signals of the same source at multiple resolutions;
- *Multiresolution* algorithms offer computational advantages and, moreover, appear to be robust.

In the earliest multiresolution approaches to signal and image processing, a *coarse resolution signal* was generated by sub-sampling a *fine resolution signal* [167]. A *pyramidal representation* can then be obtained by subtracting from each level an interpolated version of the next coarser level; the best known example is the *Laplacian pyramid* [29]. From a frequency point of view, the resulting difference signals (known as *detail signals*) form a signal decomposition in terms of bandpass filtered versions of the original signal. Moreover, there is neuro-physiological evidence that the human visual system indeed uses a similar kind of decomposition [107]. This tool has been one of the most popular multiresolution schemes used in image processing and computer vision. More information about multiresolution processing can be found in [141].

Wavelet analysis provides an effective multiresolution representation of a signal/image, and is capable of quickly capturing the essence of a data set with only a small set of coefficients.

2.3 Wavelets: The State of Art

The subsequent sections, discuss about wavelets from two different aspects. The first is the ability of the wavelet functions to span both spatial and frequency domains. The second aspect is to establish the link between multiresolution analysis and wavelet decomposition theory. Mallat [107] has shown that a signal at any resolution can be decomposed into a signal at a coarser resolution, plus the associated wavelet components. In other words, the wavelet transform of a signal at a finer resolution, can be interpreted as the difference between the information content of the signal at two successive resolutions. The multiresolution characteristics of the wavelet transform is very attractive to different image processing and vision applications.

The word "wavelet" was first proposed by the French geophysicist Morlet in the early eighties. The seismic data he was studying, exhibited rapidly changing frequency contents for which Fourier analysis did not suffice as an analyzing tool. He investigated the possible existence of functions, which had sufficient compact support in both time and frequency domain. He called them wavelets, meaning "small waves". Grossman gave the mathematical basis of Morlet's idea [74] and this has triggered the attempt for the construction of a complete mathematical framework, which is known as the wavelet theory. A rapidly growing number of applications of this framework is now emerging fast.

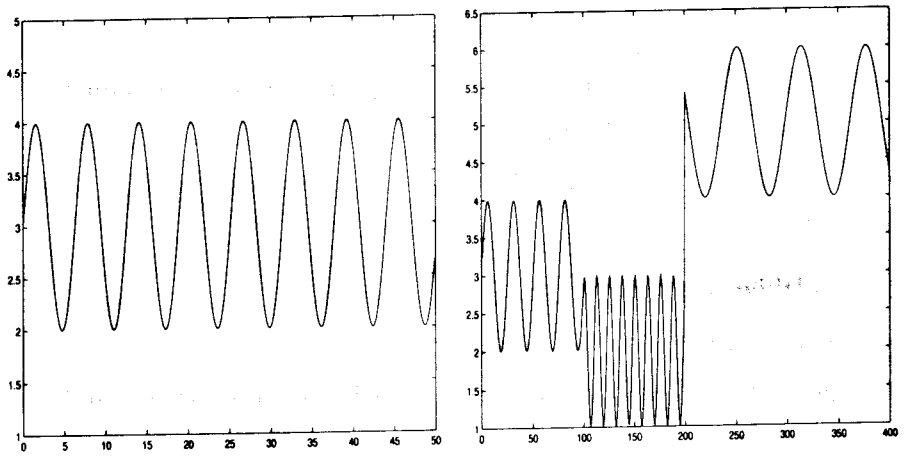
Wavelet theory is not wholly new, rather it is the result of cross-fertilization of several fields and generalizes concepts known in geophysics, signal/image analysis and compression, physics, mathematics and statistics. Some essential results that lead to the development and enrichment of this field are mainly due to Ingrid Daubechies a Belgian theoretical-physicist and Stephane Mallat, a French mathematician-signal analyst. Daubechies has constructed families of orthonormal wavelets with compact support [49]. Mallat put the wavelet transform in the framework of multiresolution signal decomposition and has provided a fast implementation of the transform based on concepts of filtering theory [107]. All this research have resulted in a comprehensive mathematical framework for signal/image analysis with several applications and beyond the scope of this thesis. A thorough and detailed mathematical analysis is given in [50] and several other books and review papers are available on this topic [139, 152, 28, 157].

Before we go into the description of wavelet theory, we shall discuss in brief, about the time frequency representation of a signal, and which lead to the development of the wavelet theory.

2.4 Time Frequency Representation of Wavelet

An important issue in the analysis of signals is their representation. The representation should be such that any kind of information required, can be easily accessible from it. For instance, in signal analysis it is very often necessary to study the frequency content of a signal. One of the classical tools to extract such information is the Fourier

analysis, which maps a signal $f(t)$ from the time domain to the frequency domain (global)[1]. The Fourier Transform assumes that the signal to be analyzed is either of infinite duration or is at least of one complete period. This condition is often not met in real world signals. Let us consider for example, a signal which exists for all time (Fig 2.1(a)). The signal is clearly periodic and the Fourier transform is able to analyze its frequency contents. In Fig. 2.1(b) the signal contains three different frequencies which exist during three different intervals of time. Fourier analysis is not appropriate for the analysis of this signal, and is not able to convey which frequency is present during what interval of time. This example clearly points out the need for a localized *time-frequency* representation of the signal to be analyzed.



(a) (b)

Figure 2.1: Example of 2 signals with different time-frequency characteristics.

An obvious way to obtain such a representation is to place a localized window on the signal which is shifted along the time axis and to perform a Fourier Transform (FT) at each point in time. This yields the Windowed Fourier Transform (WFT). If we define the Fourier Transform of a function $f(t)$ by

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i u t} dt \tag{2.1}$$

where u represents the frequency, the WFT will be

$$W_F(u, T) = \int_{-\infty}^{\infty} f(t) g(t - T) e^{-2\pi i u t} dt \tag{2.2}$$

[1]In case of images, one talks of the *spatial domain* $f(x, y)$ and the *frequency* domain $\hat{f}(u, v)$

$g(T)$ is a window centered around T. Whereas the classical Fourier analysis uses the *plain waves* $e^{-2\pi iut}$ (the so called *analyzing functions*) as the elementary "building blocks" to decompose the signal, the WFT uses the set $G_{uT}(t) = g(t-T)e^{-2\pi iut}$ which maps the signal on a two-dimensional (2D) representation; which is the *time-frequency plane*. The coefficients $W_F(u,T)$ will have large magnitude when the signal $f(t)$ resembles the characteristics of a member of the analyzing function (i.e. similar time localization and frequency content).

From (2.2) and Fig. 2.2(a) we can see that the WFT is the inner product of $f(t)$ with a set of functions

$$G_{u,T} = g(t-T)e^{-2\pi iut} \tag{2.3}$$

The window function, $g(t)$ should be long enough so that the product $f(t)g(t)$ contains some useful frequency elements in the application under study. But this function cannot be too long as it would defeat the idea of windowing the signal (*i.e.*, localization in time). Therefore, the time and frequency localization of the WFT should be carefully designed.

The main drawback of the WFT, however, is that once the width of the window function is decided it cannot be changed during the entire duration of a particular analysis. In other words we have a fixed resolution in time because the same window function $g(t)$ has been translated in time. We also have a fixed resolution in frequency, because the time resolution is fixed (time resolution determines the frequency resolution and vice versa. Fig. 2.3(a) schematically gives the *tiling* of the time-frequency plane of WFT. It is not possible to obtain simultaneously arbitrary fine localization in time and frequency due to the uncertainty principle. The blocks tiling the time-frequency plane all have the same area and thus, once the width of the window $g(t)$ is chosen, this tiling is fixed. In nature however, we encounter signals which contain multiresolution information both in time and frequency.

The wavelet transform is another time-frequency representation for signals with multiresolution properties. One might desire to study the slowly varying properties of a signal (low frequencies) over a longer time span and vice versa for the high frequencies. for these representation wavelets are particularly suited. Such a tiling is represented in Fig. 2.3(b) and is provided by employing wavelets as analyzing functions.

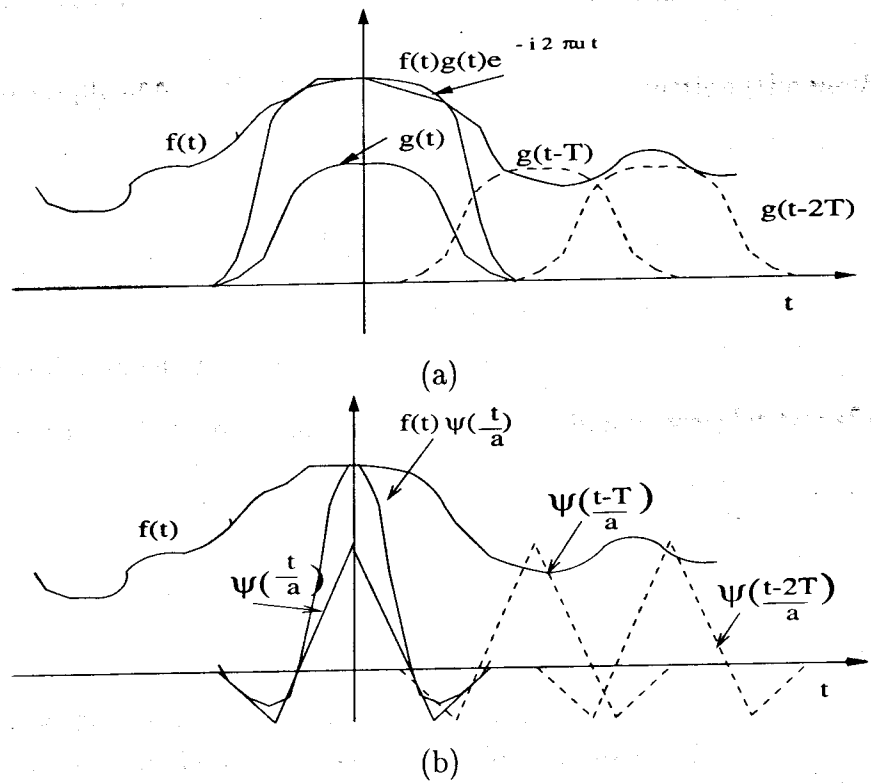


Figure 2.2: (a) The Windowed Fourier Transform (b) The Wavelet Transform.

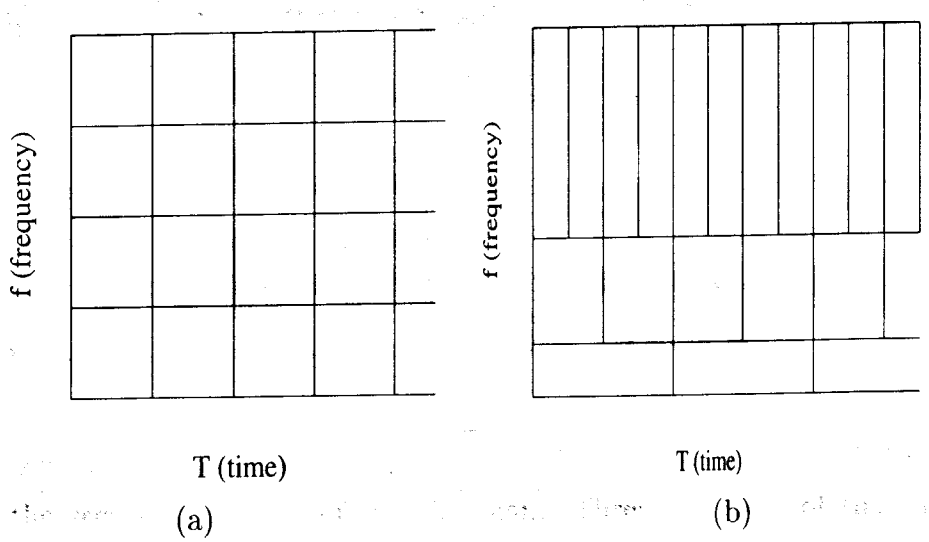


Figure 2.3: Tiling of the time frequency plane: (a) as obtained by the WFT (b) as obtained by the wavelet transform.

2.5 Continuous Wavelet Transform

The wavelet family of functions is derived from one single function (the *mother* wavelet) which is indexed by two parameters a and b.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{2.4}$$

The parameter a gives the dilation (which is inversely proportional with frequency) and b the displacement (time localization).

In analogy with (2.2) the one-dimensional (1D) continuous wavelet transform (CWT) is defined by

$$Wf_a(b) = \int_{-\infty}^{\infty} f(t)\psi^{\star}_{a,b}(t)dt = \langle f, \psi_{a,b}\rangle, a \neq 0 \tag{2.5}$$

and the inverse wavelet transform is given by

$$f(t) = C_{\psi}^{-1} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \langle f, \psi_{a,b}\rangle \psi_{a,b}(t)\frac{dadb}{a^2} \tag{2.6}$$

where we require that the wavelet satisfies (*admissibility condition*)

$$C_{\psi} = 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(u)|^2}{|u|}du < \infty \tag{2.7}$$

If $\hat{\psi}$ is continuous, condition (2.7) is reduced to

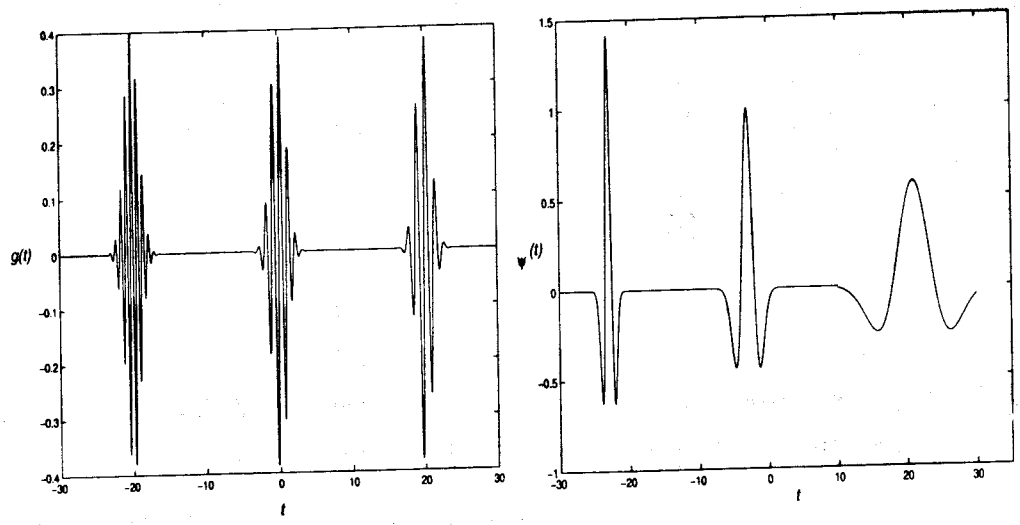
$$\hat{\psi}(0) = 0 \Leftrightarrow \int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{2.8}$$

This shows that a wavelet is a zero mean function and hence exhibits some oscillatory behavior. It can be proven that the admissibility condition is necessary to ensure that the wavelet transform is invertible (i.e. that (2.6) makes sense).

A classical example of such a family is the *Mexican hat* given by

$$\psi(t) = (1-t^2)e^{-t^2/2} \tag{2.9}$$

which is the second derivative of the Gaussian. Three members of this family are represented in Fig. 2.4(b) and compared to the decomposition functions of the WFT in Fig. 2.4(a). It shows that the G_{uT}'s all consist of the same window g filled in with high frequency oscillations. All the G_{uT}'s have the same width regardless of their



(a) (b)

Figure 2.4: Example of analyzing functions of (a) the WFT (real part of G_{uT} (b) the CWT (Mexican Hat wavelet). From left to right: high to low frequency.

frequency. The wavelets ψ_{ab} on the other hand have the time-width adapted to their frequency, high frequency wavelets are very narrow and vice versa.

As it can be seen by comparing Figs. 2.2(a) and 2.2(b), the similarity between the Windowed Fourier and Wavelet Transforms is that both equation (2.2) and (2.5) take the inner product of $f(t)$ with family of windowing functions indexed by two labels. The indices in (2.3) are time and frequency and in (2.4) are time and scale. The scale, however, represents different frequency ranges because large values of the scaling parameter, a, correspond to small frequencies and small values of it correspond to large frequencies. The value of a also affects the length of $\psi(\frac{t}{a})$ in time as shown in Fig. 2.4(b). For larger a the wavelet becomes "broader" in the time domain and thus "smaller" in the frequency domain and hence it will capture more low frequency characteristics of the signal.

2.5.1 How do wavelets provide a variable size window?

We can consider the function ψ in (2.4) as a window function. Formally, a non-trivial function $w \in L^2(\Re)$ is called a window if $tw(t)$ is also in $L^2(\Re)$. By non-trivial we mean not a zero or a constant function. The center c_w and radius Δ_w of the window

function $w(t)$ is defined as follows

$$c_w = \frac{1}{||w||^2} \int_{-\infty}^{\infty} t|\ w(t)\ |^2 dt \tag{2.10}$$

$$\Delta_w{}^2 = \frac{1}{||w||^2} \int_{-\infty}^{\infty} (t - c_w)^2 |\ w(t)\ |^2 dt \tag{2.11}$$

where, $||w||$ is the $L^2(\Re)$ norm. The width of the window is $2\Delta_w$. Let ψ be a wavelet function such that both ψ and $\hat{\psi}$ are window functions. Let their centers be c_ψ and $c_{\hat{\psi}}$ and radii be Δ_ψ and $\Delta_{\hat{\psi}}$ respectively. It follows from the above definitions (2.10) and (2.11) that the center and radius of $\psi(\frac{(t-b)}{a})$ are respectively $b + ac_\psi$ and $a\Delta_\psi$.

Applying Parseval's identity to RHS of (2.5) we have

$$\frac{1}{|\ a\ |^2} \int_{-\infty}^{\infty} f(t)\psi^*(\frac{(t-b)}{a})dt = \frac{1}{2\pi|\ a\ |^2} \int_{-\infty}^{\infty} \mathcal{F}(\omega) a e^{ib\omega} \eta^*(a\omega - \omega_0) d\omega \tag{2.12}$$

where, $\eta(\omega) = \hat{\psi}(a\omega + \omega_0)$ and ω_0 is constant. The function $\eta(a\omega - \omega_0)$ provides a window along the frequency axis. It can be shown that the center of this window is ω_0 and its radius is $\frac{\Delta_\psi}{a}$. We choose $\omega_0 = c_{\hat{\psi}}$. The width of the frequency window is $\frac{2\Delta_{\hat{\psi}}}{a}$ and the width of the time window is $2a\Delta_\psi$. Hence the area of the time-frequency window is, $4\Delta_\psi\Delta_{\hat{\psi}}$ which is constant once we choose the mother wavelet ψ. Therefore, the most important observation from this is that, as scale a decreases, the width of the frequency window increases, which is shifted towards the higher frequencies while the time spread decreases. Thus the window automatically adjusts for small width in time domain at high frequencies and vice versa.

Note that the CWT defined above is essentially redundant which means that there is no one-to-one mapping from the function space $\mathrm{L}^2(R)$ to the space of all wavelet transforms. However, this redundancy can be reduced or eliminated by constraining the choice of wavelet ψ and by discretizing the parameters a and b. This leads to the discrete wavelet transform (DWT) which allows computing a wavelet transform in an efficient way.

The different DWT schemes can be subdivided into two types:

1. Redundant: If a digitized signal is represented by N samples, then a redundant DWT maps it on $M > N$ samples. This leads to the theory of wavelet frames, an example of which will be discussed in Section 2.6.

2. Non-redundant: In this case, the wavelets, which form an orthogonal basis are used. It is a particularly interesting case since construction of a fast algorithm is possible to perform the DWT. This case will be discussed in Section 2.7.

2.6 Redundant Discrete Wavelet Transform

2.6.1 Wavelet frames

Plugging in (2.4) and (2.5) one can notice that the wavelet transform is essentially a convolution of $f(t)$ with scaled wavelet $\psi_a(t)$. By the convolution theorem, this becomes a multiplication in the frequency domain. Hence for sampled signals, the transform is carried out by employing a Fast Fourier Transform (FFT), multiplication and Inverse Fast Fourier Transform (IFFT). The computational complexity for this certainly becomes very high for images.

It is therefore desirable to discretize the transform by restraining a and b to a discrete grid: *i.e.*, $a = a_0^j$ and $b = kb_0a_0^j$ with $j, k \in Z$ (where, Z is the set of all integers) and $a_0 > 1$ and $b_0 > 0$ fixed. A popular choice is $a_0 = 2$ and $b_0 = 1$. Since the scale varies by powers of 2 this yields a so called *dyadic* frequency progression. The advantage of this choice becomes evident while considering a digitized signal/image where, each sample is taken at equidistant time intervals. The evaluation of the DWT coefficients, is then to be taken at positions $b = k2^j$, which coincide with the positions of (some) samples of the original signal. However, other discretizations are possible and sometimes useful.

By discretizing a transform, its inverse may not be readily computable. Thus the question arises how to reconstruct the original function from its wavelet transform. A detailed analysis shows that, to be able to obtain a stable reconstruction, the family of analyzing functions $\{\psi_{jk}\}_{j,k\in Z}$ should constitute a *frame*, which means that there should exist two constants A and B such that $0 < A \leq B < \infty$ and

$$A||f||^2 \leq \sum_{j,k} |\langle f, \psi_{j,k}\rangle|^2 \leq B||f||^2 \quad \forall f \in \boldsymbol{L}^2(R) \tag{2.13}$$

A and B are called the *frame bounds*. When $A = B$ we get a *tight frame*. Frames are essentially a generalization of a basis (an orthonormal basis is a tight frame for which

($A = B = 1$). The term B/A measures the deviance of a frame from a tight frame. Good reconstruction properties are obtained with frames for which $B/A < 2$. Several examples of frames can be found in literature (see [50] for an overview).

2.6.2 A dyadic discrete wavelet transform

We will now discuss about a particular wavelet transform introduced by Mallat [109] (which has been used in chapters 3 & 4 in the present thesis). It is based on a discrete wavelet frame with a dyadic decomposition. The advantage of this scheme is that it can be implemented by applying Finite Impulse Response (FIR) filters (*i.e.*, filters which have a finite number of non-zero coefficients) and this circumvents the use of an FFT.

This transform takes a dyadic frequency progression ($a = 2^j$) and relies on a specific form of the wavelet, which is derived from a smoothing function. Let $\phi(t)$ be such a (1D) smoothing function. Assuming ϕ is differentiable, one can define a wavelet as

$$\psi(t) = \frac{\partial \phi(t)}{\partial t}. \tag{2.14}$$

The wavelet transform of a signal $f(t)$ can then be written by

$$Wf_a(b) = (f * \psi_a)(b). \tag{2.15}$$

Combining (2.14) and (2.15) one can write

$$Wf_a(b) = (f * \psi_a)(b) = (f * (a\frac{d\phi_a}{dt}))(b) = a\frac{d}{dt}(f * \phi_a)(b). \tag{2.16}$$

This yields an interesting interpretation of the $Wf_a(b)$. For fixed a, $Wf_a(b)$ contains the gradient of the signal which is smoothed by ϕ_a. If a becomes larger, f will be smoothed more and Wf_a contains gradient information concerning the lower frequency variations of the signal. Wf_a thus provides specific information regarding the signal at the particular scale a, and is therefore called the *detail signal.*

To obtain a dyadic frequency progression, the scale and dilation parameters are discretized as $a = 2^j$ and $b = k$ ($j, k \in Z$). In practice, the scale can not be varied over all possible values 2^j ($j \in Z$). Let the scale at which the signal $f(t)$ is digitized be equal to $j = 0$. Then scales $j > 0$ and shifts $k = 1, 2, \ldots$ correspond to coarser

scales (or higher scales or lower frequencies or coarser resolutions) in which less signal information are present. This process of increasing scale can not go on indefinitely. This scale limitation is dealt with by introducing a *scaling function* $\phi(t)$, which is also scaled dyadically and used to compute the *low resolution signal*

$$Lf_{2^j}(t) = f * \phi_{2^j}(t). \tag{2.17}$$

ϕ is constructed in such a way that it contains all information that is not present in the detail signal. Finally, the sequence of discrete signals $\{Lf_{2^J}, \{Wf_{2^j}\}_{1 \leq j \leq J}\}$ is a (discrete, dyadic, redundant) wavelet representation of the signal $f(t)$.

A practical advantage of this wavelet representation is that the detail and low resolution signals can be computed fast and easily. This is performed by filtering the signal by a set of FIR filters, associated with the analyzing functions ϕ and ψ.

A pseudo code for performing a wavelet decomposition and reconstruction of a signal $f(t)$ using FIR filters is given below. Such filters are noted as $filt$ $(h,\ g,\ k,\ \tilde{h})$. $filt_j$ at different scales are obtained by putting $2^j - 1$ zeros between filter-taps of $filt$.

decomposition
for$(j = 0; j < J; j = j + 1)$
{
$Lf_{2^{j+1}}(t) = h_j * Lf_{2^j}$
$Wf_{2^{j+1}}(t) = g_j * Lf_{2^j}(t)$
}
reconstruction
for$(j = J; j > 0; j = j - 1)$
{
$Lf_{2^{j-1}}(t) = k_j * Wf_{2^j} + \tilde{h}_j * Lf_{2^j}$
}

A class of such functions and the associated filters are described in [109]. The extension of the decomposition scheme to images (*i.e.*, when the signal f is two-dimensional) is dealt with, by applying the above filtering scheme along the rows and columns of the image. The relevant formulas will be given in the later part of this chapter.

2.7 Non-redundant Discrete Wavelet Transform

The wavelet transforms presented in the previous section are all redundant. Although this redundancy can be exploited to obtain stable representations, but the major drawback is that the transforms are computationally inefficient (the convolution requires a FFT, multiplication and IFFT). The transform described in Section 2.6.2 is still redundant but already faster since it can be implemented by FIR filters in the time domain. However, it would be computationally efficient to get rid of this redundancy altogether. At this juncture, one requires a wavelet basis, which is orthogonal (or orthonormal).

A possibility is the *Harr wavelet*

$$\psi^h(t) = \begin{cases} 1 & \text{for } 0 < t < 1/2 \\ -1 & \text{for } 1/2 \leq t < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.18}$$

for which it is easy to verify that $\{\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k)\}_{j,k\in Z}$ is an orthonormal basis for $\boldsymbol{L}^2(\Re)$. Note that, here discretization parameters $a = 2^j$ and $b = k2^j$ are used which will be pursued throughout this section. The Harr wavelet was historically speaking the first known wavelet, but is not often used since it suffers from some drawbacks. First, it is not a smooth function (it is not even continuous), and is therefore not a suggestive basis to decompose smoothly varying signals. Second, the Harr wavelet is well localized in the real domain, but a strongly oscillating and very badly localized function in the frequency domain. Many researchers have attempted to find orthonormal wavelet basis which are smoother and have better localization properties in both time and frequency domain. We mention here the construction proposed by Battle [21]. Better localization properties are reported in the Daubechies wavelet-families [45].

The beauty of all these constructions is that they can be put in the framework of a *multiresolution analysis* conceived by Meyer and Mallat, which has been the motivation for construction of yet other wavelet bases. The details are found in [107] and are summarized in the next section.

2.8 Multiresolution Analysis using Wavelets

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

Let us consider a signal $f \in \boldsymbol{L}^2(\Re)$ and introduce a projection operator P_j such that $P_j f$ is the approximation of f at the resolution j. If we associate this j with a such that, $a = 2^j$ then "higher j" means "lower resolution". It can be proven that P_j is an orthogonal, linear operator and that $P_j f$ is an element of the vector space $\mathcal{V}_j$ defined by

$$\mathcal{V}_j = \{P_j f | f \in \boldsymbol{L}^2(\Re)\}. \tag{2.19}$$

A multiresolution analysis of f is then performed by projecting f on successive lower resolution approximations.

Formally, a *multiresolution approximation* of $\boldsymbol{L}^2(\Re)$ consists of a series of nested subspaces $\mathcal{V}_j \subset \boldsymbol{L}^2(\Re)$ for which

$$\ldots \mathcal{V}_{-2} \subset \mathcal{V}_{-1} \subset \mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \ldots \tag{2.20}$$

with

$$\overline{\bigcup_{j \in Z}} \mathcal{V}_j = \boldsymbol{L}^2(\Re) \text{ and } \bigcap_{j \in Z} \mathcal{V}_j = 0. \tag{2.21}$$

In order to make this ladder of subspaces a multiresolution representation, we require

$$f(t) \in \mathcal{V}_j \quad \Leftrightarrow \quad f(t/2) \in \mathcal{V}_{j+1} \quad \forall j \in Z, \tag{2.22}$$

and we also require invariance of $\mathcal{V}_0$ under translation, such that

$$f(t) \in \mathcal{V}_0 \quad \Leftrightarrow \quad f(t-k) \in \mathcal{V}_0 \quad \forall k \in Z. \tag{2.23}$$

This means that the space containing high resolution signals will consist of those of low resolution signals also. This ensures elements in a space are simply scaled versions

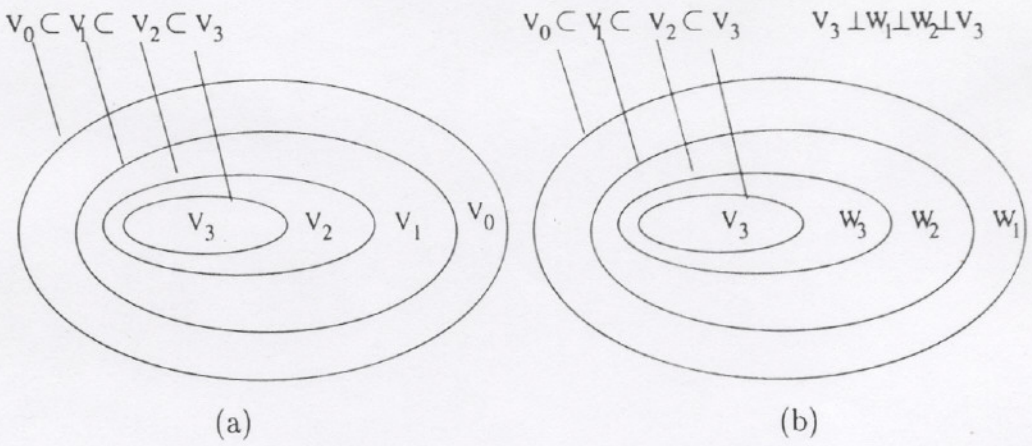


Figure 2.5: Nested subspaces spanned by the (a) scaling function (b) scaling and wavelet vector spaces.

of the elements in the next space. The nesting of the subspaces is illustrated in Fig. 2.5(a), and is achieved by requiring that if a function $\phi(t) \in \mathcal{V}_0$, then it is also in $\mathcal{V}_1$, the space spanned by $\phi(t/2)$. This means that $\phi(t)$ can be expressed in terms of a weighted sum of shifted $\phi(t/2)$ as

$$\phi(t) = \sum_k h(k) 2^{-1/2} \phi(2^{-1}t - k), \tag{2.24}$$

where the coefficients $h(k)$ are a sequence of real (may be complex) numbers called *scaling function coefficients* or *scaling filter*. Finally, we require that there exists some $\phi(t) \in \mathcal{V}_0$ such that

$$\{\phi_{0,k}\}_{k \in Z} \text{ is an orthonormal basis for } \mathcal{V}_0. \tag{2.25}$$

The latter (2.25), together with (2.22) leads to a two dimensional family of functions generated from $\phi(t)$ by scalings and translations

$$\{\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t - k)\}_{k \in Z}. \tag{2.26}$$

$\phi_{j,k}(t)$ is an orthonormal basis for $\mathcal{V}_j$. The function $\phi(t)$ is called the *scaling function* (or sometimes the father wavelet). An example of a function projected on several $\mathcal{V}_j$ is presented in Fig. 2.5(a).

The process of projecting f from $\mathcal{V}_j$ to $\mathcal{V}_{j+1}$ results in loss of signal information. We now define the space $\mathcal{W}_{j+1}$ to be the orthogonal complement of $\mathcal{V}_{j+1}$ in $\mathcal{V}_j$, *i.e.*,

$$\mathcal{V}_j = \mathcal{V}_{j+1} \oplus \mathcal{W}_{j+1} \tag{2.27}$$

The projection of f on $\mathcal{W}_{j+1}$ then contains the information lost between the transition from V_j to $\mathcal{V}_{j+1}$. In other words: $\mathcal{W}_{j+1}$ contain the difference of information between $P_j f$ and $P_{j+1} f$ and is therefore called the *detail space at scale j.*

It can be shown that for each multiresolution approximation (with corresponding basis), there exists an orthonormal basis of wavelet functions $\{\psi_{j,k}\}_{j \in Z}$ for each subspace $\mathcal{W}_j$, which can be used to compute the projection of f onto $\mathcal{W}_j$. Fig. 2.5(b) pictorially shows the nesting of the scaling function spaces $\mathcal{V}_j$ for different scales j and how the wavelet spaces are the disjoint differences or, the orthogonal complements. Since these wavelets reside in the space spanned by the next higher scaling function, $\mathcal{W}_1 \in \mathcal{V}_0$, they can be represented by a weighted sum of shifted scaling function $\phi(t/2)$ defined in (2.24) by

$$\psi(t) = \sum_k g(k) 2^{-1/2} \phi(2^{-1}t - k) \tag{2.28}$$

for some set of coefficients $g(k)$ called *wavelet filters.* The function generated by (2.28) gives the prototype or the *mother wavelet* $\psi(t)$ for a class of expansion functions of the form

$$\{\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k)\}_{k \in Z}. \tag{2.29}$$

In practical applications, the signals to be analyzed are given in finite intervals of Z. Since every physical recording device has a finite resolution, the samples $f(n)$ represent f at the highest possible scale. Let us arbitrarily set this scale to $j = 0$ (we thus assume that $f \in \mathcal{V}_l$). It is then only of interest to compute the $P_j f$ for $j > 0$. By (2.27) we may write

$$\mathcal{V}_0 = V_J \bigoplus_{j=1}^{J} \mathcal{W}_j \tag{2.30}$$

and, assuming $f(t) \in \mathcal{V}_0$, it can be decomposed as follows

$$f(t) = \sum_k L_J(k) \phi_{J,k}(t) + \sum_{j=1}^{J} \sum_k D_j(k) \psi_{j,k}(t). \tag{2.31}$$

The first term in this expression is the *low resolution signal* $L_J(k)$; and the coefficients $D_j(k)$ constitute the *detail signal* at scale j for $k \in z$. The whole of wavelet coefficients $\{L_J, \{D_j\}_{1 \leq j \leq J}\}$ is called the *wavelet representation* at *depth* J of signal f. An example of a wavelet representation is presented in Fig. 2.6.

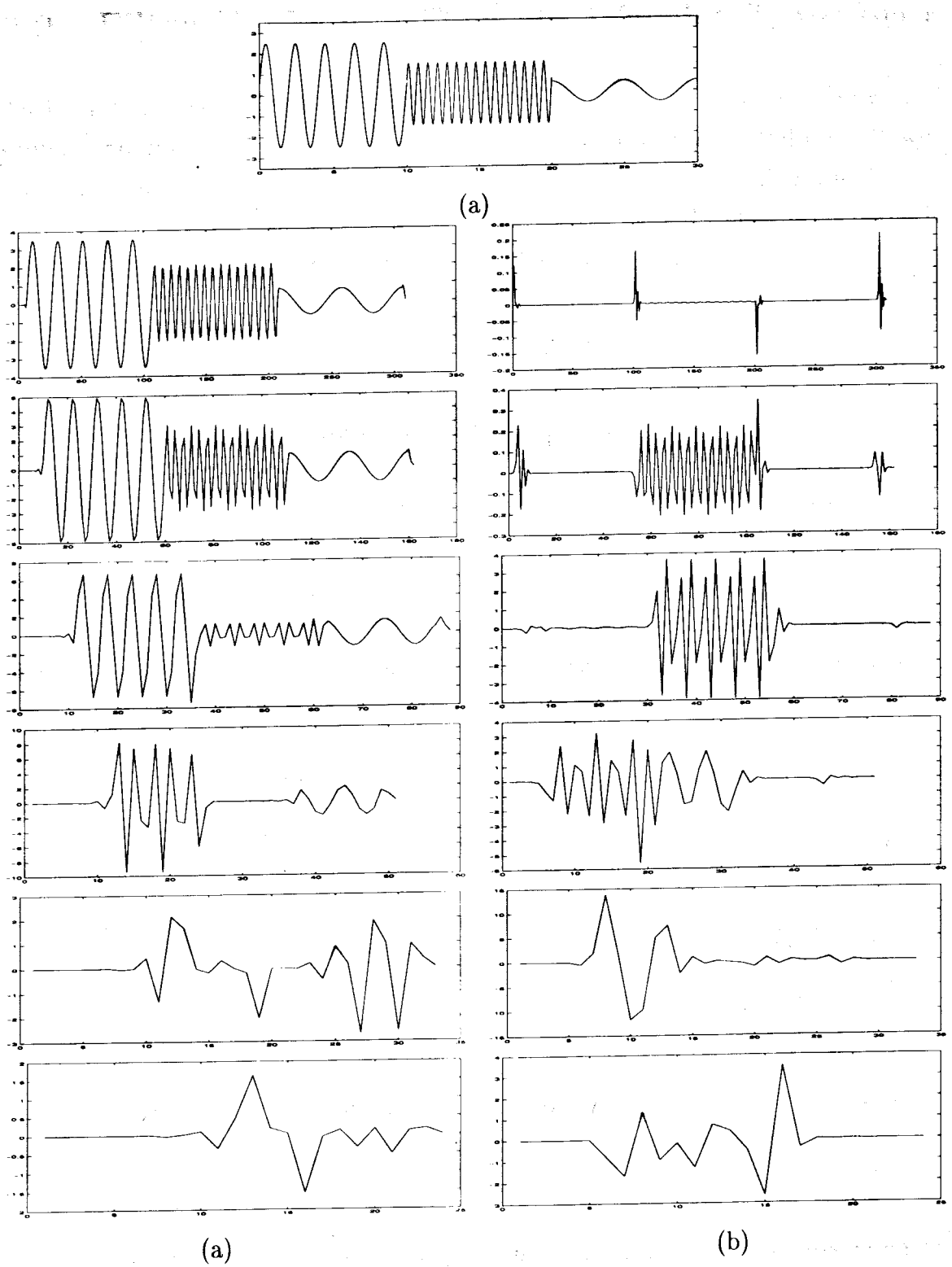


Figure 2.6: Example of a one dimensional (1D) DWT of depth 6 (a) original signal (b) low resolution signals (c) detail signals (with increasing resolution).

2.9 Filter Banks and Discrete Wavelet Transform

An important step in wavelet theory is the connection of this decomposition with subband filtering schemes. In many applications one never has to deal directly with the scaling functions or wavelets. Only the coefficients $h(n)$ and $g(n)$ in the defining equations (2.24) and (2.29) and $L_J(k)$ and $D_j(k)$ in (2.31) need to be considered. The filter bank point of view is necessary for a real understanding of this new tool.

2.9.1 Discrete wavelet transform in one-dimension

The wavelet decomposition can be interpreted as signal decomposition in a set of independent, spatially oriented frequency channels. It turns out that the wavelet coefficients can be computed by iterative filtering of the signal [107]. Under these constraints an efficient real space implementation of the transform using quadrature mirror filter (QMF) exists [107]. This consists of a low pass filter h and a high pass[2]filter g which are related to the bases of $\mathcal{V}_j$ and $\mathcal{W}_j$ respectively. From the requirement that the wavelets span the "difference" or orthogonal complement spaces, and the orthogonality of integer translates of the wavelet (scaling) function, it can be shown that the wavelet coefficients (modulo translations by integers of two) are required by orthogonality to be related to the scaling function coefficients (QMF) by

$$g(k) = (-1)^k h(1-k). \tag{2.32}$$

Further they employ the mirror filters $\tilde{h}$ and $\tilde{g}$ defined by

$$\tilde{h}(k) = h(-k) \text{ and } \tilde{g}(k) = g(-k).$$

The discrete normalized scaling $\phi_{j,k}$ and wavelet $\psi_{j,k}$ basis functions are defined as,

$$\begin{aligned} \phi_{j,k}(t) &= 2^{-j/2} h_j(2^{-j}t - k) \\ \psi_{j,k}(t) &= 2^{-j/2} g_j(2^{-j}t - k) \end{aligned} \tag{2.33}$$

where j and k are respectively the dilation and translation parameters and h_j and g_j are respectively the sequence of low pass and bandpass filters of increasing width

[2]Strictly speaking, g is a bandpass filter, but since it captures all the remaining high frequencies of the signal, it is commonly called a high pass filter.

indexed by j, which are expanded by inserting an appropriate number of zeros between filter taps (filter coefficients). They satisfy the quadrature mirror filter condition and are called the analysis (synthesis filters). The full discrete wavelet expansion of a signal $f(t) \in l_2$ (l_2 is the space of square summable functions) is given by (2.31).

The L_J's and D_j's can be interpreted in terms of simple filtering and downsampling operations.

$$\begin{cases} L_J(k) &= 2^{-J/2}[h_J^T * f]_{\downarrow 2^J(k)} \\ D_j(k) &= 2^{-j/2}[g_j^T * f]_{\downarrow 2^j(k)} \text{ for } j = 1, \cdots, J \end{cases} \tag{2.34}$$

where h^T denotes the transpose of h ($i.e., h^T(k) = h(-k)$) and $[.] \downarrow m$ is the downsampling by factor m.

The decomposition scheme is illustrated in Fig. 2.7. This scheme takes the original samples, filters and subsamples them, which produces a low resolution and detail signal on a lower scale. Mathematically speaking this subsampling stems from the factor 2^{-j} in the basis functions $\phi_{j,k}$, but it is intuitively clear that when a signal is transformed to a lower resolution scale, fewer samples are required to represent it. Note that, due to this subsampling, there are exactly as many wavelet coefficients as there are samples in the original signal $f(k)$. This representation is thus indeed non-redundant, which at the first place was the point of introducing orthogonal wavelets.

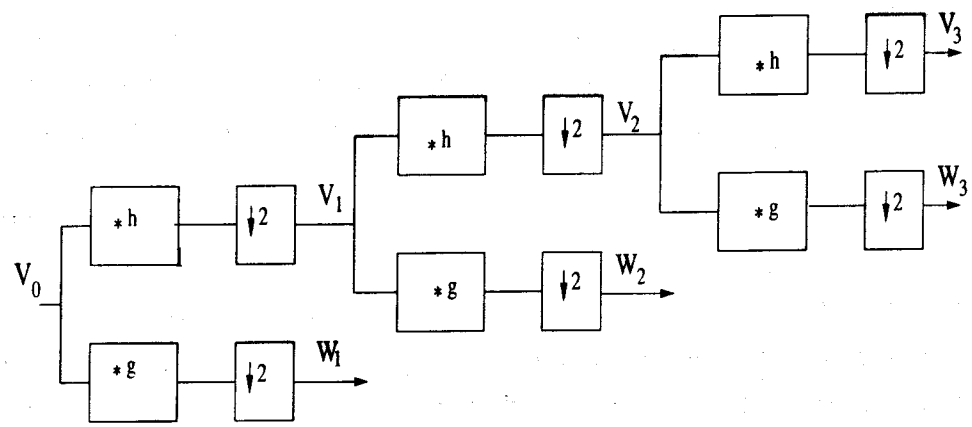


Figure 2.7: Decomposition scheme for computation of the wavelet coefficients using quadrature mirror filters (QMF). * denotes convolution and $\downarrow 2$ downsampling by a factor of 2.

The frequency response of a digital filter is given by the discrete Fourier transform of its impulse response (coefficients) $h(n)$. That is, $H(\omega) = \sum_{-\infty}^{\infty} h(n)e^{i\omega n}$. The magnitude

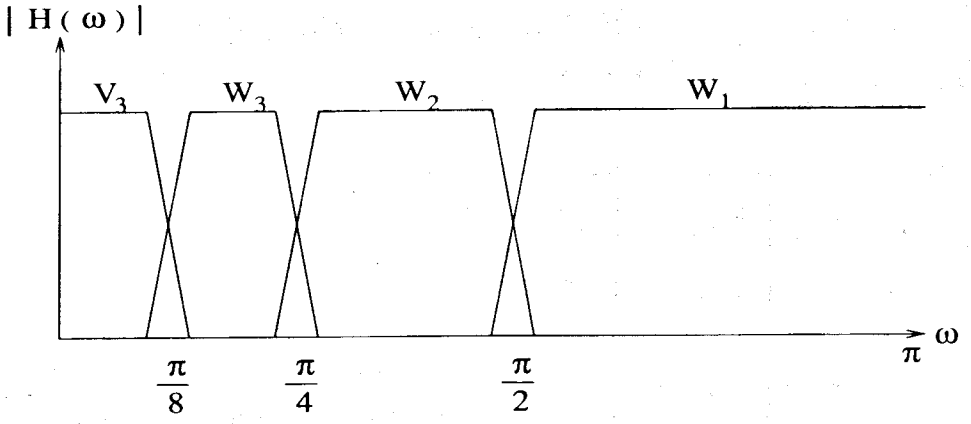


Figure 2.8: Frequency response of for the analysis tree of 2-band wavelet.

of this complex valued function gives the ratio of the output to the input of the filter for a sampled sinusoid at a frequency of ω in radians per second. The angle of $H(\omega)$ is the phase shift between the output to the input. The first stage of the two banks (Fig. 2.7) divides the spectrum of $f(k)$ into a low pass and high pass band resulting in the scaling coefficients and wavelet coefficients at the next lower resolution. The second stage then divides the low pass band into another lower low pass band and a bandpass band. The first stage divides the spectrum into two equal parts. The second stage divides the lower half into quarters and so on. This results in a logarithmic set of bandwidths as illustrated in Fig. 2.8.

2.9.2 Discrete wavelet transform in two-dimensions

The extension of this scheme to images $f(x, y)$ is usually performed by taking the tensor products of the spaces $\mathcal{V}_j$ and $\mathcal{W}_j$ and their corresponding bases. This approach has the benefit that the transform can be implemented by filtering with the one-dimensional quadrature mirror filters along the rows and columns of the image. Fig. 2.9 represents this scheme. In each filtering step we now have a *low resolution image* L_j and three *detail images* D_j^1, D_j^2 and D_j^3. Since the detail images are obtained by applying the low pass and/or high pass filters along rows and columns, they contain the vertical (D_j^1), horizontal (D_j^2) and diagonal (D_j^3) details of the original images at a certain scale j.

The computation of the 2D extension of DWT is obtained in two steps by successive application of the 1D filtering along rows and columns of an image (Fig. 2.10) (i.e., by

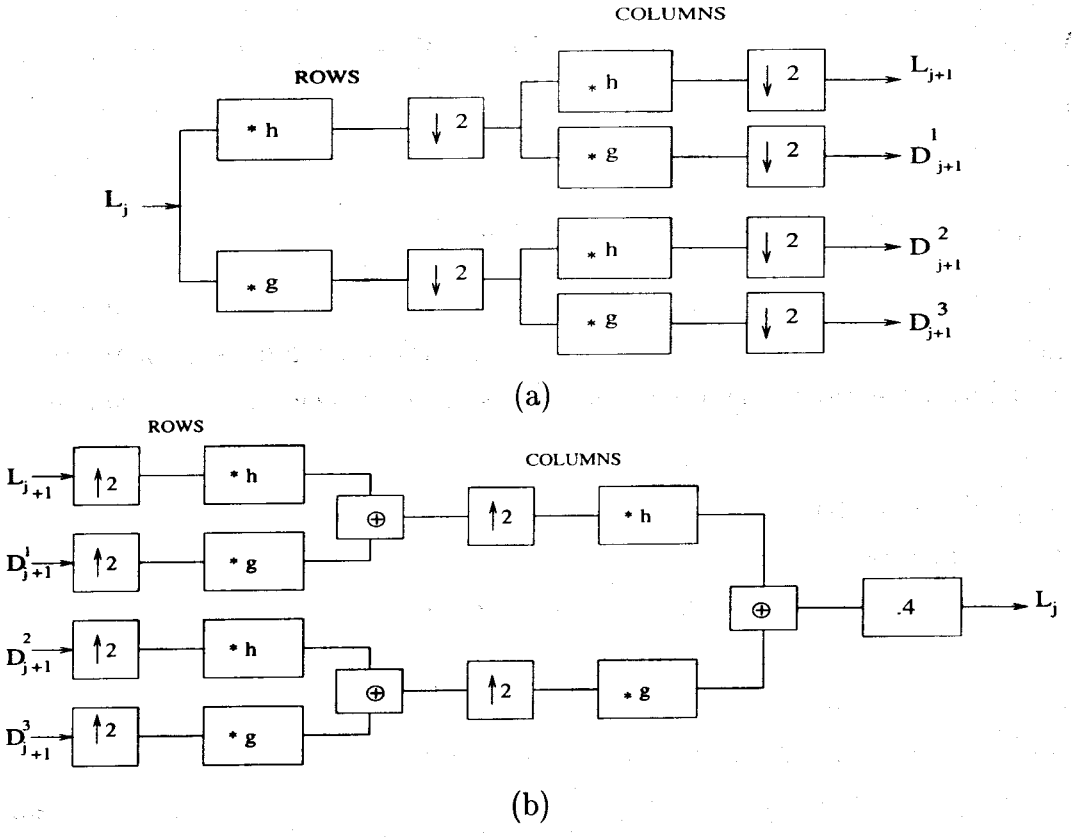


Figure 2.9: (a)Decomposition (b) reconstruction schemes for computation of the wavelet coefficients in 2 dimensions using quadrature mirror filters, * denotes convolution and $2\downarrow$ ($2\uparrow$) downsampling (upsampling) of the rows and columns by a factor of 2

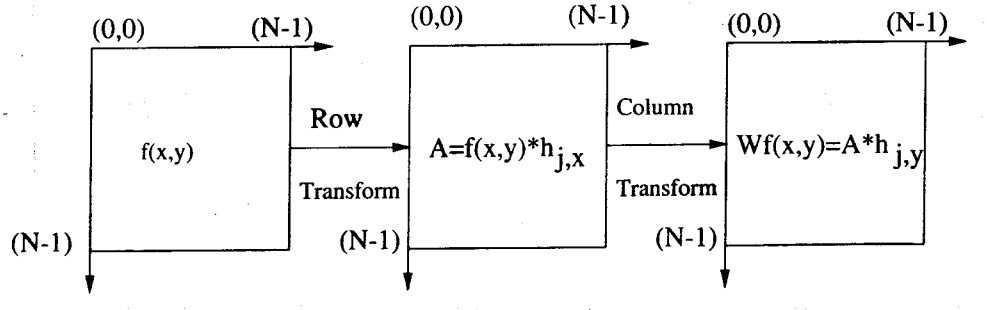


Figure 2.10: Separable filtering for 2D discrete wavelet transform.

applying a separable filter bank to the image). These are

$$L_j(x,y) = [h_{j-1,x} * [h_{j-1,y} * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(x,y), \quad (2.35)$$

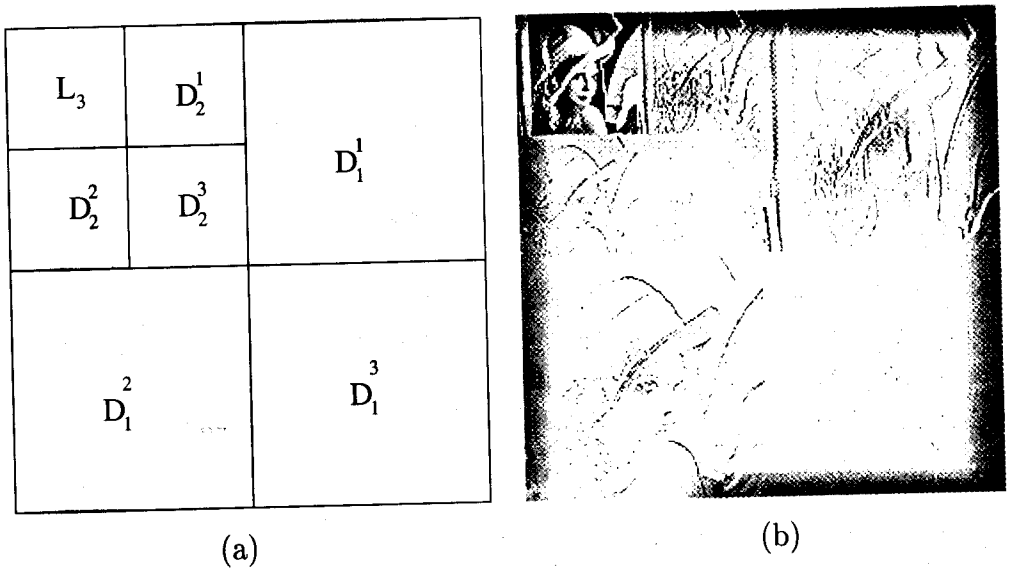
$$D_j^1(x,y) = [h_{j-1,x} * [g_{j-1,y} * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(x,y), \quad (2.36)$$

$$D_+^2(x,y) = [g_{j-1,x} * [h_{j-1,y} * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(x,y), \quad (2.37)$$

$$D_j^3(x,y) = [g_{j-1,x} * [g_{j-1,y} * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(x,y), \quad (2.38)$$

where, $*$ denotes the convolution operator, $\downarrow 2,1$ ($\downarrow 1,2$) denote subsampling along the rows (columns) and $L_0 = I(x,y)$ the original 2D signal. $h_{j,x}$ ($g_{j,x}$) and $h_{j,y}$ ($g_{j,y}$) are the low pass (bandpass) filtering along x and y directions respectively for different scale.

$L_j(x,y)$ corresponds to the lower frequencies. D_j^n are obtained by bandpass filtering in a specific direction and thus contain the detail information at scale j. $D_j^1(x,y)$ corresponds to the vertical high frequencies (horizontal edges), $D_j^2(x,y)$ the horizontal high frequencies (vertical edges) and $D_j^3(x,y)$ the high frequencies in both direction (the corners). $I(x,y)$ is represented at several scales by, $\{L_J, D_j^n \mid n = 1,2,3,\ j = 1,\ldots,J\}$. Figure 2.11 depicts the typical organization of these images and an example of a wavelet transformed image.



(a) (b)

Figure 2.11: (a) Typical organization of the detail images within the wavelet transform (b) Example of a wavelet transform of the Lenna image depth (2).

2.9.3 Wavelet packets

We can extend our discussion in defining wavelet packets as a generalization of orthonormal and compactly supported wavelets [49]. The classical wavelet system described above results in logarithmic frequency resolution. The low frequencies have narrow bandwidths and high frequencies have wide bandwidths. To overcome this limitation, the *wavelet packet* system [47] is used, which allows a finer and adjustable resolution at high frequencies. Wavelet packets may be described by the collection of functions $\{WP_i(t) \mid i \in Z^+\}$ such that

$$2^{-(j+1)/2}WP_{2n}(2^{-(j+1)}t - k) = \sum_m h(m - 2k)2^{-j/2}WP_n(2^{-j}t - m) \tag{2.39}$$

and

$$2^{-(j+1)/2}WP_{2n+1}(2^{-(j+1)}t - k) = \sum_m g(m - 2k)2^{-j/2}WP_n(2^{-j}t - m), \tag{2.40}$$

where Z^+ is the set of positive integers. Here $WP_0(t) = \phi(t)$, and $WP_1(t) = \psi(t)$. An approximation of the function $f(t)$ using wavelet packet WP_n at scale j is

$$f(t) = \sum_k WP_{n,j,k}2^{-p/2}WP_n(2^{-j}t - k). \tag{2.41}$$

Here $WP_{n,j,k}$'s, are the wavelet packet coefficients, given by [47].

Wavelet packet coefficients may be efficiently computed in terms of coefficients at coarser scales as

$$WP_{2n,j,k} = \sum_m h(m - 2k)WP_{n,j-1,m} \tag{2.42}$$

and

$$WP_{2n+1,j,k} = \sum_m g(m - 2k)WP_{n,j-1,m}. \tag{2.43}$$

Note that for standard wavelet decomposition, only two wavelet packets WP_0 and WP_1 are used. In this case the index n is restricted to $n = 0$, and only WP_0^j are decomposed from (2.42) and (2.43).

From the subband filtering point of view, the difference between discrete wavelet packet transform (DWPT) and standard wavelet transform (DWT) is that the former, unlike the latter, recursively decomposes the high frequency components as well, thereby resulting in a tree structured multi-band extension of the wavelet transform. Fig.

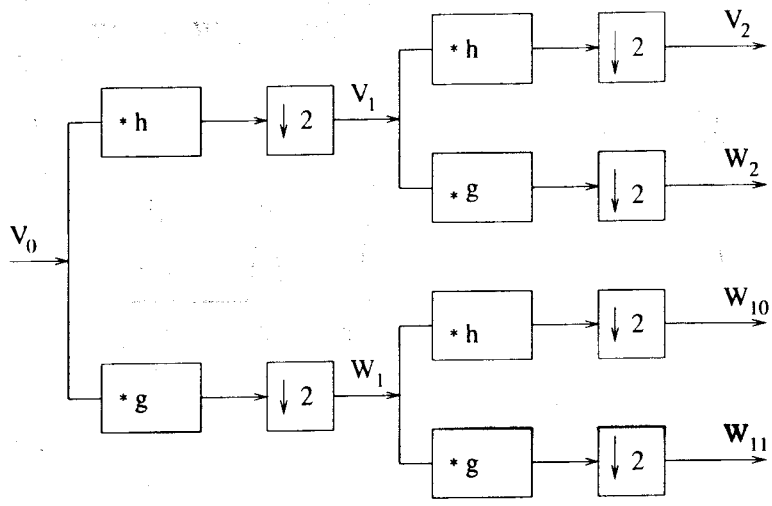


Figure 2.12: Two scale wavelet packet transform.

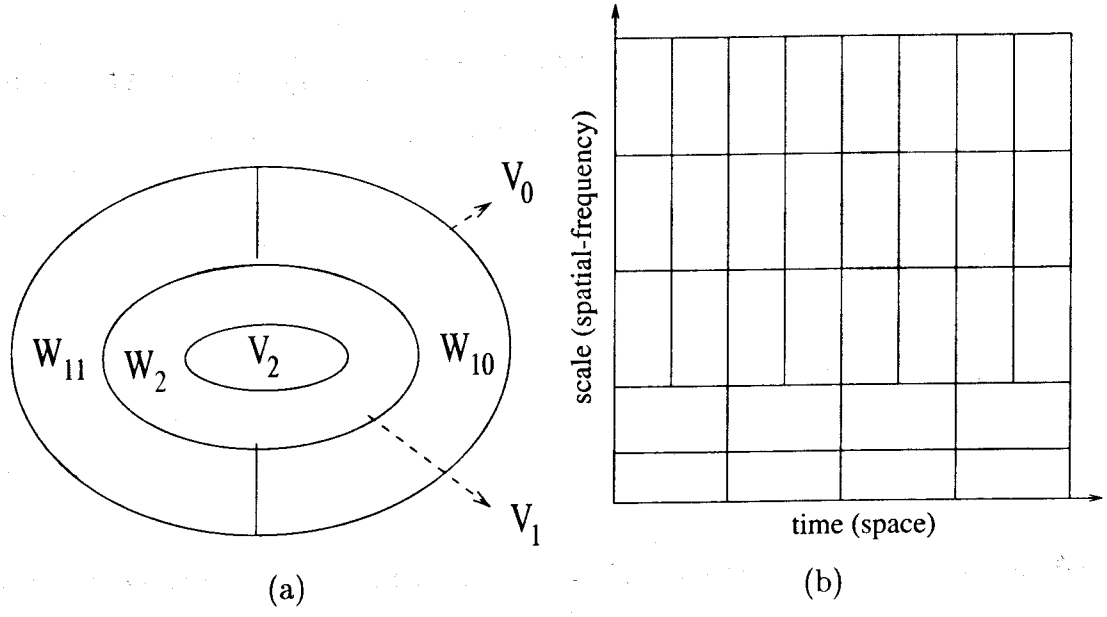


Figure 2.13: (a) Vector spaces of the scaling and wavelet functions (b) basis tiling of a wavelet packet tree.

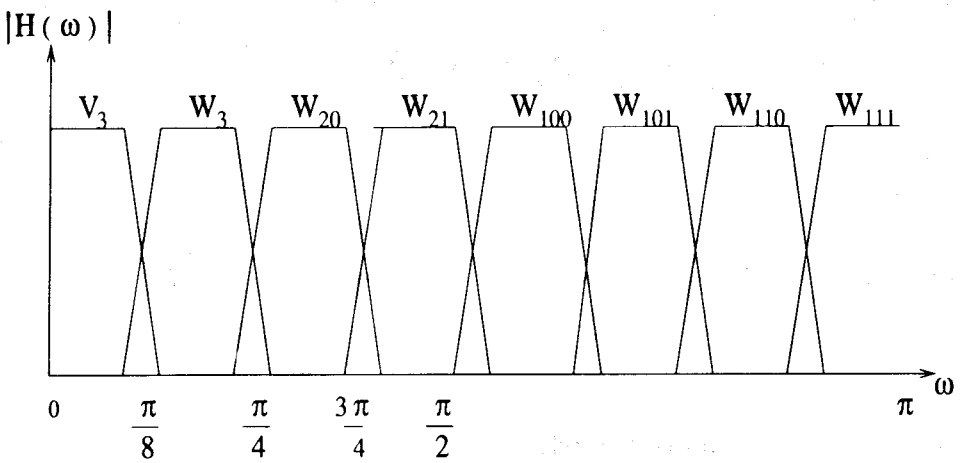


Figure 2.14: Frequency response of the packet filter bank.

2.12 illustrates the filter bank structure of the full tree for a two scale wavelet packet transform. Fig. 2.13(a) pictorially shows the signal vector space decomposition for the scaling and wavelet functions, and the frequency tiling of a three scale wavelet packet transform is shown in Fig. 2.13(b). Fig. 2.14 illustrates the frequency response of the packet filter bank.

2.9.4 M-band wavelets

M-band wavelets are a direct generalization of the conventional 2-band wavelets [150] [138] [49], where, M can have any value larger than 2. Part of the motivation for a larger M comes from a desire to have a more flexible tilting of the time-scale plane than that resulting from the standard M=2 band wavelet. It also comes from the desire to have some regions of uniform band widths rather than the logarithmic spacing of the frequency responses as illustrated in Fig. 2.8. An M-band wavelet system form a tight frame for the set of square integrable functions defined over the set of real numbers $L^2(R)$ [28].

Let $\phi(t)$ be a more general multiresolution formulation of the scaling function satisfying,

$$\phi(t) = \sum_k h(k)\sqrt{M}\phi(Mt - k), \tag{2.44}$$

where the sequence $h(k)$ is the scaling filter [28] and satisfies the following linear and

quadratic constraints,

$$\sum_k h(k) = \sqrt{M} \tag{2.45}$$

and

$$\sum_k h(k)h(k+Ml) = \delta l. \tag{2.46}$$

Additionally there are $M-1$ wavelets associated with the scaling function and satisfy

$$\psi^{(i)}(t) = \sum_k \sqrt{M} g^{(i)}(k)\psi(Mt-k) \quad \text{for } i = 1, \ldots, M-1. \tag{2.47}$$

$g^{(i)}(k)$ vectors called the wavelet filters satisfy the equation,

$$\sum_k g^{(i)}(k)g^{(i1)}(k+Ml) = \delta(l)\delta(i-i1). \tag{2.48}$$

In the discrete form these equations can also be written as

$$\phi_{jk}(t) = \sum_k M^{-j/2}\phi(M^{-j}t-k) \tag{2.49}$$

and

$$\psi_{jk}^{(i)}(t) = \sum_k M^{-j/2}\psi^{(i)}(M^{-j}t-k), i = 1, \ldots, M-1. \tag{2.50}$$

For any function $f(t) \in L^2(R)$, it can be shown that

$$f(t) = \sum_{i=1}^{i=M-1} \sum_{j\in Z} \sum_{k\in Z} \langle f(t), \psi_{j,k}^{(i)}(t)\rangle \psi_{j,k}^{(i)}(t). \tag{2.51}$$

where Z represents the set of integers and $\langle , \rangle$ is the inner product operator.

Multiresolution analysis

The scaling function and the $M-1$ wavelet functions also define a multiresolution analysis [28]. A multiresolution analysis is a sequence of approximation spaces for $L^2(R)$.

The subspaces spanned by the functions $\phi_{j,k}(t)$ and $\psi_{jk}^{(i)}(t)$ be respectively defined as

$$\mathcal{V}_j = \overline{span\phi_{j,k}; \forall k \in Z} \tag{2.52}$$

and

$$\mathcal{W}_{i,j} = \overline{span\psi_{j,k}^{(i)}; \forall k \in Z} \tag{2.53}$$

It can be shown [28] that

$$\mathcal{W}_{1,j} = \bigoplus_{i=1}^{M-1} \mathcal{W}_{i,j+1} \tag{2.54}$$

and

$$\lim_{j\to\infty} \mathcal{W}_{i,j} = L^2(R) \tag{2.55}$$

Thus $\mathcal{W}_{1,j}$ spaces form a multiresolution space for $L^2(R)$.

It follows from equation (2.44) that the $\mathcal{V}_j$ subspaces have a nested property. If the scaling and the wavelet functions satisfy the orthonormality condition, then the subspaces $\{\mathcal{W}_j^{(i)}\}$ form an orthogonal decomposition of the $L^2(R)$ function space and are related to the $\mathcal{V}_j$ nested subspaces by

$$\mathcal{V}_j = \mathcal{V}_{j+1} \oplus [\bigoplus_{i=1}^{M-1} \mathcal{W}_{i,j+1}]. \tag{2.56}$$

$f(t)$ can also be expressed in terms of the sum of projections onto subspaces $\mathcal{V}_j$ and $\mathcal{W}_{i,j}$. A function $f(t) \in L^2(R)$ can be constructed from a discrete sequence $a(k) \in l^2(R)$ in the form,

$$f(t) = \sum_k a(k)\phi(t-k) \tag{2.57}$$

i.e.,

$$f(t) = \sum_k L(k)\phi_{j,k}(t) + \sum_{i=1}^{M-1}\sum_k D_j(k)\psi_{j,k}^{(i)}(t). \tag{2.58}$$

The expansion coefficients can be expressed as

$$\begin{aligned} L(k) &= \langle f, \phi_{j,k}\rangle \\ D_j(k) &= \langle f, \psi_{j,k}^{(i)}\rangle \quad i = 1, \ldots M-1 \end{aligned} \tag{2.59}$$

Using (2.44) and (2.47) in (2.59), it can be shown that

$$L(k) = \frac{1}{\sqrt{M}} \sum_l a(l)h(Mk-l) \tag{2.60}$$

$$D^{(i)}(k) = \sum_l a(l)g^{(i)}(Mk-l) \tag{2.61}$$

which is equivalent to processing the sequence $a(k)$ with a set of linear time-invariant filters of impulse responses $p_i = \frac{1}{\sqrt{M}} g^{(i)}(k)$ and downsampling filter outputs by M. Fig. 2.15 shows a typical M-band filter bank.

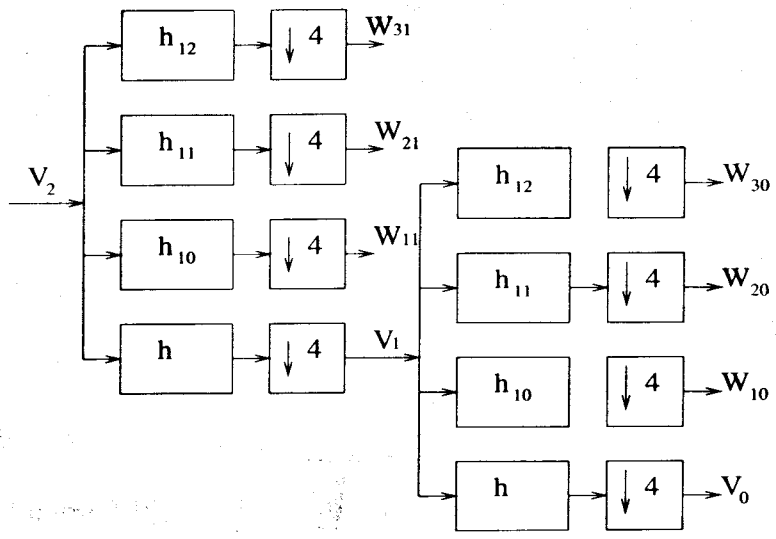


Figure 2.15: M-band filter bank structure (M=4).

An important aspect of M-band wavelets is that, a given scaling filter h specifies a unique $\phi(t)$ and consequently a unique multiresolution analysis.

For example with $M = 4$,

$$\mathcal{V}_0 = \mathcal{V}_1 \oplus \mathcal{W}_{11} \oplus \mathcal{W}_{21} \oplus \mathcal{W}_{31}, \ \mathcal{V}_1 = \mathcal{V}_2 \oplus \mathcal{W}_{12} \oplus \mathcal{W}_{22} \oplus \mathcal{W}_{32}$$

where $\mathcal{V}_j$'s are the spaces spanned by the scaling function at different resolution and $\mathcal{W}_j$'s are the spaces spanned by the wavelet functions. It is pictorially illustrated in Fig. 2.17(a). The scale-space tiling for M-band wavelet ($M = 4$) is depicted in Fig. 2.17(b). The figure clearly shows that while the frequency tiling in the standard wavelet decomposition are logarithmic (octave band) (refer Fig. 2.3(b)), the M-band decompositions give a mixture of logarithmic as well as linear frequency tiling. The frequency responses of the 4-band filter bank are given in Fig. 2.16.

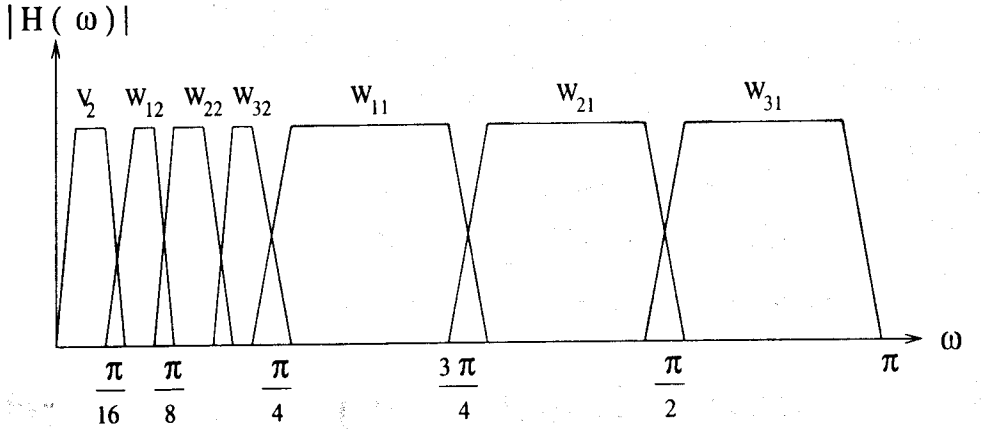


Figure 2.16: Frequency response of the 4-band filter bank.

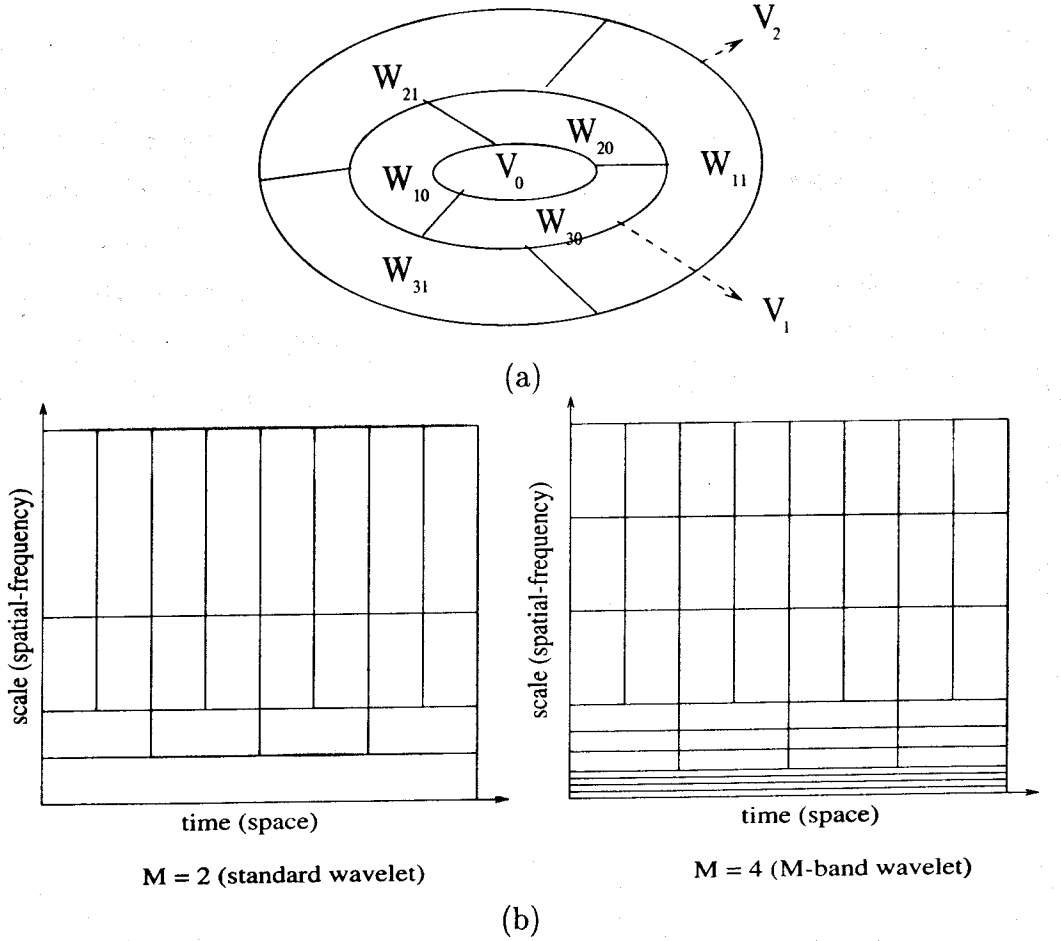


Figure 2.17: (a) Nested vector spaces spanned by the scaling and wavelet functions (b) basis tiling in $M(=2)$-band and $M(=4)$-band wavelets.

Chapter 3

Robust Texture Classification using Wavelet Frames

3.1 Introduction

Texture is one of the basic characteristics of a visible surface. Texture analysis plays an important role in human vision, computer vision and pattern recognition and is widely applied to many areas (discussed earlier in Section 1.2.2 of Chapter 1).

The majority of the existing works on texture analysis ideally assume that all images are acquired from the same line of view. A good texture analysis approach should ideally be invariant to the orientation of the texture. In many texture analysis problems the directional parameters of the image acquisition sensor relative to an image, is to be recorded carefully in order to have correct features for comparison. This means, either a detailed directional flow of the texture is required (*e.g.*, edge detection), or we require a feature extraction technique, which is independent of orientation of the texture. Therefore, extraction of rotation-invariant texture features is an important task. Due to different imaging techniques an image may have a translational shift. Texture has a shift invariant property and so, features should be shift invariant as well. Real images are often degraded by some random variations of intensity due to inherent imperfection of sensor and communication channel, this degradation is usually called noise. Noise can occur during image capture, transmission or processing. A good feature extraction process should also be tolerant to this type of aberration.

Kashyap *et al.* [91] first realized the importance of rotation-invariant texture feature extraction and have developed a texture classification scheme based on autoregressive (AR) model. But the difficulty with this model is that, this type of AR models are invariant to rotations which are multiples of 45^0 only. Cohen *et al.* [46] have modeled textures as Gaussian Markov random fields (GMRF) and have made use of the maximum likelihood (ML) method to estimate coefficients and rotation angles. The problem with this method is that the likelihood function is highly nonlinear and local minima may exist. Moreover, this algorithm is computationally very expensive. Kovacevic *et al.* have addressed the problem by the implementation of non-separable filter banks using non-separable subsampling lattices in the decomposition scheme (quincunx lattice) [94]. Even then it is very difficult to get sufficient angular localization by this method.

The orientation of texture can be used in itself as a texture classification attributes, or used for grading or detecting oriented texture. Chaudhuri *et al.* [33] have de-

fined a technique based on Hough transform for calculating texture orientation. Chen and Kundu [38] have used multichannel subband decomposition and a hidden Markov model (HMM) to solve this problem. The images are decomposed into subbands using quadrature mirror filter (QMF) bank, and then features of these subband images are modeled as an HMM. Texture samples with different orientations are treated to be in the same class. But for an image the signal components in each subband will be different for different orientations. So variability in texture feature vectors increases with increase in the number of textures to be analyzed. The approach of Greenspan *et al.* [72] involves employment of 4 angular filters and interpolation of their responses to obtain rotation-invariant features.

Wu *et al.* [171] have tried to solve this problem by converting the 2D textured images into 1D signals by spiral re-sampling, then QMF banks have been used to decompose the signal into subbands. For each band high order autocorrelation functions are taken as features. These features are then modeled as an HMM. That is, in this case rotational invariance is simulated as translation-invariance. But this method too is computationally expensive. Wouer *et al.* [166] have made use of continuous wavelet transform (CWT) for the purpose of rotation-invariant texture segmentation. CWT has the ability to distribute directional information in a continuous way. Haley and Manjunath [77] have described a method of rotation invariant texture discrimination based on Gabor filters. They have developed a 2D Gabor wavelet polar representation and a multiresolution family of these wavelets is used to compute micro-features characterizing the directional behavior of a texture. In addition, macro-features are derived from the estimated selected parameters of the micro-model. A rotation invariant set of macro-features is used for classification. However, a large combination of parameters makes texture discrimination using Gabor filters computationally expensive.

More recently Fountain *et al.* [63] have worked on rotation invariant texture features by taking the Fourier transform (FT) of the gradient direction histograms of the textures. The direction histograms being a periodic function of 2π, a rotation of the image is reflected as a translation of this histogram function. Therefore magnitudes of the FT of this function provide rotation invariant features. Ojala *et al.* [119] have proposed a set of rotation-invariant texture features based on local pixel neighborhood and are derived from the measure of covariance and symmetric autocorrelation. A rotation invariant signature of 2D contours based on the fractal theory is proposed by Tang

et al. [158]. The rotation-invariant signature is a measure of the fractal dimensions, which is rotation-invariant based on a series of central projection transform (CPT) groups. This approach is applicable to a wide range of areas such as image analysis, pattern recognition, etc.

Although a wide variety of approaches have been suggested and implemented by several researchers, we are of the opinion that, a more general way to cope with this problem is to incorporate rotational information in the texture features themselves. This is exactly what we try to implement in this work.

This chapter presents the proposed methodology for extraction of rotation and translation invariant as well as noise tolerant features for texture identification. Briefly stating, let there be a finite number of texture classes K. We assume that the image to be recognized has arbitrary orientation and is subjected to translational shifts and corrupted by noise. A number of training images belonging to each class is available. Based on the information extracted from these images, a given test image of unknown class can be classified to one of the K classes.

To achieve this we characterize textures at multiple scales using wavelet frames [1, 2]. Non-separable wavelet transform which gives an overcomplete/redundant wavelet decomposition is employed. Theoretical arguments are given to explain why they are preferred over separable transforms. An obvious advantage of using non-separable transforms is a more flexible filter design, which is exploited to build rotation as well as translation invariant features using circularly symmetric filters. Here, a texture is characterized by a set of values of variances of the frequency bands in which it is decomposed. These variances serve to provide the texture features, which are then used to perform classification (supervised) of the textured images. In an extensive experimental study, the performance in respect of translational and rotational invariance and immunity to noise is demonstrated for texture classification. Classification performance is compared with wavelet packet features given in [98]. The classification performance using our methodology is reported to be very promising.

In the next section a brief overview of wavelet frames is given. Section 3.3 discusses about the motivation to use non-separable frames and describes the wavelet parameter computation which gives the features for texture discrimination. Section 3.5 presents results of our investigation.

3.2 Wavelet Frames

Wavelet frame leads to an overcomplete decomposition of the signal

$$\begin{cases} D_{j+1}^{DWF}(k) &= \langle g_j(t-k)f(t)\rangle_{l_2} \\ L_{J+1}^{DWF}(k) &= \langle h_J(t-k)f(t)\rangle_{l_2} \end{cases} \tag{3.1}$$

The frame is a spanning set, that requires finite limits on an inequality bound of inner products. If we want the coefficient in an expansion of a signal to represent the signal well, these coefficients should have certain properties, that are stated best in terms of energy and energy bounds.

The family of sequences $F = \{g_0(t-k), \cdots, g_J(t-k), h_J(t-k)\}$ constitutes a frame of the Hilbert space l_2, if there exists two constants $A > 0$ and $B < \infty$ such that,

$$\begin{aligned} A\|f\|_{l_2}^2 \;\leq\; & \textstyle\sum_{k\,\in\, Z} \langle f(t), h_J(t-k)\rangle^2 \\ + & \textstyle\sum_{j=0}^{J} \sum_{k\,\in\, Z} \langle f(t), g_j(t-k)\rangle^2 \leq B\|f\|_{l_2}^2 \end{aligned} \tag{3.2}$$

With the help of Parseval's theorem it is easy to show that the energy conservation property is preserved

$$\|f\|_{l_2}^2 = \|L_{J+1}\|_{l_2}^2 + \sum_{j=0}^{J} \|D_{j+1}\|_{l_2}^2 \tag{3.3}$$

By definition, $L_{j+1}(k) = \langle f(t), h_j(t-k)\rangle$ and $D_{j+1}(k) = \langle f(t), g_j(t-k)\rangle$, where $\langle .,. \rangle$ is the corresponding inner product. The fundamental difference from an orthogonal system is that the representation may be redundant. This property together with the definition of wavelet coefficients in (2.15), leads to the simple reconstruction formula

$$f(t) = \sum_{k\,\in\, Z} L_{J+1}(k)h_J(t-k) + \sum_{j=0}^{J} \sum_{k\,\in\, Z} D_{j+1}(k)g_j(t-k) \tag{3.4}$$

3.3 Proposed Method for Feature Extraction

In this section we describe the methodology that we propose for texture feature extraction. In this study we make use of a circularly symmetric wavelet filter and decompose the texture images in a non-separable manner. We then find out the channel variances of these filter outputs, which provide the features for classification. Here, we first give a theoretical argument why we opt for non-separable filter decomposition. Then a brief account of the choice of wavelet function used is discussed and finally the textural measure used for rotation and translation invariant features is described.

3.3.1 Why non-separable wavelets

The 2D discrete wavelet decomposition in Section 2.9.2 is non-redundant and invertible to yield perfect reconstruction of the signal (image). Non-redundancy and perfect reconstruction is of major interest in image compression (allowing compact coding with little loss of quality). Texture analysis does not impose non-redundancy and completeness as a prime issue, instead it requires the extraction methodology to adequately characterize different textures. So care is taken, such that the wavelet representation is stable and can be employed to extract rotation-invariant features, which would give an adequate spatial/frequency characterization.

Moreover, the decomposition in the 2D space is achieved by applying the 1D filters in a separable manner as explained in Section 2.9.2 of Chapter 2 (Fig. 2.10). Separable filters have strong orientation sensitivity in the horizontal and vertical directions [152]. It is thus obvious that such decomposition can not efficiently characterize directions other than 0^0 and 90^0. That is, this type of decomposition has the problem of angular selectivity. This inefficiency is more pronounced when rotated textures are dealt with. In addition to this, the orthogonal or non-redundant DWT requires that the transfer functions of the decomposing filters do not overlap and are confined to rectangular regions in the frequency domain. With this type of localization of the transfer functions it is very difficult to extract rotation and translation invariant features. What we need is smooth transfer functions of the decomposing filters, and appreciable overlapping of these in the frequency domain. This is achieved via a non-separable filtering of the wavelet function which leads to the redundant wavelet frames. These frames use a non-orthogonal basis set. We discuss in the next section the choice of the filter bank, and the computation of the wavelet parameters used as texture features.

Another important drawback of the DWT is that, a simple integer shift of the input signal will yield a completely different wavelet transform. Hence DWT is not invariant to translation. A texture feature extraction scheme has to be independent of any translational shift, for texture has translation invariance (or stationary) property. A solution to overcome this limitation is to compute the discrete wavelet transform for all possible integers shifts of the input signal [164]. This approach leads to the oversampled DWF in which the wavelet decomposition is carried out without downsampling.

What we need is a nonorthogonal basis set or frames. This facilitates the decomposition

of a signal in 2D space by applying one dimensional non-separable filters along rows and columns independently.

3.3.2 Choice of Wavelet Function

The wavelet function in 2D is defined by $\psi_a(\vec{b}) = \frac{1}{\sqrt{a}}\psi(\frac{\vec{x}-\vec{b}}{a})$. The directional information, can be incorporated in the wavelet function, by including a rotational parameter in it [50].

$$\psi_{a,\theta}(\vec{b}) = \frac{1}{a}\psi(R^{\theta}(\frac{\vec{x}-\vec{b}}{a})) \tag{3.5}$$

where, R^{θ} is the rotation operator denoted by the matrix

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Now if the wavelet is circularly symmetric *i.e.*, R^{θ} has no influence in (3.5), such a wavelet would generate rotation invariant features. Therefore the wavelet function has to satisfy two conditions one is $\hat{\psi}(0) = 0$, which implies that the wavelet function has to be a zero mean function and $\hat{\psi}(\vec{r},\theta) = \hat{\psi}(\vec{r},0)$, this ensures that the rotation operator has no effect in (3.5).

We have chosen the dyadic wavelet ψ used by Mallat in [108], which is the second derivative of a smoothing function ϕ. This choice has been made because this closely approximates the second derivative of Gaussian, which has circular symmetry and the basis functions are symmetrical, which means that there is no phase distortion, and that the spatial localization of the wavelet coefficients is well preserved. Moreover these filters are also useful in alleviating boundary effects by simple methods of mirror extension.

The filter corresponding to the second derivative of a smoothing function can be written as $G(\omega) = \omega^2 H(\omega)$, where $H(\omega)$ is the frequency response of the smoothing (low pass) filter and should have a non-zero value at $\omega = 0$. For the filter used $G(\omega)$ has an order 2 zero at $\omega = 0$, and exhibits sharper out-band attenuation and thus better frequency separation. The wavelet function used in this work is graphically shown in Fig. 3.1. The basis function of the spline wavelet that we have chosen is symmetric which means that there is no phase distortion, and spatial localization of the wavelet coefficients is well preserved.

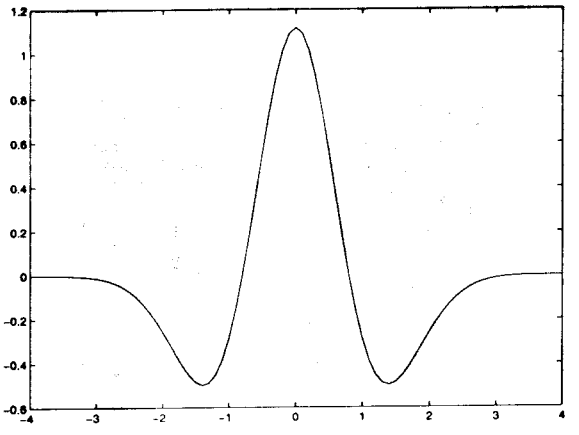


Figure 3.1: Graph of the dyadic wavelet used

3.3.3 Multiscale 2D wavelet transform

The technique used by Mallat in [109] using discrete wavelet frames has been used in the present work. Let $\phi(x, y)$ be a two-dimensional (2D) smoothing function. We call the smoothing function the impulse response of a low pass filter, whose integral over x and y is one, and which converges to zero at infinity (*i.e.*, have compact support),

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_j(x,y)dxdy = 1 \text{ and } \exists \epsilon > 0 : \phi_j(x,y) = 0, \forall \mid x \mid, \mid y \mid > \epsilon. \tag{3.6}$$

$\phi_j(x, y) = \frac{1}{j}\phi(\frac{x}{j})$, the dilation of $\phi(x, y)$ by j. Considering $\phi(x, y)$ to be double differentiable, we define two functions, $\psi^1(x, y)$ and $\psi^2(x, y)$ such that,

$$\psi^1(x,y) = \frac{\partial^2\phi(x,y)}{\partial^2 x} \text{ and } \psi^2 = \frac{\partial^2\phi(x,y)}{\partial^2 y} \tag{3.7}$$

By, definition the functions $\psi^1(x, y)$ and $\psi^2(x, y)$ can be considered to be wavelets because their integral is equal to zero

$$\int_{-\infty}^{\infty} \psi^1(x,y)dxdy = 0 \text{ and } \int_{-\infty}^{\infty} \psi^2(x,y)dxdy = 0$$

Let,

$$\psi_j^1(x,y) = \frac{1}{j^3}\psi^1(\frac{x}{j},\frac{y}{j}) \text{ and } \psi_j^2(x,y) = \frac{1}{j^3}\psi^2(\frac{x}{j},\frac{y}{j}) \tag{3.8}$$

be the dilations of the functions ψ^i by a factor j.

Let $I(x, y)$ be an image in 2D and $I(x, y) \in L^2(R)$. The wavelet transform of $I(x, y)$ at scale j has two components defined by

$$W_j^1(x,y) = I * \psi_j^1(x,y) \text{ and } W_j^2(x,y) = I * \psi_j^2(x,y) \tag{3.9}$$

Table 3.1: First six coefficients of the impulse response of the filters H and G corresponding to the wavelet in Fig. 3.1

No. of taps (n)	*h(n)*	*g(n)*
0	0.4347	0.7118
1	0.2864	-0.2309
2	0.0450	-0.1120
3	-0.0393	-0.0226
4	-0.0132	0.0062
5	0.0032	0.0039

where, j is the dilation or scale parameter. It is usually denoted by powers of 2 as, 2^j with $j = 0, \cdots, J$. This represents the traditional octave band wavelet transform of depth J. W_j^1 and W_j^2 contain the horizontal and vertical frequency information of I at scale j and are referred to as the detail images.

The computation of the 2D transform is carried out by means of a set of low pass filters $H_j = h_j(n)|_{n \in Z}$ associated with the smoothing function ϕ and bandpass filters $G_j = g_j(n)|_{n \in Z}$ associated with the wavelets ψ^1 (ψ^2) and have finite impulse responses. Table 3.1 gives the first six coefficients of the impulse response of filters $h_j(n)$ and $g_j(n)$ [108] that we have used in this study. The discrete filter responses (in the time domain) h_j and g_j are obtained by putting $2^j - 1$ number of zeroes between each coefficient of the filters $h_0(n)$ and $g_0(n)$. The transfer functions of these filters are respectively $H(2^j\omega)$ and $G(2^j\omega)$. The filtering is done in an iterative manner and without downsampling leading to an oversampled wavelet transform (Fig. 3.2).

$$\begin{aligned} L_{j+1}(x,y) &= [h_{j,x} * [h_{j,y} * L_j]](x,y) \\ W^1_{j+1}(x,y) &= [\delta_{j,x} * [g_{j,y} * L_j]](x,y) \\ W^2_{j+1}(x,y) &= [g_{j,x} * [\delta_{j,y} * L_j]](x,y) \end{aligned} \tag{3.10}$$

$L_0 = I(x,y)$ is the original image and δ is the Dirac filter whose impulse response equals 1 at 0 and 0 otherwise. The '$*$' denotes the convolution operator. $A * (B * C)$ denotes separate convolution of the rows and columns respectively of the image A with 1D filters B and C.

Then the wavelet representation of depth J of the image $I(x,y)$ consists of the low resolution images $\{L_{j+1}\}$ and detail images $\{W^i_{j+1}\}$ for $\{i = 1, 2\}$ and $\{j = 0, \cdots, J\}$.

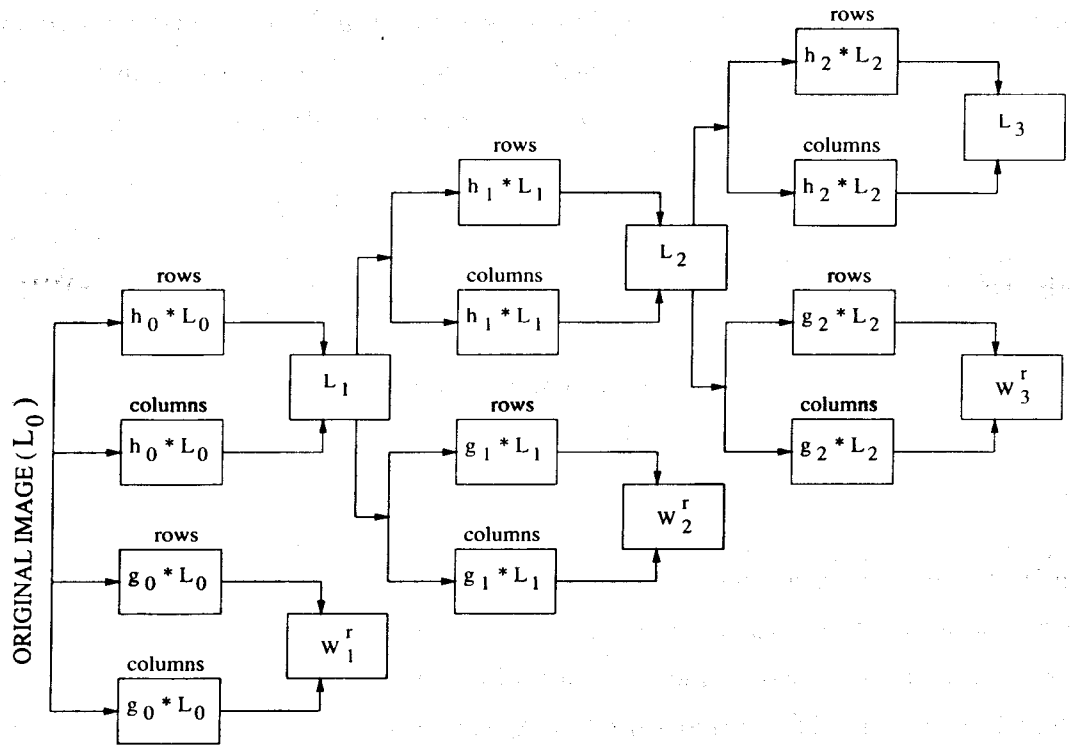


Figure 3.2: The wavelet decomposition scheme

3.3.4 Rotation invariant features

Feature extraction is a crucial step in accomplishing reliable classification. A good feature representation should be consistent among the pixels of the same class, while being as disparate as possible between classes for authentic and reliable classification. This means that it should reflect some global view while keeping some discrimination capability at the pixel level. Therefore the problem in general is one of spatial-spatial frequency analysis. We will be explaining how the rotation invariant features are being extracted from wavelet transformed images.

Substitution of (3.7) and (3.8) in (3.9) yields the following

$$\begin{aligned} \begin{pmatrix} W_j^1(x,y) \\ W_j^2(x,y) \end{pmatrix} &= j^2 \begin{pmatrix} \frac{\partial^2}{\partial^2 x}(I * \phi_j)(x,y) \\ \frac{\partial^2}{\partial^2 y}(I * \phi_j)(x,y) \end{pmatrix} \\ &= j^2 \nabla^2 (I * \phi_j)(x,y) \end{aligned} \tag{3.11}$$

where ∇^2 denotes the Laplacian. It defines edge magnitudes of the image and since it has the same property in all directions, is invariant to rotations of the image. That

is the wavelet transform of an image consists of the components which give a measure of the edge magnitudes of the image, smoothed by the dilated smoothing function ϕ_j. The edge magnitudes of the image are given by

$$W_j^r(x,y) = \sqrt{\left(W_j^1(x,y)\right)^2 + \left(W_j^2(x,y)\right)^2} \tag{3.12}$$

The $W_j^r(x,y)$ contain a measure of the edge magnitude at (x,y), which in turn gives a measure of the local gray level variation of the image. A rotation invariant multiscale representation of the image can be formulated as $h_{j+1} = \{\left(W_{j+1}^r\right)_{(j=0,\cdots,J)}\}$.

Experimental validity of the assumption

Fig. 3.3 shows some of the textured images that we have used in our experiment. We find out the histograms of the gradient values given by (3.12) of these textures at different rotations. A histogram so obtained has many local minima and maxima, which can be removed by local smoothing of the histogram. This is accomplished by local averaging of neighboring histogram elements.

We have taken three adjacent elements at a time and the process is repeated for a number of iterations. A plot of these histograms after smoothing are given in Fig. 3.4 to give a feeling that the edge magnitudes of the textures are indeed rotation invariant. We calculate the mean and the second and third order central moments of the unrotated texture edge magnitude histogram and also it's several rotated versions, to measure the similarity between these histograms, which are given in Table 3.2.

$$x_{mom}^n = E\{(x - E(x))^n\}. \tag{3.13}$$

where, n gives the order of moment. The moments so calculated suggest that the texture features are indeed rotation invariant.

3.3.5 Invariance to translation

The standard discrete wavelet transform is not shift invariant, *i.e.*, there is no simple relationship between the wavelet coefficients of the original and the shifted signal[1].

[1]Since we deal with finite length signals; here we mean circular shift.

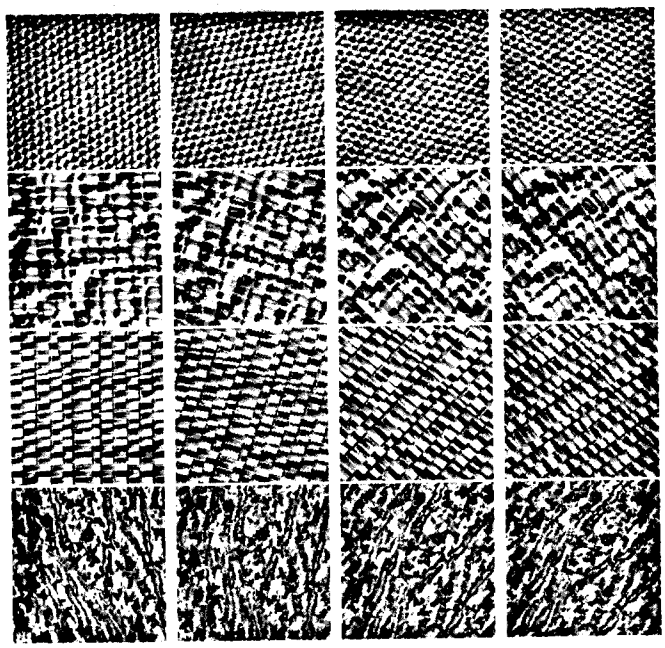


Figure 3.3: Rotated samples (0^0, 20^0, 45^0 and 55^0) of some of the textured images $D77$, $D84$, $D55$ and $D24$ used in our experiment

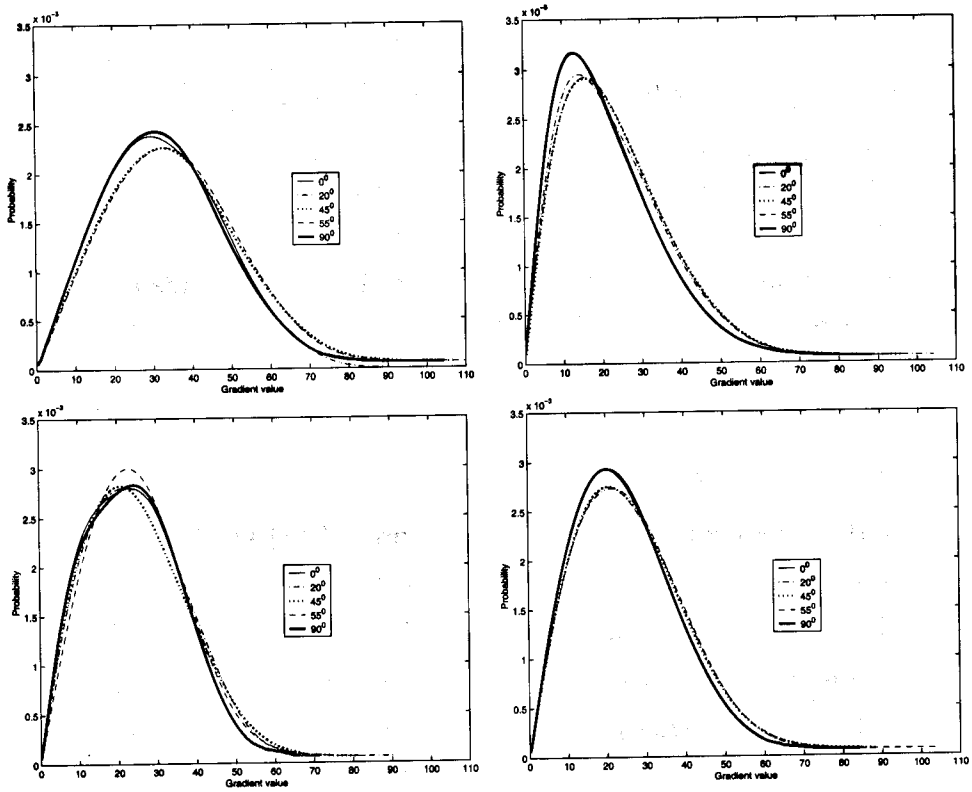


Figure 3.4: Gradient magnitude histogram plots of the rotated textures in Fig. 3.4

Table 3.2: Moments of the magnitude histograms tabulated for comparing the similarity between the unrotated texture histograms and their rotated versions Fig.3.4.

Rotn. in deg.	Texture 1			Texture 2			Texture 3			Texture 4		
	Order of moments			Order of moments			Order of moments			Order of moments		
	First	Second	Third	First	Second	Third	First	Second	Third	First	Second	Third
0	1.126	.0014	.0016	1.288	.0024	.0029	1.411	.0029	.0041	1.334	.0024	.0032
20	1.127	.0014	.0016	1.307	.0023	.0030	1.444	.0028	.0039	1.339	.0024	.0033
45	1.128	.0014	.0016	1.313	.0023	.0031	1.419	.0029	.0042	1.336	.0024	.0032
55	1.129	.0014	.0016	1.317	.0024	.0031	1.432	.0030	.0043	1.341	.0024	.0033
90	1.126	.0014	.0016	1.291	.0023	.0029	1.405	.0029	.0040	1.331	.0024	.0032

This shift invariance can be achieved by using ideas of non-decimated filter bank or a redundant discrete wavelet transform. Because the decomposition we are using is non-decimated, our wavelet system is redundant/oversampled, so it is not a basis but a frame.

Let $F = WI$ be the (orthogonal) DWT of the image I and S_R be a matrix performing a circular right shift by R, where $R \in Z$. Then

$$F_s = WI_s = WS_RI = WS_RW^{-1}F,$$

establishes the relation between the wavelet transforms of two shifted versions of a signal, I and I_s, by the orthogonal matrix WS_RW^{-1}. The first and most obvious way of computing a shift invariant discrete wavelet transform is simply computing the wavelet transform for all shifts, (*i.e.*, non-decimated wavelet transform).

3.3.6 Noise invariant features

Real images are often degraded by some random errors. This degradation is usually called noise. Noise can occur during image capture, transmission or processing and may be dependent on, or independent of, image content. Noise is usually described by its probabilistic characteristics. When an image is transmitted through some channel, noise which is usually independent of the image signal occurs. This signal independent degradation is called additive noise. While multiplicative noise results when the noise magnitude depends on the signal magnitude.

Another important aspect of our scheme is that the 2D wavelet transform that we have implemented gives a multiscale zero crossing representation of the image. So zero crossings appear at those locations where there is gray level differences corresponding to edges and also appear at non-edge locations with increasing noise levels. But our purpose is not to locate the edge points but to estimate the constituent energy of the texture by averaging over the entire image, to characterize the 2D signal. Any local distortion is thus avoided, and the features so extracted have inherent noise regularization.

3.4 Texture Classification

Texture analysis is usually carried out in two major steps. The first step is to extract meaningful features in which the image information is reduced to a small number of descriptive features. Classification of these features on the basis of a collection of pixels are called texture classification.

3.4.1 Features for classification

Since only one feature vector per image is extracted the features can be directly obtained from the subbands. In this approach we arrange the outputs of the filter banks for $J = 2$ into component subbands

$$\begin{aligned} \mathbf{f}(x,y) &= (f_{j+1}(x,y))_{j=0,\cdots,J} \\ &= [L_1(x,y),\ldots,L_{J+1}(x,y),\ W_1(x,y),\ldots,W_{J+1}(x,y)]^T \end{aligned} \tag{3.14}$$

The energies $Eng\{f_{j+1}\}$ of the decomposed image into different frequency channels at different resolutions (subbands) are given by

$$Eng_{j+1} = \frac{1}{M \times N} \sum_{x,y \in M \times N} (f_{j+1}(x,y))^2, \quad j = 0,\cdots,J \tag{3.15}$$

M and N are the number of rows and columns of the digitized image $I(x,y)$.

A more compact representation in terms of the channel variances $Var\{f_{j+1}\}$ is envisaged. In practice the channel variances are estimated from the average sum of squares

over a region *Reg* of the given texture. Each channel extracts a particular aspect of the texture at different frequencies and resolutions.

$$Var_{j+1} = \frac{1}{M \times N} \sum_{(x,y) \in M \times N} (f_{j+1}(x,y) - \overline{f}_{j+1}(x,y))^2 \tag{3.16}$$

where, $\overline{f}_{j+1}(x,y)$ is the mean value of f corresponding to each channel. We have considered the logarithm of these energies and channel variances as the features given as

$$Feat_{Eng_{j+1}} = log(Eng_{j+1}), \text{ and } Feat_{Var_{j+1}} = log(Var_{j+1}). \tag{3.17}$$

3.4.2 Discrimination using k_NN classifier

Once an appropriate set of texture features is computed, the next step is to adopt a suitable classification algorithm to assess the features discriminative power. Texture classification is a process whereby texture samples are assigned to a finite set of texture classes. We have resorted to a feature-based supervised classification method and have used a non-parametric classifier (k_NN) in our experiment. Classification of a feature vector $\vec{Feat}_{Eng_{j+1}}/\vec{Feat}_{Var_{j+1}}$ is performed by searching the k closest design vectors according to some metric $d(\vec{x},\vec{y})$. The vector $\vec{Feat}_{Eng_{j+1}}/\vec{Feat}_{Var_j}$ is assigned to that class to which the majority of these k nearest neighbors belong. More complex classification scheme such as the neural network could be considered, but these typically depend on initialization and learning length, while k_NN classifier does not. Since we emphasize on the feature extraction scheme we resort to this simple classifier. k_NN provides a robust and efficient classification scheme for evaluation of recognizing features and comparisons of feature sets. Also since this rule is derived from the Bayes rule, it is optimal in the sense that it minimizes the conditional risk.

If the distance measure d is chosen to be Euclidean, then some of the features (the ones with largest variances) tend to dominate this measure. This could be resolved by employing other distance measures (*e.g.*, Mahalanobis distance), but we have chosen to normalize the features in the following way. Features are normalized to ensure that they lie within the same dynamic range. This ensures that, the features in the non-dominant spatial frequency channels, provide important information for discriminating between different texture classes during the classification process. Each of the feature

values are normalized to lie between 0 and 1 so that they can be conveniently used for classification task.

3.5 Experimental Results

3.5.1 Data Set for Classification

We perform our classification experiments using 24 textures, some of which clearly exhibit anisotropy. All the 24 textures are shown in Fig. 3.5. These images are obtained from common sources, like Brodazt album [27] and MIT Vision Texture database [114].

The original photographs are scanned at 150 dpi and digitized into 256×256 image arrays. The images are then reduced to 128×128, which practically means that the highest octave frequency band is removed. No useful information is present in this band, since it contains mainly high frequency noise due to scanning. The mean gray value is set equal for all images, so that it is impossible to characterize the textures by their mean intensity alone.

The images that we have experimented with are given below.

- *D*77, *D*84, *D*55, *D*24, *D*3, *D*4, *D*20, *D*9, *D*12, *D*17, *D*111, *D*28, *D*51, *D*104, *D*8, *D*92, *D*6, *D*19, *D*103, *D*93 [27].
- *Fabric*.0000, *Fabric*.0006, *Fabric*.0008 and *Fabric*.0018 [114].

Several rotations like 5^0, 15^0, 20^0, 30^0, 45^0, 60^0, 75^0 and 90^0 are considered. All the test images are subjected to five different translations *viz.*, 8, 16, 32, 48 and 64 units. Five different Gaussian noise levels like 5, 15, 25, 30 and 50 *db.* are used to corrupt the images. The rotated, translated and noisy samples of some of the textures are shown in Figs. 3.3, 3.6 and 3.7.

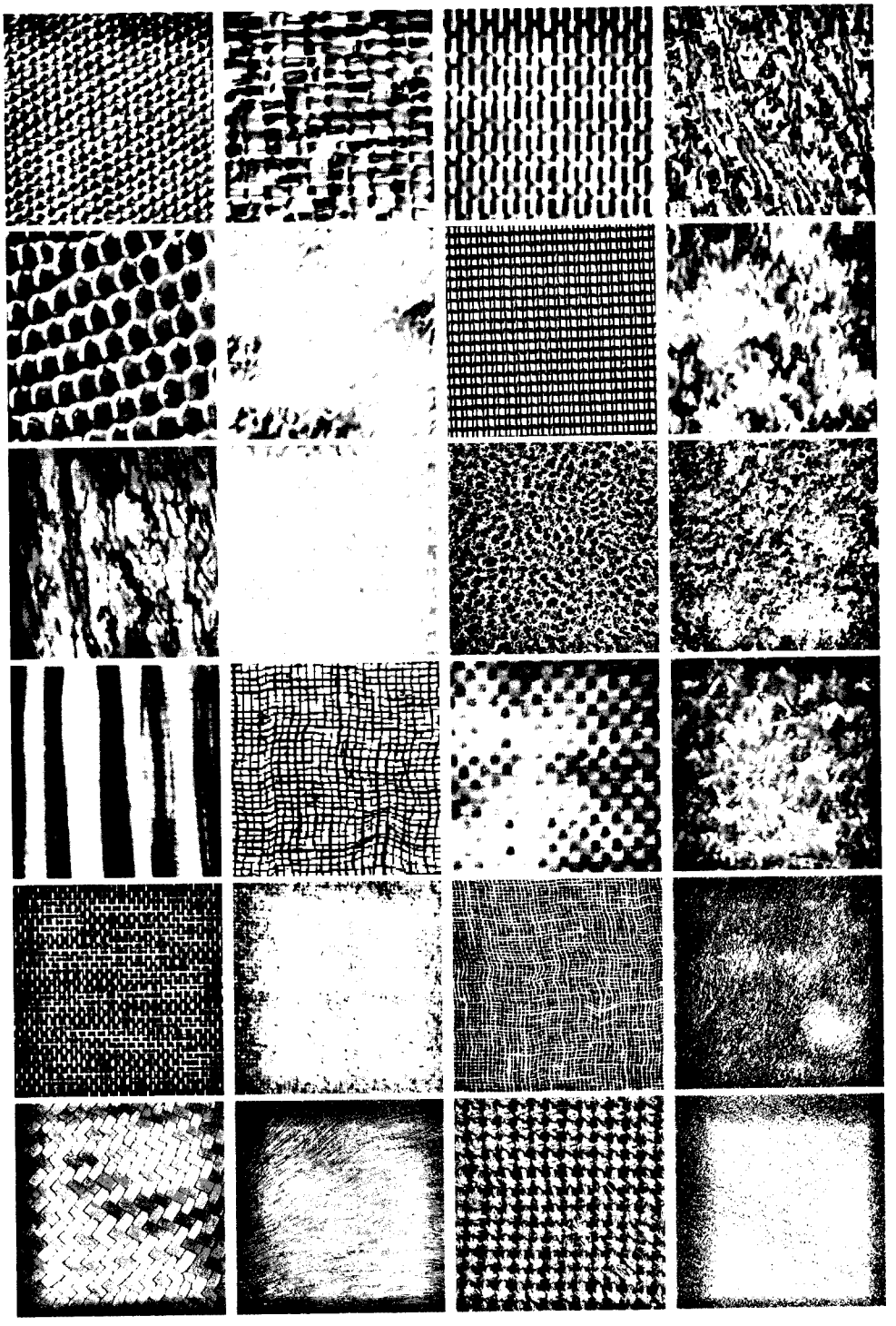


Figure 3.5: The textured images ($D77$, $D84$, $D55$, $D24$, $D3$, $D4$, $D20$, $D9$, $D12$, $D17$, $D111$, $D28$, $D51$, $D104$, $D8$, $D92$, $D6$, $D19$, $D103$, $D93$, $Fabric.0000$, $Fabric.0006$, $Fabric.0008$ and $Fabric.0018$) used in our experiment

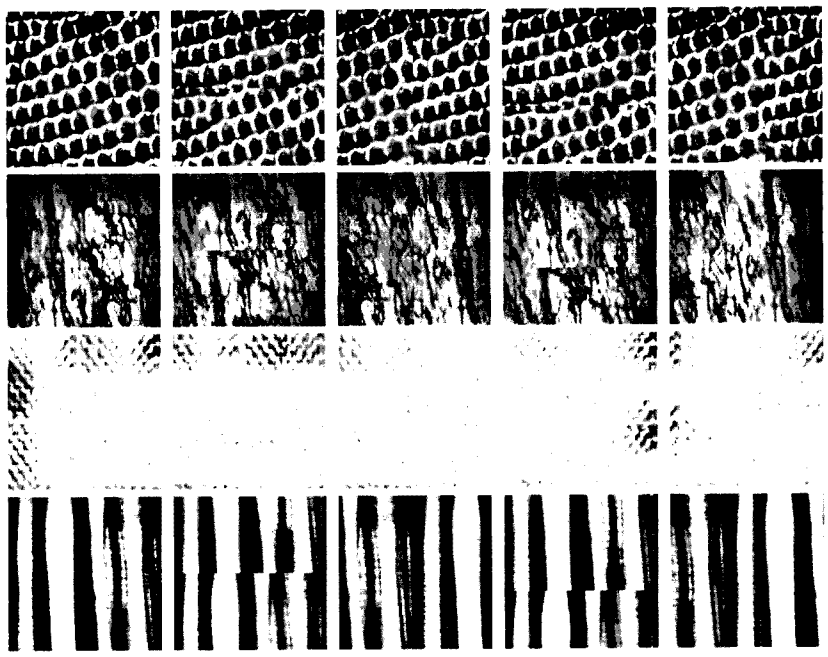


Figure 3.6: Translated samples of some of the textured images ($D3$, $D12$, $D17$ and $D51$) down-shift - 64, 48 units and right-shift - 64, 48 units

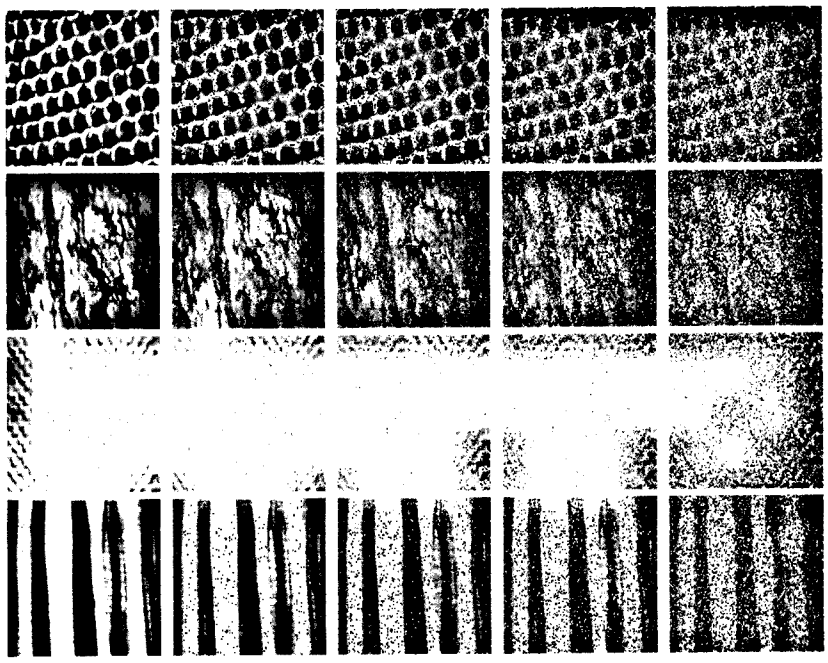


Figure 3.7: Noisy samples (15, 25, 30, 50 *db.* noise level) of some of the textured images ($D3$, $D12$, $D17$ and $D51$) used in our experiment

3.5.2 Classification Results

So far in texture classification problems, researchers have usually divided each image into overlapping (or non-overlapping) sub-regions and evaluated the feature vectors for each sub-region in each class. This is adopted to increase the number of data sets.

But here in this work we adopt a different method for extracting experimental data sets. Each texture class is subjected to five different translations in all directions (left, right, up and down), and oriented in five different rotations. Five different levels of Gaussian noise are incorporated in each class as mentioned above. There can be several combinations of these five rotations, translations and noise levels. Of which we take 35 different samples of size 128×128 for each class which are all different from each other either in translation or noise level or rotation. The percentage of correct classification is used as a performance measure.

The wavelet frame decomposition according to (3.10) is performed by processing individual images without downsampling. Each image is subjected to three levels (scales) of decomposition, and each image is represented by 6 features as shown in Fig. 3.2. The energies and variances for 5 subbands leaving out the subband corresponding to the coarsest approximation of the image (L_3) are computed. We choose the *Log of energy* and *Log of variance* as the information measures. The energy measure characterizes the luminance of the texture, while variance measure gives an idea about the contrast of it.

In the first series of experiments we compare the performance of the discrete wavelet frame (DWF) and discrete wavelet packet transform (DWPT) approaches. The DWPT is achieved by Daubechies analyzing filter of length 16 (daub16) following (2.42) and (2.43). The data sets are obtained using the five sets of parameters shown in Table 3.3. The percent of correct classification is measured by the $leave - n - out$ method. That is out of N data sets each time n distinct samples are taken as test data without repetition and the rest are used for training. We get an excellent classification result while considering the redundant DWF decomposition which is consistent with our expectation. Very poor classification is obtained with DWPT, which supports our argument that the orthogonal transforms using separable filters are not suitable for feature extraction of rotated textures. We however find wavelet packet decomposition to perform quite well as a feature extractor for unrotated textures. The observation

from Table 3.3 is that, a 45^0 rotation affects classification results. This is because some of the textures that we have used are highly anisotropic and this anisotropy is more pronounced in the diagonal direction (45^0).

Table 3.3: Classification performance considering the discrete wavelet frame transform and the discrete wavelet packet transform

Test data set			Training data set			% of classfn.for	
transl.	rotn.	noise	transl.	rotn.	noise	DWF	DWPT
8	5	15	16 8 16 8	20 30 60 45	25 15 25 15	99.99	88.88
16	20	25	8 8 16 8	5 30 60 45	15 15 25 15	99.92	73.70
8	30	15	16 16 8	20 60 45	25 25 15	99.99	83.70
16	60	25	8 8 16 8	5 30 20 45	15 15 25 15	99.99	77.00
8	45	15	8 8 16 16	5 30 60 20	15 15 25 25	95.92	77.00

In the next set of experiments we change the parameter values. We want to test the performance of the classifier in absence of rotation but subject each texture class with different levels of translation and noise. Noise as high as 50 *db.*, is incorporated corrupting 70% of the pixels and translation as large as 64 units is given. The classification result for this set of parameters is summarized in Table 3.4. It is observed that when rotation is not considered classification results are almost perfect.

The local periodicity of the texture elements can break due to translation by circular shift of the images. False energy peaks can emerge at the boundaries. Also incorporation of noise give rise to false energy peaks. The basic idea for texture characterization is that some of the textures may have low energies while others may have high energies. Textures are classified with a measure of these energies. We measure

the energy/variance of the transformed images over the entire image and so, a local distortion is suppressed due to averaging over the entire image.

The textured images are decomposed into several levels of resolution. This is accomplished by smoothing the textures and then extracting edge information from them. The smoothing operation inherently removes most of the noise present as we go to coarser resolutions. Classification error occurs for a translation of 64 units and noise level of 50 *db.* but is not remarkably high. This is because with such a high level of noise contamination some of the textures become almost visually indiscriminable.

We perform another experiment in which we use features for 0^0 rotation to design a classifier and test its performance in classifying features from rotated samples ranging from 20^0 to 90^0. The classification performances tabulated in Table 3.5 clearly provides us an idea as to how the classifier responds to the rotated samples of the textures when only unrotated ones are used for classifier design.

Table 3.4: Classification performance without considering rotation

Type of decomp	No. of features	Test data set			Training data set			% of classfn.
		transl.	rotn.	noise	transl.	rotn.	noise	
DWF	5	8	0	15	16	0	25	99.99
					32	0	30	
					64	0	50	
					48	0	15	
		16	0	25	8	0	15	99.99
					32	0	30	
					64	0	50	
					48	0	15	
		32	0	30	8	0	15	99.99
					16	0	25	
					64	0	50	
					48	0	15	
		64	0	50	8	0	15	98.89
					32	0	30	
					16	0	25	
					48	0	15	
		48	0	15	8	0	10	99.99
					32	0	30	
					64	0	50	
					16	0	25	

Another interesting study is to compare the performance of the classifier considering both the information measures used in our experiment. We find the *Log variance* to outperform the *Log energy* measure. It can be seen form Table 3.6 that if we have instances of all rotations in the training data and we employ $leave-n-out$ method for classification of rotated data we can get excellent results. The 45^0 rotation classification is comparatively poor than the others.

Table 3.5: Percent of correct classification considering rotation only and comparison of the performance of the two information measures

Training data set rotation in deg.	Test data set rotation in deg.	Information measure	Percent classfn.
0	20		97.38
0	30		93.33
0	45		85.00
0	60	Log energy	88.00
0	75		93.38
0	90		93.33
0	20		98.82
0	30		98.82
0	45	Log variance	95.29
0	60		97.64
0	75		97.64
0	90		97.64

Table 3.6: Percent of correct classification considering rotation only by $leave-n-out$ method

Test data set rotation in deg.	Training data set rotation in deg.	Percent classfn.
0	20, 30, 45, 60	98.89
20	30, 45, 60, 0	100
30	0, 20, 45, 60	100
45	0, 20, 30, 60	100
60	0, 20, 30, 45	100

3.6 Conclusion

In this chapter we have presented our studies on the rotation & translation invariant and noise tolerant texture classification. Simple and computationally efficient features have been extracted based on the fact that circularly symmetric wavelets give rotationally invariant features, as these wavelets are independent of orientation. We have seen that wavelet frame analysis is appropriate over the standard sub-sampled (critically sampled) wavelet analysis for rotational invariance. We have also verified experimentally that the features that we have extracted are really invariant to rotation by studying the gradient magnitude histograms of the several rotated versions of the same texture class. One more important aspect of our scheme is that the features extracted have inherent noise regularization.

So we can summarize our results as, for classification of rotated textures, multiresolution representation of the images is clearly preferable. As texture features are prevalent in the intermediate frequency bands, which suggests that wavelet packet signatures can classify textures very efficiently, but they perform poorly when rotation is taken into account. So an overcomplete wavelet decomposition, which leads to wavelet frames, happens to be a preferred solution. We also find that number of features used is only five, which is small, compared to the number of features used for classification purpose found in the literature. That is in essence we have achieved a large reduction in dimensionality of the feature space. In the set of textures classes that we have selected for experimentation, there are some textures which show high anisotropy. For this whole series of experiments, classification errors have usually occurred in textures that are almost difficult to discriminate even visually and also which are highly anisotropic. We can conclude that texture signatures based on multiresolution wavelet frames analysis holds great potential for accomplishing subtle discrimination and robust classification against rotation, translation and noise and is computationally simple and efficient.

To further strengthen the robustness of the extracted features, we consider the gray-scale variations, which is one of the commonest aberration caused due to imperfection in the imaging sensors. Feature extraction is a pre-process for texture segmentation. A wide variety of applications of texture segmentation exist. Keeping all these aspects in view the following chapter discusses a gray-scale transform invariant feature extraction scheme for robust texture segmentation.

Chapter 4

Rotation and Gray-scale Transform Invariant Texture Segmentation

4.1 Introduction

In this chapter we intend to discuss on another important issue, how to extract gray-scale transform invariant features for robust texture characterization besides the fact that these are also invariant to rotation and tolerant to noise tolerant as described in the earlier chapter 3. An important problem of image (texture) analysis is, gray-scale variations of an image, due to usage of various equipment and processes having different device characteristics for acquisition. So, it is desirable that features characterizing a texture should also be invariant to any gray-scale transformation of the image (device independent) and to be used for robust texture segmentation [3, 10].

To our knowledge, no significant attempt (with a few exceptions) has been made to study gray-scale transform invariant texture analysis. Chen and Kundu [38] have used a linear function with an offset to describe the phenomenon of gray-scale transform invariance. Wu *et al.* [171] have converted the 2D textured images into 1D signals by spiral re-sampling, then the signal is decomposed into subbands by QMF. High order auto-correlation functions are taken as features and a normalization operation was performed to achieve invariance to gray-scale transformation. Ojala *et al.* [119] have presented a multiresolution approach to gray-scale and rotation-invariant texture classification based on local binary patterns and non-parametric discrimination of sample and prototype distributions. The method is based on recognizing that certain local binary patterns, termed "uniform" which are fundamental properties of local image texture and their occurrence histogram considered as texture feature.

In texture segmentation, the problem is to automatically identify the different regions with homogeneous textures. The segmentation of textured images has been recognized as a difficult problem in image analysis, because one usually does not know *a priori* the number of texture classes that exist in an image, and also their region-wise distribution. Texture segmentation is generally done by extracting some meaningful features from an image. Of the several approaches available for texture feature extraction, we focus on a particular approach, which is referred to as multichannel filtering. The basic idea is to decompose textures into their constituent signal energies. Different textures should possess different energies and thus a composite image consisting of several textures can be identified by a measure of these energies (features). The features are extracted by filtering with a linear filter and estimating the local energy of the filter response.

We employ the non-separable wavelet transform in the filtering stage as discussed in Chapter 3 which gives an oversampled wavelet representation of a texture image. The filtered responses are then subjected to the local energy estimator, which estimates the energies of each of these filtered output in a local region. The features so extracted are used for texture segmentation purpose, which is invariant to rotation and gray-scale transformation. Performance of segmentation algorithm is found to be very encouraging. The segmentation performance is also evaluated under various other types of noise besides Gaussian as mentioned in the earlier chapter.

Following sections describe the wavelet parameter computation which gives the features for texture discrimination, and experimental results with critical comments about the performance.

4.2 Gray-scale Transformation Invariant Features

The gray-scale transform mentioned in Section 4.1 can be modeled as follows

$$I_t(x,y) = \gamma I(x,y) + \delta \tag{4.1}$$

where, $I(x,y)$ is the gray level at (x,y), γ is a positive scale factor, and δ is a shift factor. We assume that $0 \leq I_t(x,y) \leq I_{MAX}$, I_{MAX} is the maximum gray value of $I(x,y)$. In texture analysis we require an algorithm that is invariant to gray-scale transformation. We remove the mean of the image to nullify the effect of δ. The effect of scale factor γ is removed in formulation of the feature. The features obtained according to equation (3.12) can be modified while considering the gray-scale transformed image (4.1) as

$$Wgt_j^r(x,y) = \gamma\sqrt{\left(W_j^1(x,y)\right)^2 + \left(W_j^2(x,y)\right)^2} \tag{4.2}$$

The above equation (4.2) can be rewritten by taking logarithm on both sides as

$$log(Wgt_j^r(x,y)) = log(\gamma) + log(Wg_j(x,y)) \tag{4.3}$$

It can also be written in the following form as

$$WGT_j^r(x,y) = \Gamma + WG_j(x,y) \tag{4.4}$$

It is very apparent from equation (4.4) that the additive factor Γ can be eliminated by removing the mean from the image $WGT_j^r(x,y)$. The features are then obtained by exponentiation of $WGT_j^r(x,y)$ whose mean has been removed.

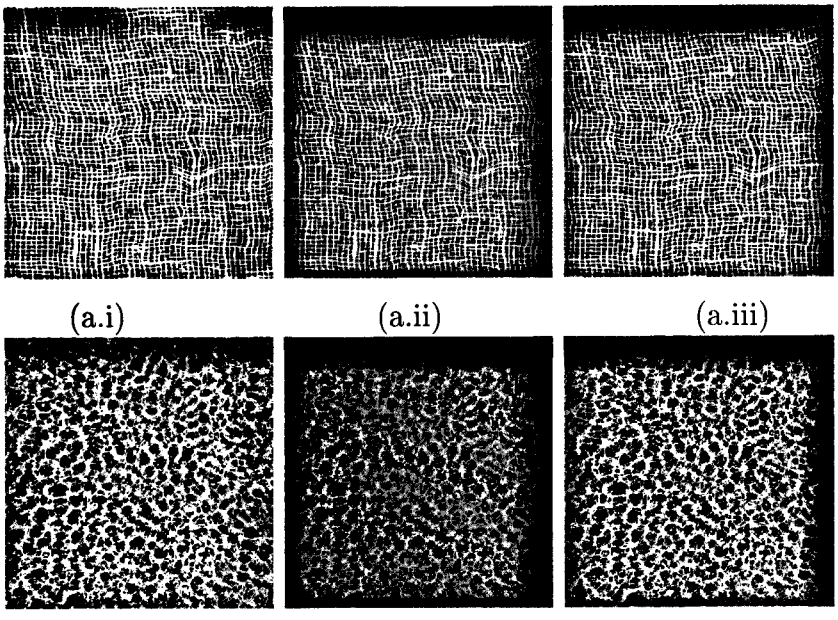


Figure 4.1: Textures without (a.i)-(b.i) $D103$ and $D111$, with gray-scale transforms, (a.ii)-(b.ii) $\gamma = 0.5, \delta = 20.0$ (a.iii)-(b.iii) $\gamma = 1.2, \delta = 0.0$

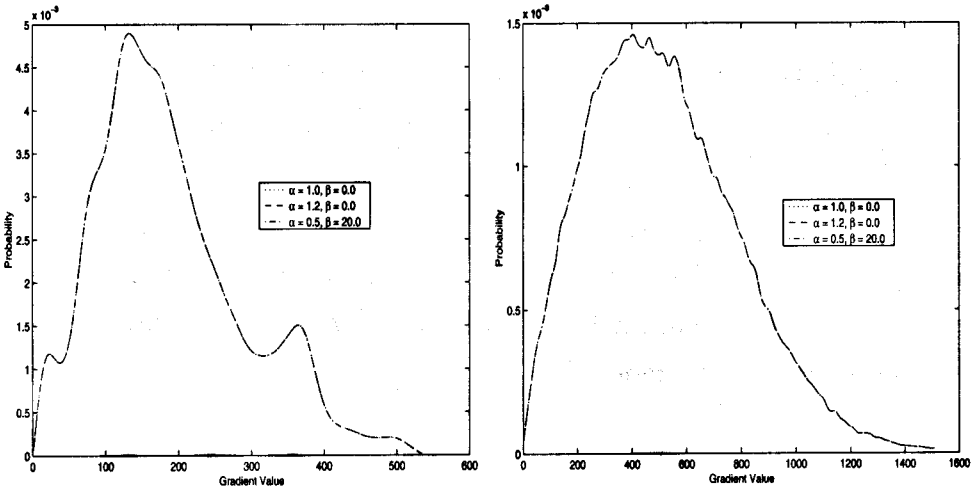


Figure 4.2: (a)-(b) Edge magnitude histogram plots

Experimental validity

Fig. 4.1 shows two textured images that we have used to generate the composite textures used in our experiment. We find out the histograms of the exponentiated values given by (4.4), of these textures for different shift and scale factors. A histogram so obtained has many local minima and maxima which can be removed by local smoothing of the histogram. This is accomplished by local averaging of neighboring histogram elements as explained in Section 3.3.4.

Plots of these histograms after smoothing are given in Fig. 4.2 to give a feeling that the features characterizing the textures are indeed invariant to any gray-scale transforms. Note that all the three plots overlap each other exactly.

4.3 Texture Segmentation

Classification obtained from individual pixels is texture segmentation. Segmentation of an image based on textural cues into different regions is a difficult task. More often, one does not have an *a priori* knowledge about the types of textures that exist, how many different textures are there and what regions contain which of the textures. We would like to emphasize on the feature extraction aspect that differ substantially from what is encountered in the case of classification. In a segmentation task features need to be computed locally, since different textures appear in one image.

4.3.1 Features for segmentation

In texture segmentation a local neighborhood is investigated. The goal is to extract some implicit local information that lets similar and dissimilar areas to be recognized. The feature extraction scheme consists of three main operating modules as is shown below (Fig. 4.3).

The first module is the filtering stage, which is followed by a nonlinear stage plus a smoothing filter (the latter two constitute the local energy estimator) and finally a clustering step. The objective of the filtering is to transform the edges between textures into detectable discontinuities (*i.e.*, to extract local frequencies of the textures). While

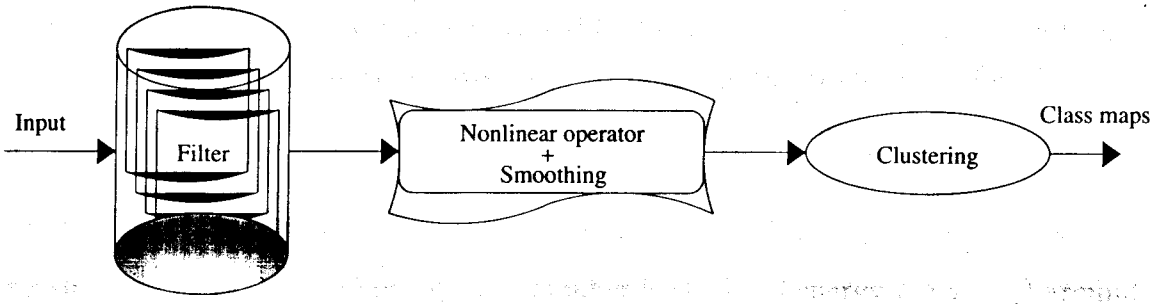


Figure 4.3: Fast iterative implementation of the algorithm used for extracting texture features

the purpose of the local energy estimator, is to estimate the energy of the filter output in a local region. The clustering step gives the final class maps.

The filtering is done as discussed in Section 4.2. The wavelet filter decomposes an image into a number of frequency subbands corresponding to different resolutions. Let these subbands be denoted by $F_b(x, y)$, where b is the index of number of subbands, and $b = 1, \ldots, B$.

4.3.2 Local energy estimation

The next step is to estimate the energy in the filter output within a local region around each pixel. The local energy estimate is utilized for the purpose of identifying areas in each channel where the bandpass frequency components are strong resulting in a high energy value and the areas where it is weak into a low energy value.

A nonlinearity is needed in order to discriminate texture pairs with identical mean brightness and second-order statistics. We have used the modulus operator as the nonlinearity. One reason for choosing this nonlinear operator is that it is parameter independent; which means it is independent of the dynamic range of the input image and also of the filter amplification.

The raw wavelet coefficients are inefficient as texture cues. They can split textured information into different frequency channels, but do not provide any local pixel information. To calculate local features of an image, we slide a fixed size window on the wavelet coefficients of an image and compute the local statistics in each individual window. These values are used as feature values of the central pixels of these windows.

A wide variety of textural measures is available. In this present study, energy measure for texture segmentation is considered, which indicates the textural uniformity, *i.e.*, pixel pairs repetitions. Energy is usually defined in terms of a squaring nonlinearity. However, in this study *average absolute deviation* (from the mean) has been used as a generalized energy definition.

For a subband image $F_b(x, y)$ of subband number b, the local energy $Eng_b(x, y)$ around the x, y^{th}pixel can be formally defined as

$$Eng_b(x, y) = \frac{1}{R} \sum_{x'=1}^{w_l} \sum_{y'=1}^{w_l} \mid (F_b(x', y') - \overline{F}_b(x, y)) \mid, \tag{4.5}$$

where w_l is the window size and $R = w_l \times w_l$. The term $\overline{F}_b(x, y)$ stands for the mean around the $(x, y)^{th}$ pixel within a neighborhood of R. Note that $F_0(x, y) = I(x, y)$, where, $I(x, y)$ is the original image of size $M \times N$.

This step is succeeded by a low pass (smoothing) filtering (using a Gaussian filter $h_G(x, y)$) to get a feature image. Formally, the feature image $Feat_b(x, y)$ corresponding to subband image $F_b(x, y)$ is given by

$$Feat_b(x, y) = \sum_{(x', y') \in G_{xy}} \Gamma(F_b(x', y') h_G(x - x', y - y'))$$

where $\Gamma(\cdot)$ gives the energy measure and G_{xy} is a $G \times G$ averaging window centered at pixel with coordinates (x, y).

Using a Gaussian weighting window as the local window w_l, results in less sparse points (*i.e.* denser feature distributions) as compared to when no weighting window is used. Using the local *average absolute deviation* values from the mean we have a good separation of different patterns in feature space, also the distribution of clusters is symmetric and similar to a Gaussian profile. Local *average absolute deviation* values from the mean values of a Gaussian window give us a robust quality of features in the feature space in all of our test images. However, one can use other local statistics depending upon the applications. Gaussian weighted windows are are also preferred over unweighted windows, because, the former are likely to result in more accurate localization of texture boundaries, since averaging blurs the boundaries between textured regions.

The above steps result in a set of feature images $Feat_b(x, y)$ from which a set of feature vectors is derived. These feature vectors, corresponding to the decomposed images

at different resolutions, are assumed to capture and characterize effectively different scales of texture of the input image. The oversampled wavelet transforms introduce redundancy in filtered images, which may be useful for a reliable, result in a recognition problem. The redundancy of information will present itself as more patterns for each cluster in feature space. In many clustering algorithms the reliability of final results will be increased by having more feature vectors per cluster.

4.3.3 Feature integration/unsupervised classifier

Having obtained the feature images, the main task is to integrate these feature images to produce a segmentation. Energy values have to be normalized to ensure that they lie within the same dynamic range. This ensures that the energy values in the non-dominant spatial frequency channels provide important information for discriminating between textures during the clustering process. The energy values are normalized to lie between 0 and 1 so that they can be conveniently used for segmentation.

We define a scale-space signature as the vector of features at different scales taken at a single pixel in an image

$$\vec{Feat}(x,y) = [Feat_1(x,y), Feat_2(x,y), \ldots, Feat_B(x,y)] \tag{4.6}$$

Suppose these scale-space signatures are considered as feature vectors in a feature space. If the signatures of one texture are distinct from the signatures of another texture a pattern recognition system can be used to identify several different textured regions in the scale-space.

Let us assume that there are K texture categories. If our texture features are capable of discriminating these categories then the patterns belonging to each category will form a cluster in the feature space which is compact and isolated from clusters corresponding to other texture categories. We emphasize on the feature extraction (representation) part in this work. So we have used a traditional *k-means* clustering algorithm. Although other sophisticated algorithms like watershed clustering could have been used.

4.4 Performance Evaluation Measures

Although we have used an unsupervised scheme for classification, that is we did not make use of any ground truth, but since we have generated the test images (using Brodazt textures), the texture boundaries corresponding to various classes are known. That means we can say that we have an *a priori* knowledge of the class labels. So we are in a position to use the following measures of classification accuracy, to evaluate the performance of our algorithm.

Percentage of correct classification: We provide the performance results in terms of percentage of correct classification, which is defined as

$$\frac{n_{ic}}{n_i} * 100 \tag{4.7}$$

where n_x is the number of pixels belonging to class x, of which n_{xc} pixels have been correctly classified. $\frac{n_{xc}}{n_x}$ is also called the Producer's accuracy [146].

In this section we provide two more statistical measures, namely User's accuracy (U) and Kappa (κ) following [48] and [142].

User's accuracy : If n'_x pixels (of all pixels n) are found to be classified into class i, then the User's accuracy (U) is defined as

$$U = \frac{n_{xc}}{n'_x} \tag{4.8}$$

where n_{xc} is as defined earlier. The User's accuracy gives a measure of the confidence that a classifier attributes to a region as belonging to a class. That is, it denotes the level of purity associated with a region.

Kappa : The coefficient of agreement called 'Kappa' measures the relationship of beyond chance agreement to expected disagreement. The estimate of kappa (κ) is the proportion of agreement after chance agreement is removed from consideration. The estimate of Kappa for class i (κ_i) is defined as,

$$\kappa_i = \frac{n * n_{xc} - n_x * n'_x}{n * n'_x - n_x * n'_x} \tag{4.9}$$

The numerator and denominator of the overall Kappa are obtained by summing the respective numerators and denominators of κ_x separately over all classes x.

We have tested our algorithm on several natural scenes, and since an *a priori* knowledge about the class labels is not available the following measure has been used to evaluate the performance of our scheme.

Quantitative index β : A quantitative index β [124] is being used to evaluate the segmentation results. β is defined as the ratio of the total variation and within class variation. Let n_x be the number of pixels in the xth ($x = 1, 2, \ldots c$) region, let Q_{xy} be the gray value of the yth pixel ($y = 1, 2, \ldots n_x$) in the xth region, and $\overline{Q_x}$ be the mean of n_x gray values in the xth region. It is formally defined as

$$\beta = \frac{\sum_{x=1}^{c} \sum_{y=1}^{n_x} (Q_{xy} - \overline{Q})^2}{\sum_{x=1}^{c} \sum_{y=1}^{n_x} (Q_{xy} - \overline{Q_x})^2} \tag{4.10}$$

where, $\overline{Q} = Q/n$, n is the size of the image and $\overline{Q}$ is the mean gray value of the image.

The numerator is constant for a given image and number of class, so the value of β is dependent only on the denominator. The denominator on the other hand decreases with homogeneity of a region. So higher the value of β better is the segmentation.

4.5 Experimental Results

4.5.1 Test data for segmentation

We have done experimentation with our segmentation algorithm on a number of textured images taken from the data set discussed in Section 3.5.1 (Fig. 3.5). Several composite textured images are generated from the rotated versions of these textured images. Since we are mainly interested in the effectiveness, reliability and robustness of the rotation invariant features which are also immune to noise, rather than the segmentation performances, experimentation is carried out with composite textured images consisting of a moderate number of texture classes.

Here we have considered three different test images comprising of 2 texture classes and of size 256 × 256 (Figs. 4.4(a.i), 4.4(b.i), 4.4(c.i)), three more test images having 3 different classes of size 256 × 256 (Fig. 4.5(a.i), 4.5(b.i) and 4.5(c.i)) and a five-texture class image of size 512 × 512 (Fig. 4.6(a.i)). All these images are composed of textures that are obtained from common sources like Brodazt album [27] and MIT

Vision Texture database [114], and are considered as benchmark images for texture analysis. Moreover, we have considered two natural scenes (*scene*1 and *scene*2). These images have been captured under different conditions using different equipment. All these textures and natural scenes are gray-scale images, and the dynamic ranges are represented by eight bits per pixel. In order to make the textures indiscriminable for the local mean gray level or local variance, *i.e.*, to abolish any variation in illumination, the images have separately been globally histogram equalized prior to being used.

The details of the images are given below. The 256×256 2-class images with simple as well as complex borders are:

- Fig. 4.4(a.i), consisting of $D77$ and $D84$ [27].
- Fig. 4.4(b.i), consisting of $D77$ and $D24$ [27].
- Fig. 4.4(a.i), consisting of $D84$ and $D55$ [27].

The 256×256 images with simple and complex borders and with 3 classes are:

- Fig. 4.5(a.i), consisting of *Fabric*.0008, $D55$ and $D24$ [114] [27].
- Fig. 4.5(b.i), consisting of $D11$, $D84$ and $D24$ [27].
- Fig. 4.5(a.i), consisting of $D1$, *Fabric*.0008 and $D77$ [114] [27].

The 512×512 image with simple border but with as many as 5 classes is:

- Fig. 4.6(a.i), consisting of $D103$, $D20$, $D19$, $D17$ and $D24$ [27].

We have also considered two natural scenes of size 512×512 each:

- *Scene*1 (Fig. 4.15(1.a.i)) comprising of four classes namely, concrete, vegetation, grass and water body.
- *Scene*2 (Fig. 4.15(2.a.i)) also comprising of four classes *viz.*, concrete, vegetation and two types of grass.

All these images are subjected to different types and levels of noise, and different levels of gray-scale transform are incorporated.

4.5.2 Segmentation results

Fig. 4.4(a.i) shows a 2-class test image that we have studied, which consists of four regions, out of which, two regions consist of unrotated textures and the rest two comprise of rotated textures. Figs. 4.4(b.i) and 4.4(c.i) show two more 2-class test images, each consisting of five regions of which, two regions correspond to unrotated textures and the rest three comprise of rotated textures.

The images are decomposed into three levels of resolution as given in Fig. 3.2. We have taken into account all the three detail images and two of the low frequency images leaving out the lowest frequency image at the third level of resolution, to estimate the local energies around each pixel at different resolutions. That is in all, we are left with only five feature elements which means a considerable reduction in the dimensionality of the feature space, compared to other methods so far reported in the literature. Segmentation is performed with *k-means* clustering with k fixed at 2, results are shown in Figs. 4.4(a.ii), 4.4(b.ii) and 4.4(c.ii) respectively, and the classification performances are tabulated in Table 4.5.2.

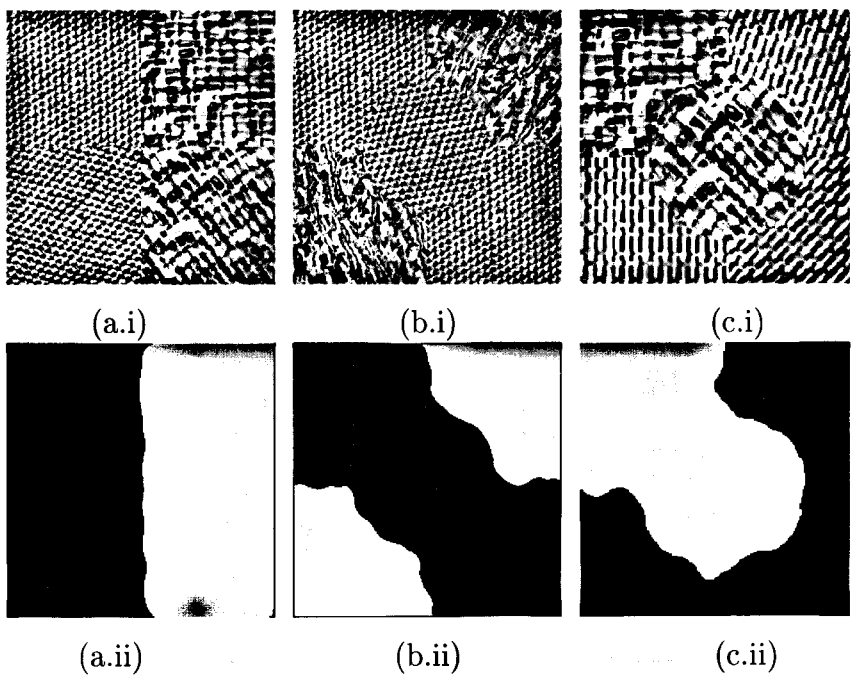


Figure 4.4: Composite textured images (a.i) $D77$ and $D84$, (b.i) $D77$ and $D24$, (c.i) $D84$ and $D55$ with their segmented outputs (k=2).

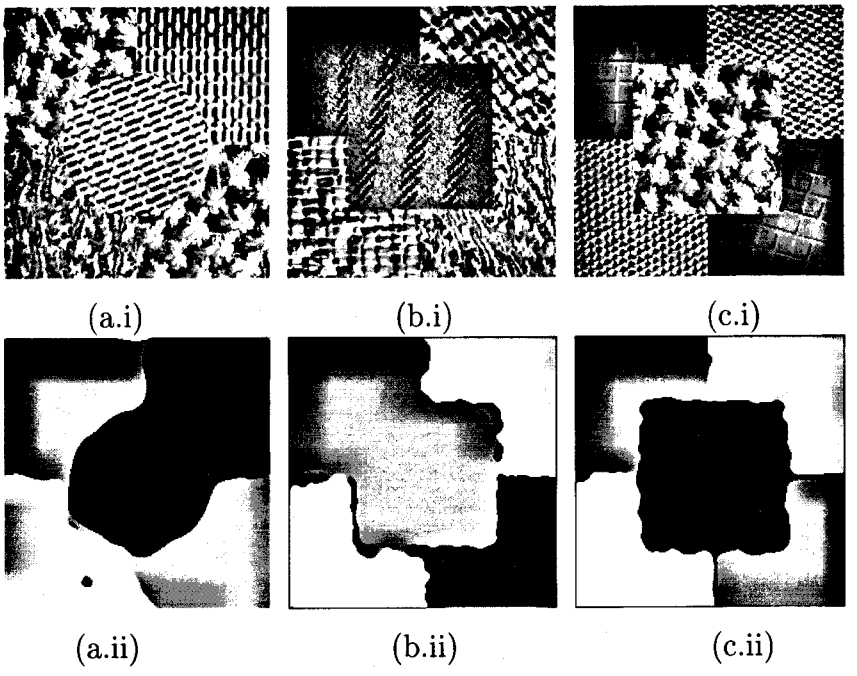


Figure 4.5: Composite textured images (a.i) *Fabric*.0008, $D55$ and $D24$ (b.i) $D11$, $D84$ and $D24$ (c.i) $D1$, *Fabric*.0008 and $D77$ with their segmented outputs (k=3).

Figs. 4.5(a.i), 4.5(b.i), and 4.5(c.i) show some more composite textured images that we have experimented with, which comprise of 3 different texture classes. The images consist of five regions with three unrotated textured regions and two rotated textured regions. Segmentation is carried out using *k-means* clustering with fixed k=3, the experimental results are shown in Figs. 4.5(a.ii), 4.5(b.ii) and 4.5(c.ii) respectively. The classification percentages for these test images are given in Table 4.5.2.

Table 4.1: % of Correct classification corresponding to the test images Figs. 4.4(a.i), 4.4(b.i),4.4(c.i) and Figs. 4.5(a.i), 4.5(b.i), 4.5(c.i)

Figure No.	4.4(a.i)	4.4(b.i)	4.4(c.i)	4.5(a.i)	4.5(b.i)	4.5(c.i)
% Classification	99.31	98.93	97.41	98.24	97.61	98.49

Fig. 4.6(a.i) shows another composite textured image that we have tested, which comprises of 5 different texture classes. The image consists of sixteen regions with

several rotated and unrotated textured regions. The segmentation result is given in Fig. 4.6(a.ii). To verify the invariance to gray-scale transformation we have performed some experiments on the same image. For this purpose two different sets of parameters have been used: γ=1.2, δ=0 (Fig. 4.6(b.i) and γ=0.5, δ=20 (Fig. 4.6(c.i)), segmentation results are shown in Figs. 4.6(b.ii) and 4.6(c.ii). It is observed that the performance of segmentation is not affected by gray-scale transformation, Table 4.5.2 validates this result.

Table 4.2: User Accuracy (U), Kappa values (κ) and classification in % for different scale and shift factors in Fig. 4.6

Parameter	Values	Class	Measure		Overall		% Classfn.
			U	κ	U	κ	
γ	1.0	1	96.27	95.36			
δ	0	2	97.38	96.78			
		3	99.48	99.36			
		4	96.31	95.49			
		5	94.34	92.52	96.76	95.85	96.73
γ	1.2	1	96.16	95.34			
δ	0	2	97.25	96.63			
		3	99.29	99.35			
		4	96.10	95.20			
		5	94.16	92.21	96.25	95.36	95.74
γ	0.5	1	96.18	95.23			
δ	20	2	96.15	95.43			
		3	98.21	98.26			
		4	95.08	95.04			
		5	94.13	92.05	95.12	94.19	94.69

The same image in Fig. 4.6(a.i) is corrupted with various Gaussian noise levels. Fig. 4.7(a.i) with mean =0, variance= 100, Fig. 4.7(b.i) with mean =0, variance= 300 and Fig. 4.7(c.i) with mean =0, variance= 300. Segmentation is carried out using *k-means* clustering with a fixed k=5.

The results obtained are very encouraging as shown in Figs. 4.7(a.ii), 4.7(b.ii), 4.7(c.ii) respectively. The percentages of correct classification of these images for different noise levels is tabulated in Table 4.5.2.

Fig. 4.8 gives a plot of percentages of correct classification for the test image [Fig. 4.7(a.i)] for various noise levels, all the parameters are given in Table 4.5.2

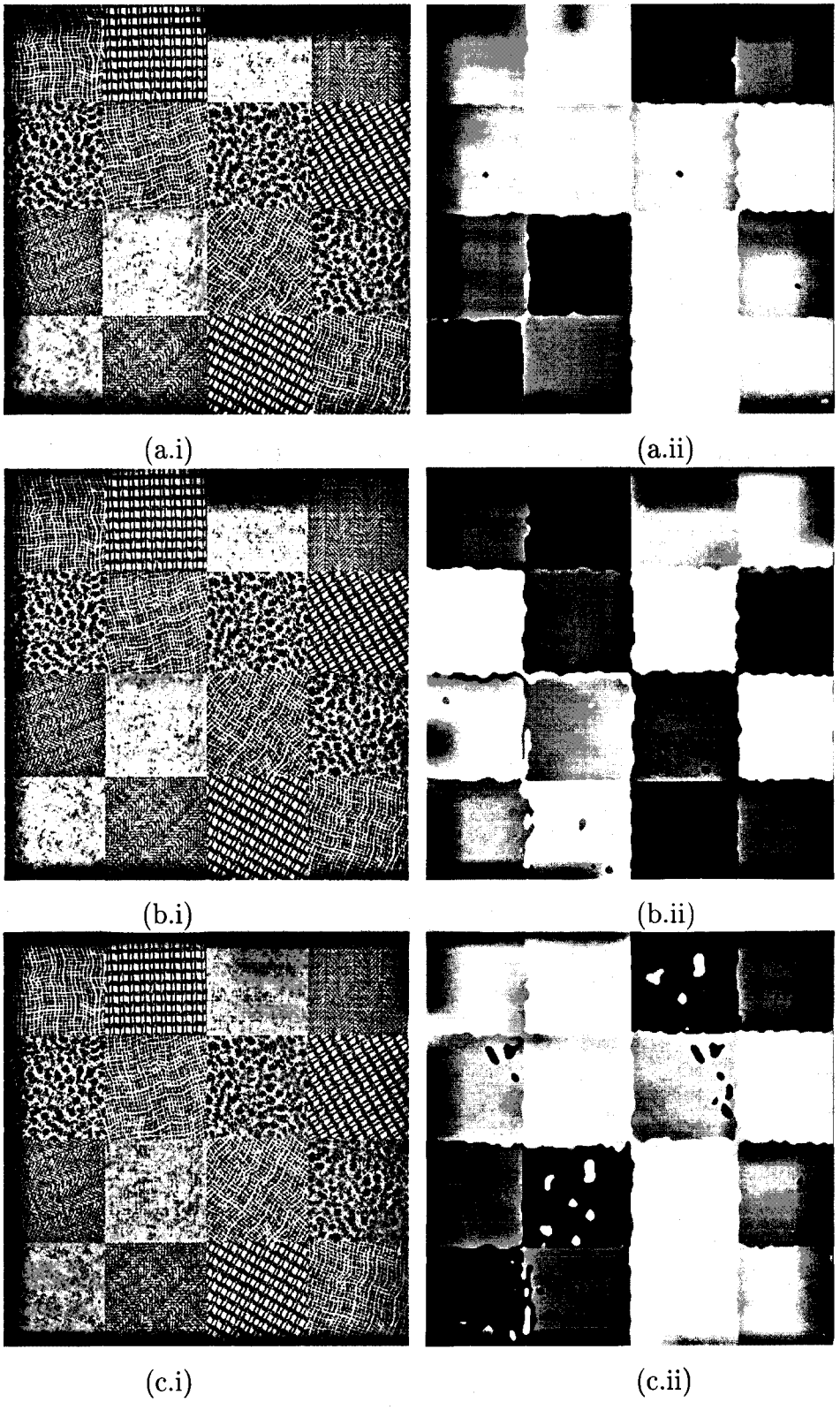


Figure 4.6: (a.i) Composite textured images $D103$, $D20$, $D19$, $D17$ and $D111$ without gray-scale transform, with gray-scale transforms for parameters (b.i) γ=1.2, δ=0 (c.i) γ=0.5, δ=20 and corresponding segmented outputs (k=5).

Table 4.3: User Accuracy (U), Kappa values (κ) and correct classification in % corresponding to the different types of noise for data in Figs. 4.6, 4.7

Type of noise	Parameter attributes	Values	Effect	Class	Measure		Overall		% Classfn.
					U	κ	U	κ	
Without noise	-	-	-	1	96.27	95.36			
				2	97.38	96.78			
				3	99.48	99.36			
				4	96.31	95.49			
				5	94.34	92.52	96.76	95.85	96.73
Gaussian	Mean 0 Variance 100.0		Additive	1	96.17	94.89			
				2	96.83	96.10			
				3	99.70	99.63			
				4	94.39	93.13			
				5	93.35	91.36	96.09	95.10	96.10
	Mean 0 Variance 200.0		Additive	1	94.76	93.01			
				2	97.31	96.69			
				3	99.52	99.42			
				4	91.44	89.53			
				5	92.97	91.30	95.20	94.00	95.04
	Mean 0 Variance 300.0		Additive	1	94.41	92.54			
				2	97.12	96.47			
				3	98.67	98.37			
				4	93.24	91.63			
				5	93.24	91.63	94.41	93.02	94.23

Table 4.4: % of Correct classification corresponding to the test image [Fig. 4.8(a.i)] for different Gaussian noise levels with mean=0

Variance	0	10.0	50.0	70.0	100.0	150.0	200.0	300.0
% Classification	96.73	96.19	96.17	96.16	96.10	96.09	95.04	94.23

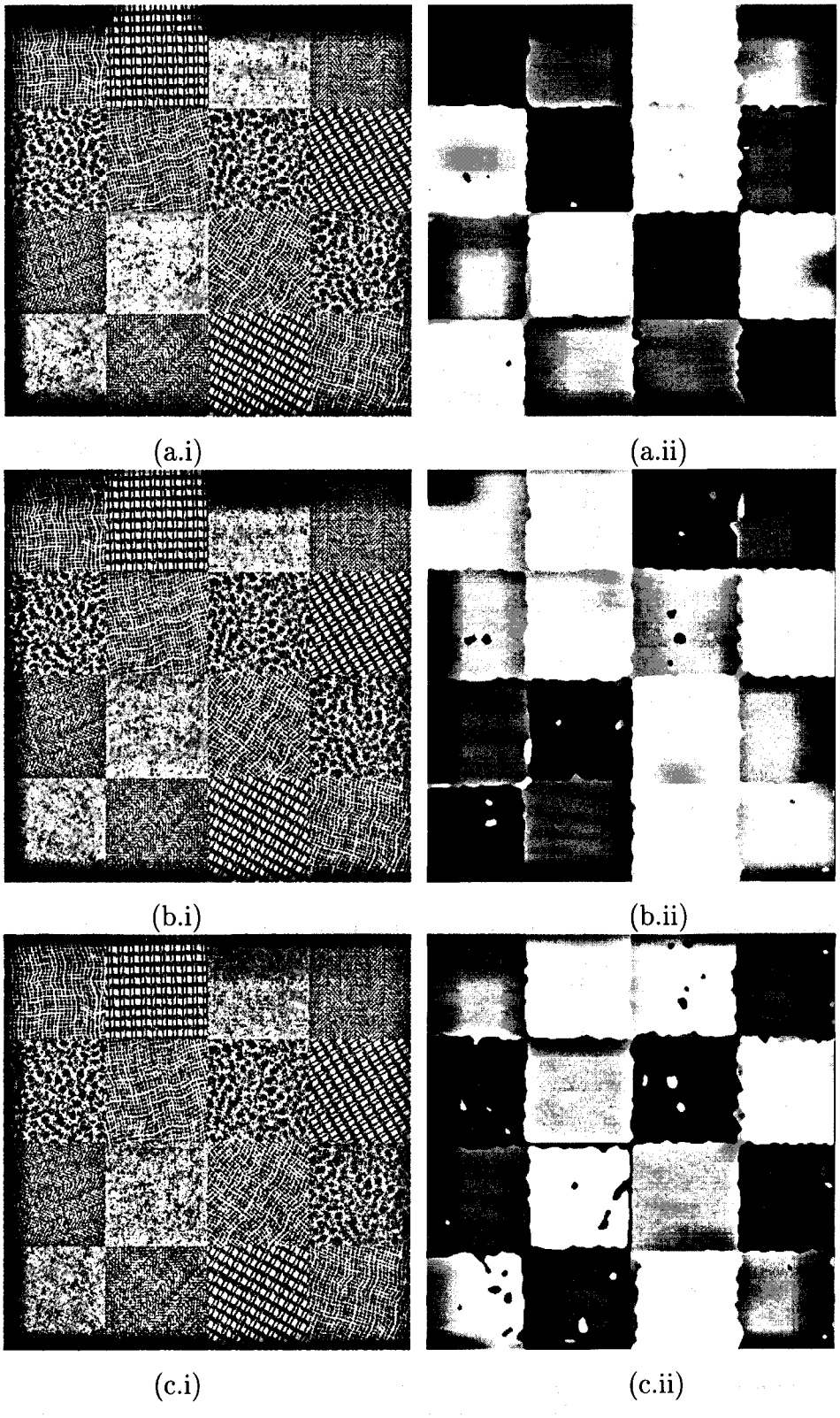


Figure 4.7: (a.i)-(c.i) Composite textured images corrupted with Gaussian noise and corresponding segmented outputs (k=5) (a.i) mean=0, variance=100.0 (b.i) mean=0, variance=200.0 (c.i) mean=0, variance=300.0.

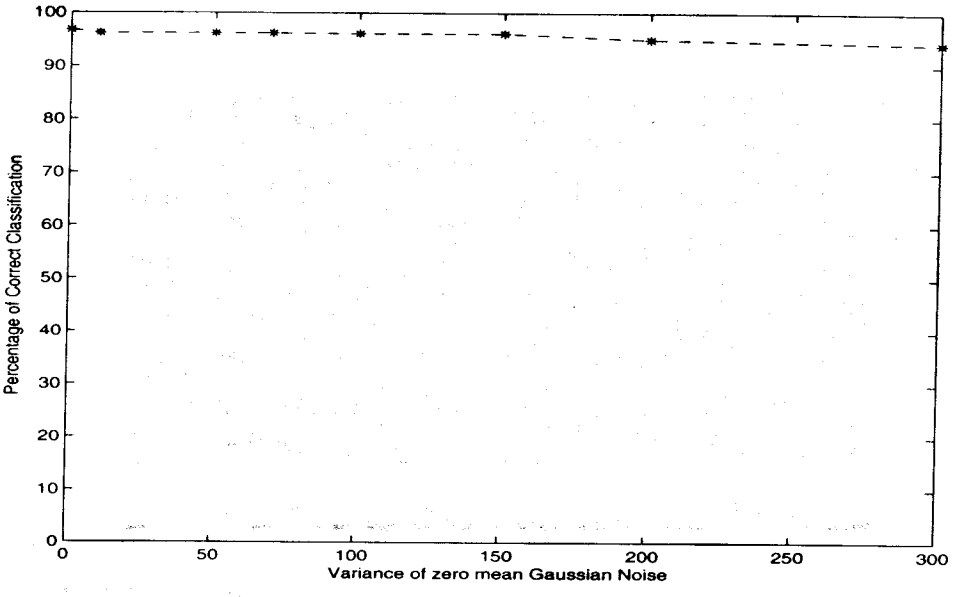
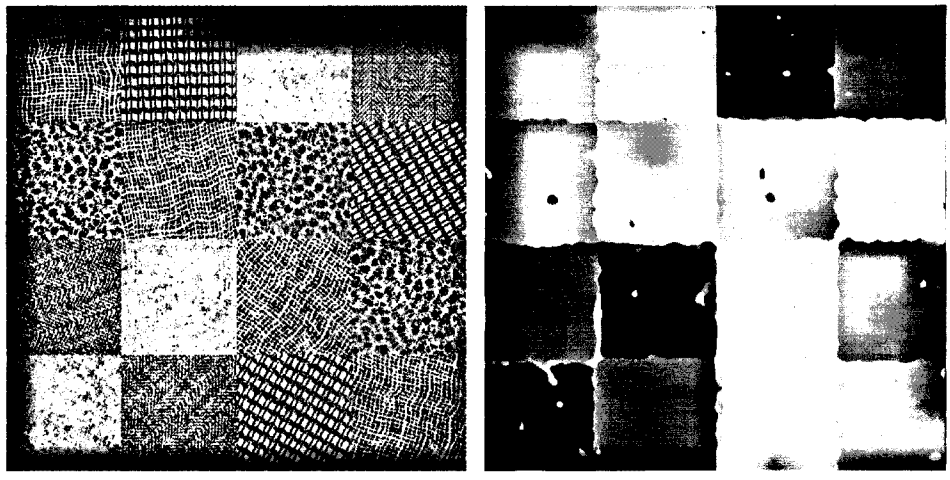


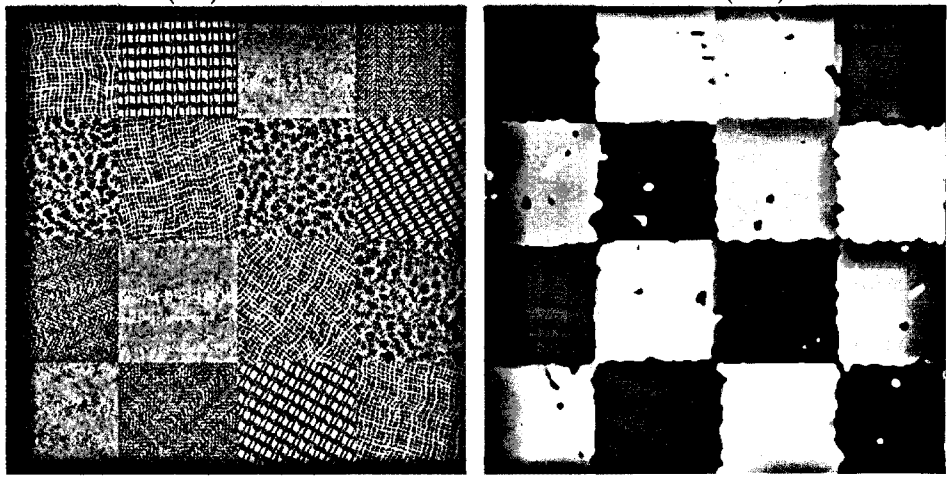
Figure 4.8: % of correct classification with Gaussian noise (noise parameters are given in Table 4.5.2)

The robustness of the proposed algorithm against noise is verified by experimentation on the test image of [Fig. 4.6(a.i)] corrupting it with several other types of noise, including Rayleigh, Exponential, Poisson, Uniform, Shot and Speckle noise having different parameter attributes. Fig. 4.9(a.i) corresponds to the image corrupted with Rayleigh noise with variance = 100, and subsequently for the Figs. 4.9(b.i) and 4.9(c.i) the variances are 200 and 300 respectively. Segmentation is carried out using *k-means* clustering with fixed k=5. The experimental results are shown in Figs. 4.9(a.ii), 4.9(b.ii), 4.9(c.ii). The classification performances are tabulated in Table 4.5.2.

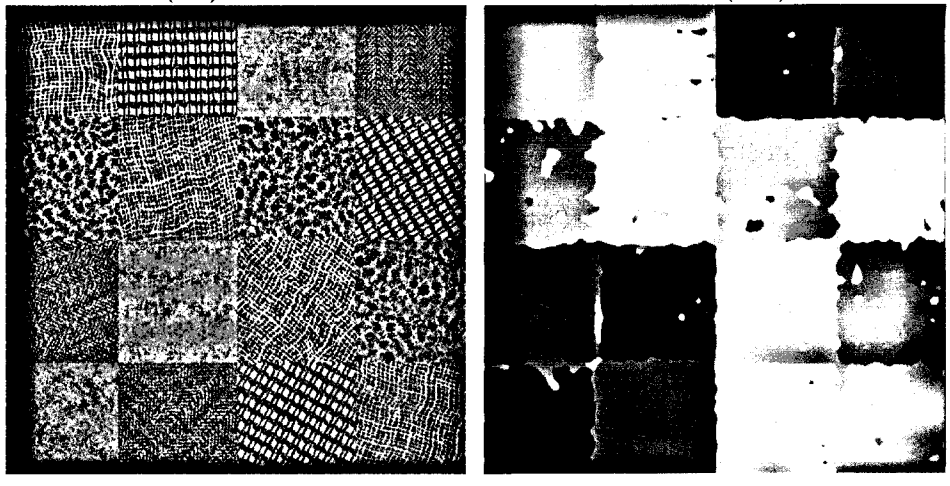
Figs. 4.10(a.i), 4.10(b.i) and 4.10(c.i) correspond to the images corrupted with exponential (variance = 200), Poisson (amount of time =200 and variance=20) and Uniform noise (minimum value = -20.0 and maximum value =+20.0). Table 4.5.2 gives the classification performances of these images. Segmentation is obtained using *k-means* clustering with fixed k=5. The experimental results are shown in Figs. 4.10(a.ii), 4.10(b.ii), 4.10(c.ii) and Table 4.5.2. Test images corrupted with speckle noise (mean = 1.0 and standard deviation = +.28) and shot noise (% of spikes 50 and value of spike 30.0) are shown in Figs. 4.11(a.i), and 4.11(b.i). The corresponding segmented outputs are shown in Figs. 4.11(a.ii), 4.11(b.ii) and Table 4.5.2.



(a.i) (a.ii)



(b.i) (b.ii)



(c.i) (c.ii)

Figure 4.9: Images corrupted with Rayleigh noise (a.i) var=100.0 (b.i) var=200.0 (c.i) var=300.0.

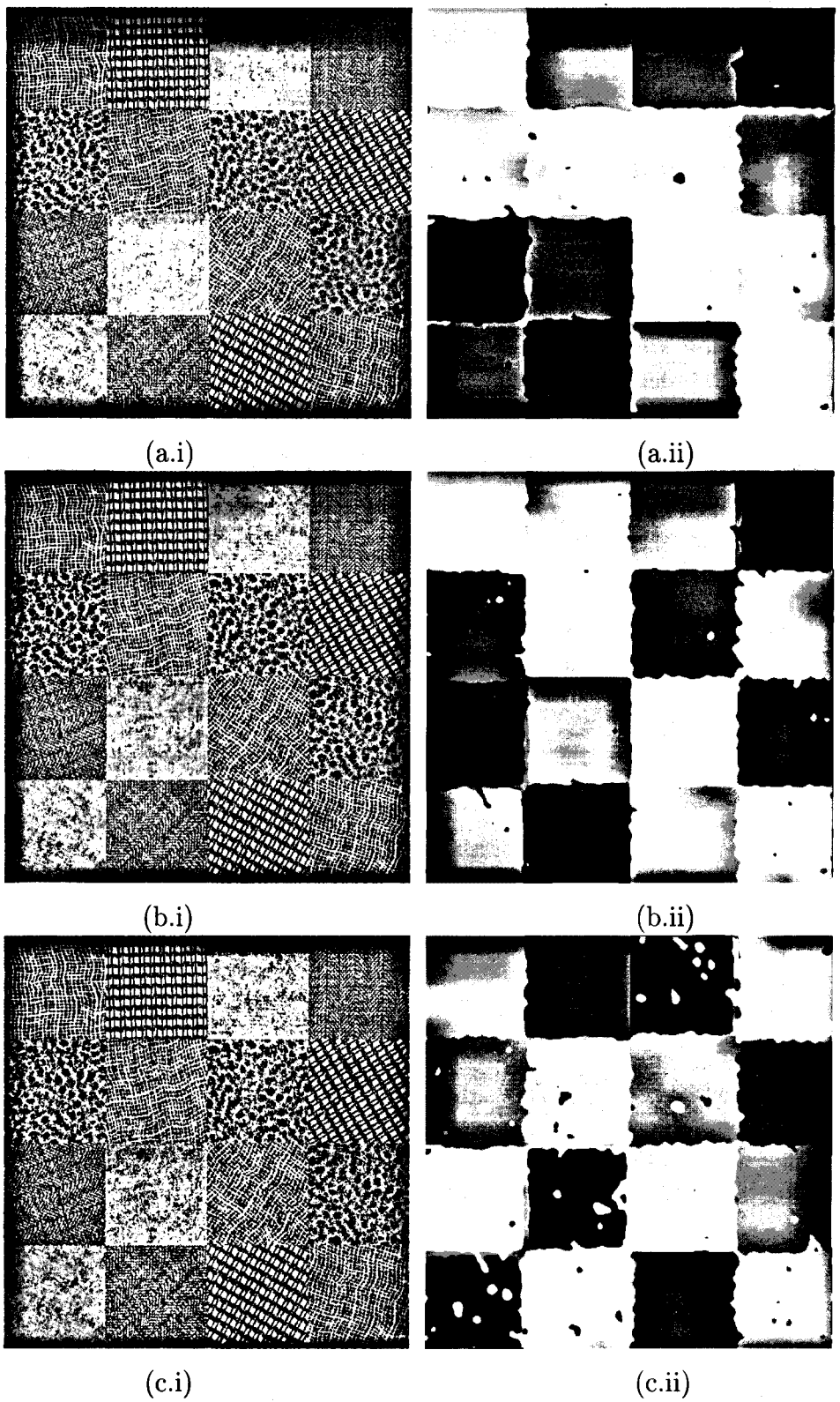


Figure 4.10: Images corrupted with (a.i) Exponential (b.i) Poisson (c.i) Uniform noises.

Table 4.5: User Accuracy (U), Kappa values (κ) and correct classification in % corresponding to the different types of noise for data in Fig. 4.9

Type of noise	Parameter attributes	Values	Effect	Class	Measure		Overall		% Classfn.
					U	κ	U	κ	
Rayleigh	Variance	100.0	Additive	1	94.48	92.63			
				2	97.04	96.36			
				3	99.72	99.65			
				4	91.95	90.15			
				5	93.00	91.34	95.24	94.05	95.06
	Variance	200.0	Additive	1	95.24	93.65			
				2	95.95	95.02			
				3	99.51	99.40			
				4	91.84	90.02			
				5	89.66	87.21	94.44	93.05	94.32
	Variance	300.0	Additive	1	93.42	91.22			
				2	94.48	93.21			
				3	99.74	99.67			
				4	90.41	89.49			
				5	91.44	89.41	94.00	92.62	93.92

In each of the above case noise is generated by 'KHOROS' in Silicon Graphics Irix 5.3. The classification percentages, the user accuracy (U) and Kappa (κ) values for each of the constituent class in the composite images are given in Tables 4.5.2, 4.5.2 and 4.5.2.

To prove the efficacy of our algorithm we have experimented on another test image Fig. 4.12 [Fig 4.2(i)] contaminated with Gaussian, Exponential and Rayleigh noise. The percentages of correct classification, user accuracy (U) and Kappa (*kappa*) of the segmented results are given in Table 4.5.2.

The above study reveals that we have obtained good quality of segmentation, which is invariant to rotation. Also no appreciable degradation in performances is observed while noise is being incorporated. It is to be noted that we have done median filtering of window size (9×9) as a post-processing step over the classified images, to simulate the benefit of local neighborhood information.

The objectives of filtering and local energy estimator, are to transform the edges between textures into detectable discontinuities. For a comparison purpose, we study the segmentation results using the Marr-Hildreth/Laplacian of Gaussian (LOG) operator as the filter, which is a zero crossing detector as well as orientation independent. Fig. 4.13(a.ii) shows the segmented output of the test image Fig. 4.13(a.i) that we have

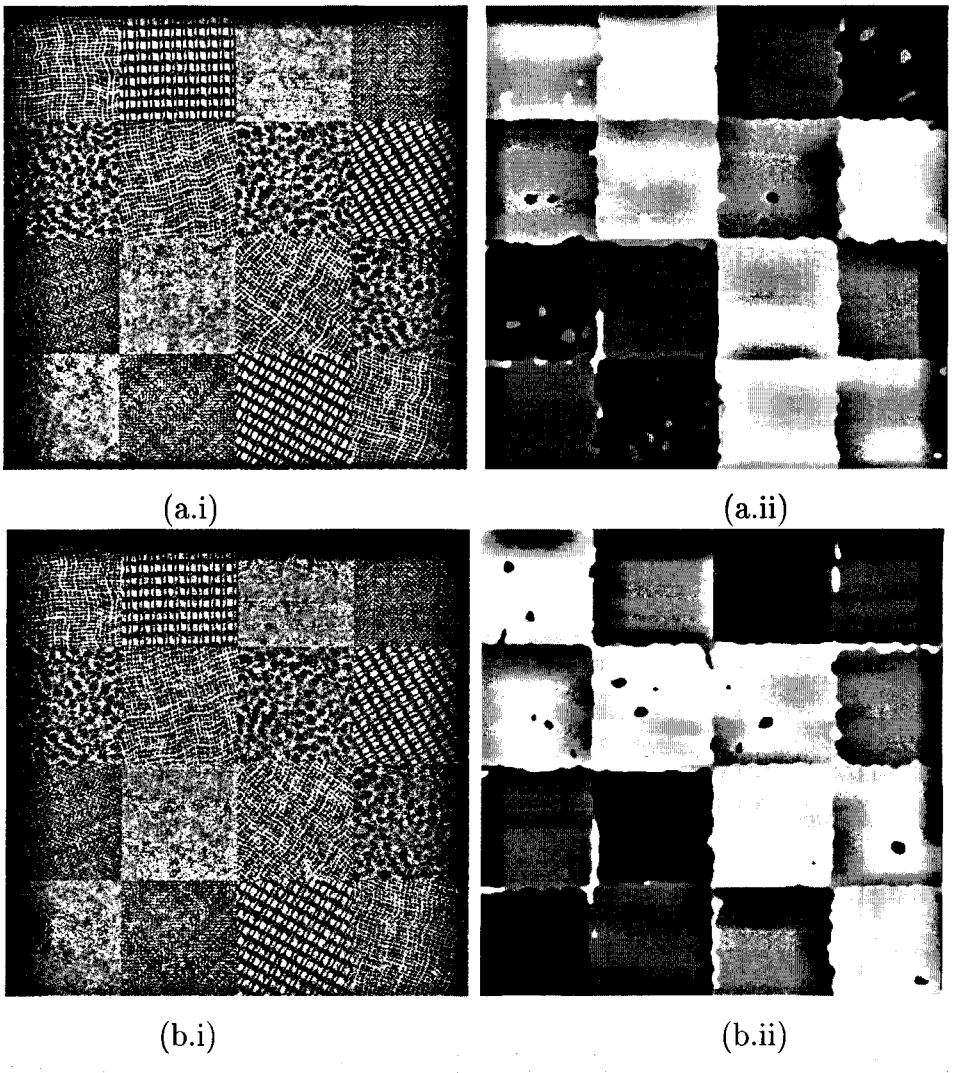


Figure 4.11: Images corrupted with (a.i) Speckle (b.i) Shot noises and their segmented output (k=5).

Table 4.6: User Accuracy (U), Kappa values (κ) and correct classification in % corresponding to the different types of noise for data in Figs. 4.10 4.11

Type of noise	Parameter attributes	Values	Effect	Class	Measure		Overall		% of Classfn.
					U	κ	U	κ	
Exponential	Variance	200.0	Additive	1	95.43	93.92			
				2	97.06	96.38			
				3	98.22	97.81			
				4	95.09	93.99			
				5	92.67	90.33	95.69	94.62	95.62
Poisson	Amount of time	200	Additive	1	95.26	93.68			
				2	96.14	95.25			
	Variance	20.0		3	99.14	98.95			
				4	95.81	94.87			
				5	91.30	89.23	95.53	94.41	95.41
Shot	% of spikes	50	Additive	1	96.44	95.26			
	Value of spike	30.0		2	95.71	94.73			
				3	98.23	97.90			
				4	93.18	91.65			
				5	90.77	88.58	94.88	93.60	94.87
Uniform	Minimum value	-20.0	Additive	1	96.18	94.91			
				2	97.79	97.28			
	Maximum value	+20.0		3	99.16	98.97			
				4	91.01	89.00			
				5	92.42	90.62	95.31	94.16	93.31
Speckle	Mean	1.0	Multipl-	1	94.59	92.72			
	Standard deviation	+0.28	-icative	2	95.86	94.91			
				3	92.95	91.32			
				4	94.85	93.70			
				5	92.77	91.05	94.19	92.74	94.19

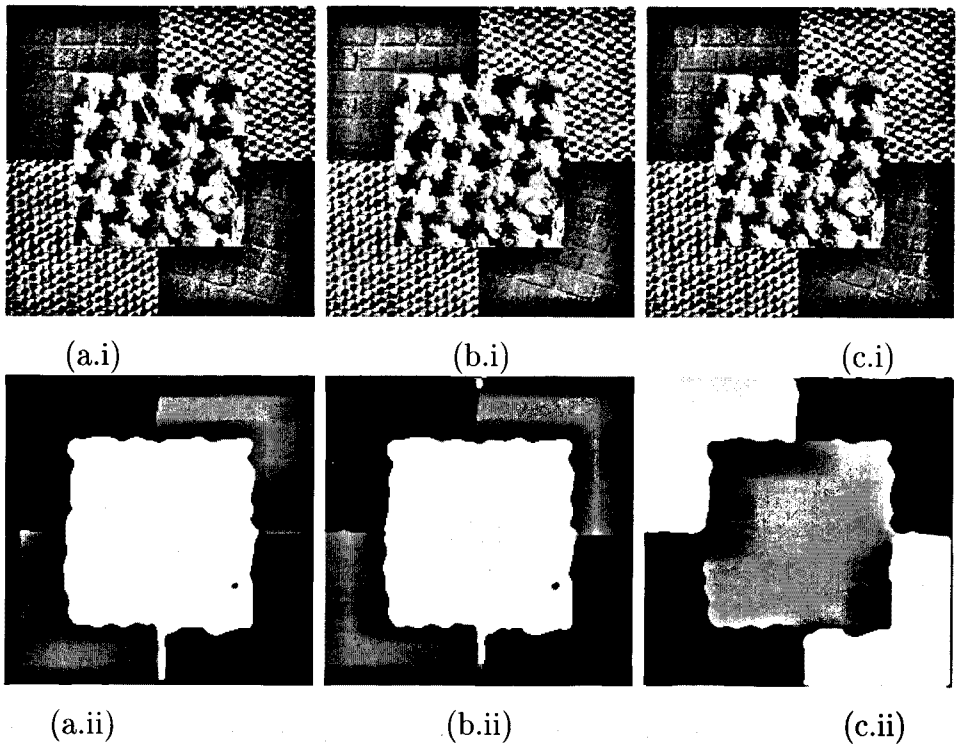


Figure 4.12: Images corrupted with (a.i) Gaussian (b.i) Rayleigh (c.i) Exponential noises.

Table 4.7: User Accuracy (U) and kappa values (κ) in % corresponding to the different types of noise for data in Fig. 4.12

Type of noise	Parameter attributes	Values	Effect	Class	Measure		Overall		% of Classfn.
					U	κ	U	κ	
Without noise	–	–	–	1	97.14	94.39			
				2	94.67	92.75			
				3	97.60	97.15	96.47	94.76	96.85
Gaussian	Mean	0	Additive	1	95.72	93.78			
	Variance	200.0		2	94.93	92.64			
				3	97.75	96.39	96.14	94.27	96.21
Rayleigh	Variance	200.0	Additive	1	95.95	94.12			
				2	94.70	92.30			
				3	97.91	96.66	96.19	94.36	96.08
Exponential	Variance	200.0	Additive	1	95.93	94.90			
				2	95.37	93.28			
				3	96.82	94.90	96.04	94.09	96.27

already used before [Fig. 4.6(a.i)], using LOG as the edge detector. Fig. 4.13(b.ii) illustrates the performance of the LOG edge detector, with the test image having variations in gray-scale (Fig. 4.13(b.i)). The segmented output of Fig.4.13(c.i), which is contaminated with Gaussian noise, is given in Fig. 4.13(c.ii). The performance results are summarized in Table 4.5.2. From the above investigation it is quite evident that our method gives a much better result over the LOG operator.

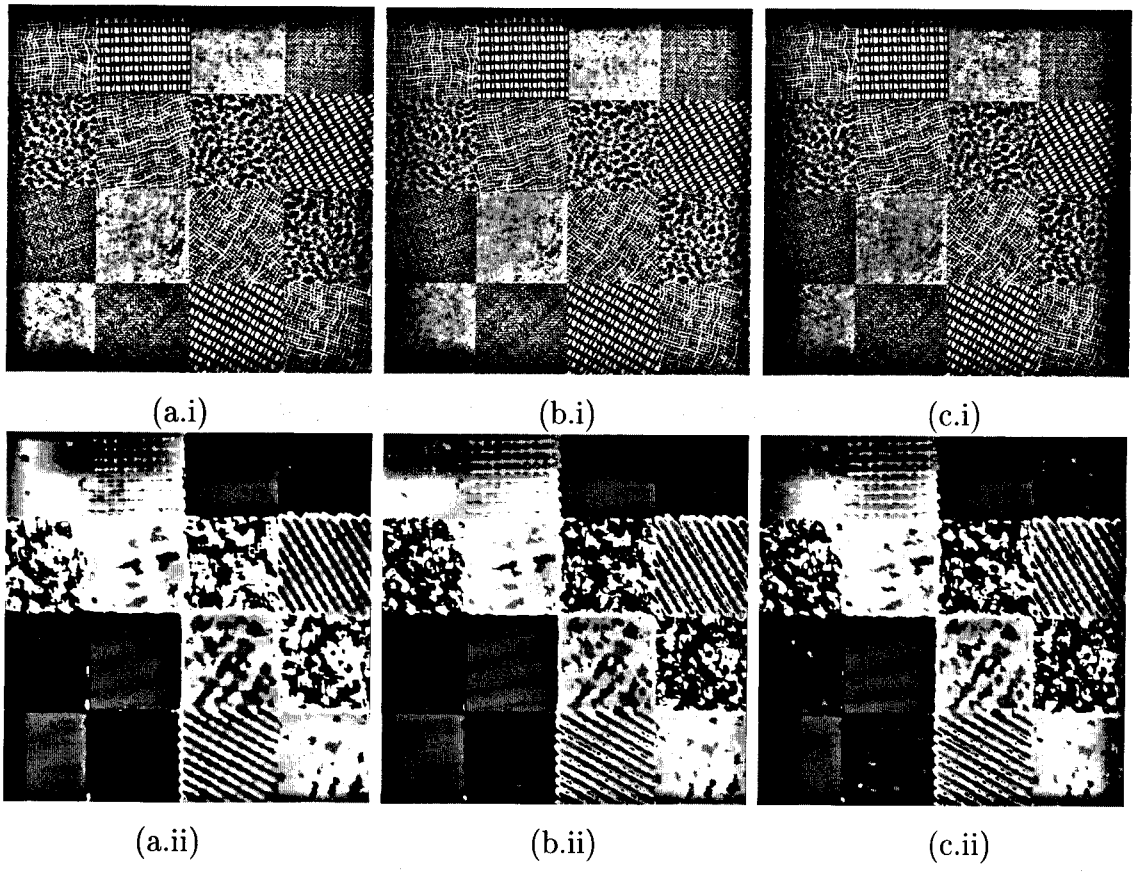
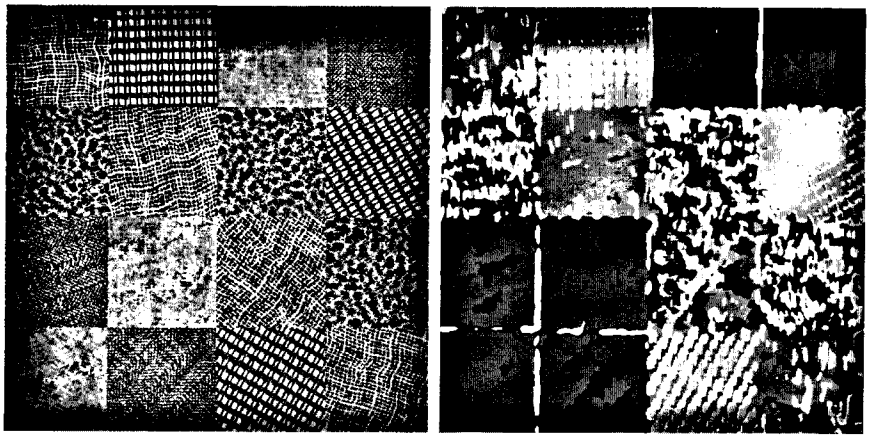


Figure 4.13: (a.i) Test image of Fig. 4.6(a.i) (b.i) with gray-scale transform α=1.2, δ=0 (c.i) with Gaussian noise (mean=0.0, variance=100.0) (a.ii)-(b.ii)-(c.ii) corresponding segmented outputs (k=5) using Laplacian of Gaussian (LOG) as the zero crossing edge detector.

The basic philosophy behind this study is to work with a filter, which is circularly symmetric. We would prove that the choice of any filter other than this will not give rotational invariance. For this we demonstrate the segmentation on Fig. 4.6(a.i) using Canny's edge operator or Derivative of Gaussian (DOG). The segmentation result is shown in Fig. 4.14(b) and a classification score of 59.23% is obtained, from which it

Table 4.8: User Accuracy (U), kappa values (K) and classification in % for test image Fig. 4.6(a.i) using Marr-Hildreth operator as edge detector, results shown in Fig. 4.13

Parameter	Values	Overall		% classfn.
		U	K	
Without noise & gray-scale transform	-	72.99	66.24	71.22
With gray-scale transform	α=1.2 δ= 0	67.68	59.63	66.56
With Gaussian noise	mean = 0.0 variance=100.0	67.60	59.61	66.57

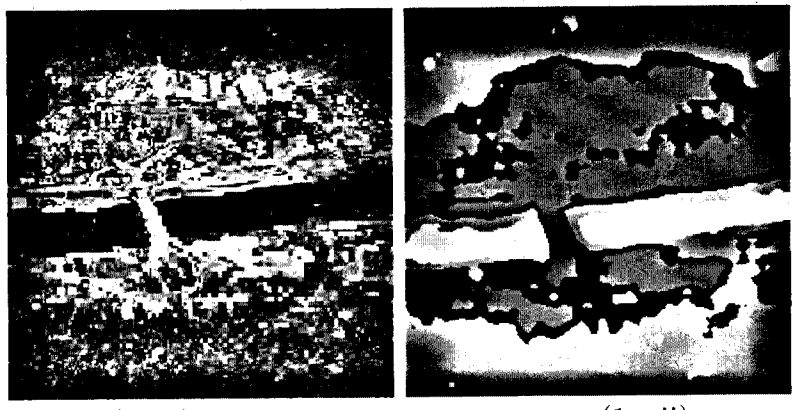


(a.i) (a.ii)

Figure 4.14: (a) Test image of Fig. 4.6(a.i) (b) corresponding segmented output (k=5) using Derivative of Gaussian (DOG) as the edge detector.

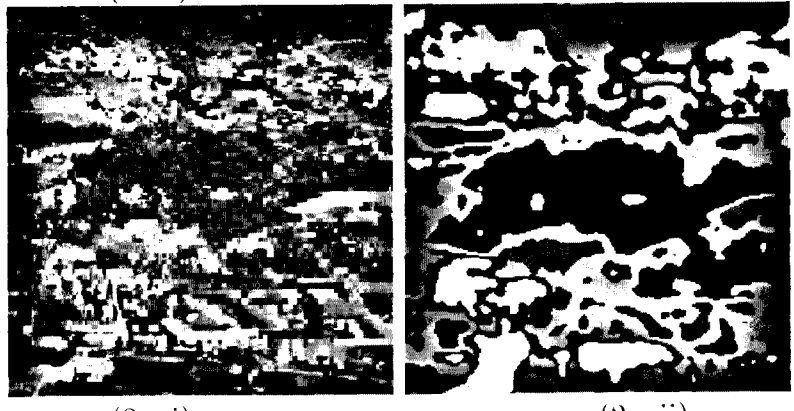
may be inferred that Canny's edge detector can not give rotational invariance.

The effectiveness of the algorithm is substantiated, by applying it on several natural scenes. We present here two such test images (Figs. 4.15(1.a.i) and 4.15(2.a.i)), each consisting of 4 classes. Both the images are corrupted with zero mean Gaussian noise, and segmentation results are reported in terms of the quantitative measure as stated in the previous Section 4.4. The β values for Figs. 4.15(1.a.ii) and 4.15(1.b.ii) are 1.46321 and 1.462958 respectively. While that for Figs. 4.15(2.a.ii) and 4.15(2.b.ii) are 1.329699 and 1.32788 respectively. From this study we can infer that, even with corruption of noise there is no appreciable degradation in the segmentation results.

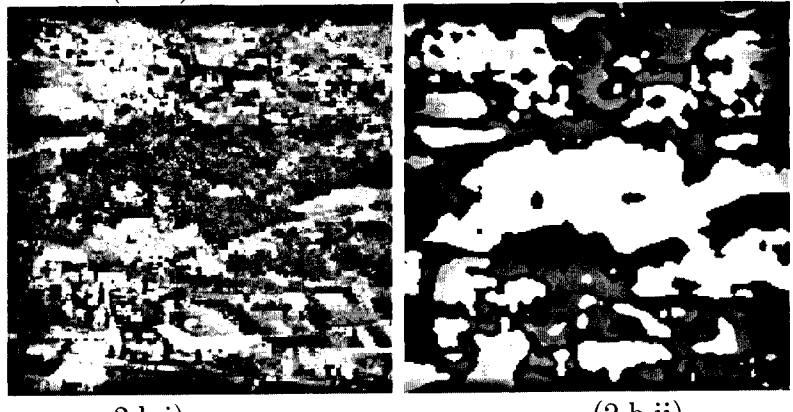


(1.a.i) (1.a.ii)

(1.b.i) (1.b.ii)



(2.a.i) (2.a.ii)



2.b.i) (2.b.ii)

Figure 4.15: (1.a.i)-(2.a.i) Natural scene-1 & scene-2 (1.b.i)-(2.b.i) corrupted with Gaussian noise (mean=0.0, variance=100.0) (1.a.ii),(1.b.ii)-(2.a.ii),(2.b.ii) with their corresponding segmented outputs (k=4).

4.6 Conclusion

In this chapter we have presented our studies on gray-scale transform invariant texture segmentation which is also invariant to rotation and tolerant to noise corruption. A computationally efficient method for feature extraction has been developed by simple modifications of the method discussed in the previous chapter. We have experimentally shown that the features are really invariant to any gray-scale transform by studying the gradient magnitude histograms of several gray-scale transformed versions of the same texture.

Judging from the results of our experiment over several composite textured images we can conclude that our segmentation scheme performs reasonably well so far as robustness is concerned. But some edge inaccuracies are observed apart from some misclassification. One reason for this might be because of the lack of information available for the selection of proper local window size.

Another important point to be noted here is, the standard two-band wavelet decomposition implies fine frequency resolution in the lower frequency bands than in the higher band and hence are not suitable for the analysis of high frequency signals with relatively narrow bandwidth. Therefore we conjecture that an M-band wavelet analysis where M is greater than two may be able to characterize textures better than the two-band wavelets. Our segmentation algorithm is based on the concept that two different texture classes comprise of different signal energies. The accuracy of the algorithm depends on how effectively it can characterize each texture in terms of its signal energies. This in turn demands for decomposition of each texture into a large number of subbands. So we are of the opinion that M-band wavelets are more suitable for this purpose of segmentation. In the next chapter we would discuss how an M-band wavelet can be applied for texture segmentation.

Chapter 5

Adaptive Texture Segmentation using M-band Wavelet Transform

5.1 Introduction

In this chapter we investigate the problem of segmentation of two-class texture images by using a generalization of the wavelet transforms to the M-band case [5, 4]. M-band wavelet is a tool for viewing signals at different scales and directions and decomposes a signal by projecting it onto a family of functions generated from a single wavelet basis via its dilations and translations [150, 17]. The mathematical details of which are already discussed in Chapter 2 (Section 2.9.4).

As mentioned in Chapter 4, standard 2-band wavelets are not suitable for the analysis of high frequency signals with relatively narrow bandwidth. So the main motivation of the present work is to use the decomposition scheme based on M-band wavelets, which may improve the segmentation accuracy. The motivation for a larger $M(M > 2)$ comes from the fact that, unlike the standard wavelet decomposition which results in a logarithmic frequency resolution, the M-band decomposition generates a mixture of a logarithmic and linear frequency resolution. That is, generates a more flexible tiling of the scale-space plane than that resulting from 2-band wavelet. Furthermore, M-band wavelet decompositions yield a large number of subbands which are required for good quality segmentation.

Texture analysis using M-band wavelet have been investigated by Greiner *et al.* [73]. They have worked with a 3-band extension of 2-band bi-orthogonal wavelets. Relevant texture features are extracted by a 3-band bi-orthogonal wavelet, and are subsequently used in classification of textures. Recently Chitre and Dhawan [41] have used M-band wavelets for texture classification.

In the present study, the image is first decomposed into $M \times M$ subbands by applying the M-band wavelet in a separable manner without downsampling, which gives an oversampled (redundant) representation of the image. Then various combinations of these subbands are taken that gives multidirectional and multiscale representation of the image. Texture features are extracted from these subbands in the next step. This is achieved by estimating in each subband, the local energy around each pixel over a small neighborhood. This requires a nonlinear operation followed by smoothing (averaging) with carefully chosen window size. In our work this is determined adaptively based on the spectral frequency content of the images.

The large number of features resulting from the above operations, may procreate information that are redundant. It becomes necessary to decrease the redundant number of features, while, at the same time, retain as much discriminatory information as possible. Our efforts are to characterize a texture most efficiently by finding out a low dimensional and near optimal set of features from the extracted feature set. Thus the problem, in a sense, boils down to extraction of multiscale features followed by the selection of the appropriate ones using some feature selection algorithms. Note that feature selection is an important and integral part of a pattern recognition system. Different useful classical techniques for feature selection are available in [53]. In the present study, the extracted features are first ranked according to some performance criterion and then select a few of them to reduce redundancy.

Superior discriminating capability of the extracted features over those obtained by several existing methods is shown. This is verified with segmentation results on various texture images including some benchmark images containing overlapping texture classes.

The chapter is organized as follows. The following section presents the M-band wavelet filtering technique and extraction of features. This section further discusses about the proposed adaptive methodology for averaging window size selection. In the subsequent section the feature ranking and selection methodology is discussed. It also presents a brief account of the statistical significance of the extracted features. This is followed by discussions on the experimental results of our investigation.

5.2 Computing Texture Features

An important aspect of texture analysis is to develop a set of texture measures (features) that can successfully discriminate arbitrary textures. The feature extraction scheme is shown in Fig. 4.2.

5.2.1 M-band wavelet filters

The filter bank in essence is a set of bandpass filters with frequency and orientation selective properties. In the filtering stage we work with the orthogonal, linear phase

M-band wavelet filters discussed in [17]. These filters have perfect reconstruction with quadrature mirror filter (PR-QMF) structure and are symmetrical. Intuitively, for good edge boundary localization, it is desirable to have a filter with compact spatial domain representation, while for reliable discrimination of different texture frequency contents the filter should have a good frequency response localization and high stop band attenuation. Also symmetry of the filter responses is an important factor. A non-symmetric filter response consistently leads to edge detection error and consequently higher classification error. The filters we have chosen satisfy all of the above requirements since, QMF filters have significantly compact frequency response.

The texture image is decomposed into $M \times M$-channels, corresponding to different directions and resolutions. The $M(=4)$-band wavelet system consist of the scaling filter ϕ and the wavelet filters ψ_m for $m = 1, 2, 3$. The impulse responses of these filters are denoted by $h(n)$ and $g_m(n)$ respectively, and are tabulated in Table 5.1. The size of filter is an important factor. The sequence of low pass and bandpass filters of increasing width corresponding to increasing level of decomposition, are expanded by inserting an appropriate number of zeros between taps of filters. So if the filter length becomes large it may bias the decomposition of an image of finite dimensions. We have chosen an eight-tap filter here, because we have assumed that a small length of filter may provide us with unbiased decomposition and less computational cost.

Table 5.1: Filter Coefficients for Eight-tap Four-band Wavelet Transform

No. of taps (n)	$h(n)$	$g_1(n)$	$g_2(n)$	$g_3(n)$
0	-0.067371764	-0.094195111	-0.094195111	-0.067371764
1	0.094195111	0.067371764	-0.067371764	-0.094195111
2	0.40580489	0.56737176	0.56737176	0.40580489
3	0.56737176	0.40580489	-0.40580489	-0.56737176
4	0.56737176	-0.40580489	-0.40580489	0.56737176
5	0.40580489	-0.56737176	0.56737176	-0.40580489
6	0.094195111	-0.067371764	-0.067371764	0.094195111
7	-0.067371764	0.094195111	-0.094195111	0.067371764

The transfer function of a digital filter is defined by the discrete Fourier transform of its impulse response ($h(n)$) (already defined in Section 2.9.1). Let us denote the transfer function of the scaling function (low pass filter) by H_1 and those corresponding to the

wavelet functions (bandpass filters) by $H_{m'}$ for $m' = 2, 3, 4$. That is

$$H_1(\omega) = \sum_{-\infty}^{\infty} h(n) e^{i\omega n}.$$

and

$$H_{m'}(\omega) = \sum_{-\infty}^{\infty} g_m(n) e^{i\omega n} \,|_{m=1,2,3,m'=2,3,4}$$

The suffix denotes the $x(columns)$ and $y(rows)$ directions in an image matrix respectively. The scaling and wavelet functions are shown in Fig. 5.1 and their respective transfer functions are illustrated in Fig. 5.2.

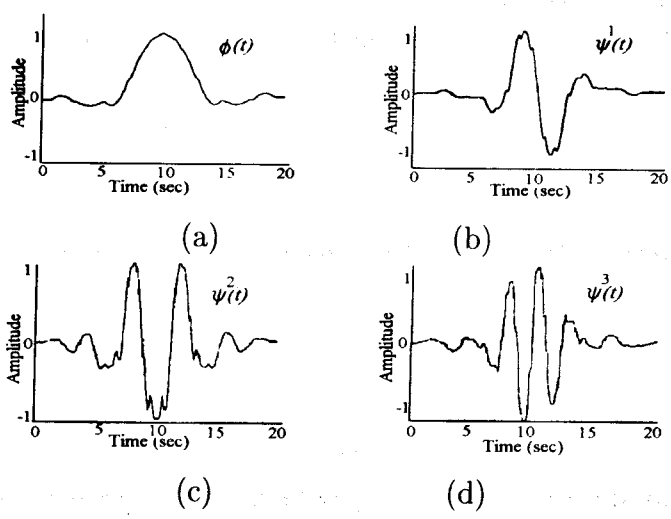


Figure 5.1: The eight-tap and 4-band (a) Scaling and (b)-(d) wavelets

In this work we have obtained the M^2-channel 2D separable transform by the tensor product of M-band 1D wavelet filters but without any subsampling. The x, y^{th} resolution cell is obtained via the filtering steps

$$H_{11} = H_1(\omega_x) H_1(\omega_y) \tag{5.1}$$

and

$$H_{m'_x, m'_y} = H_{m'_x}(\omega_x) H_{m'_y}(\omega_y) \quad \text{for } m'_x = m'_y = 2, 3, 4. \tag{5.2}$$

The decomposition of the image into $M \times M$ (=16) channels is illustrated in Fig. 5.3(a).

Since the spectral response to edges of an image is strongest in direction perpendicular to the edge, while it decreases as the look direction of the filter approaches that of the edge. Therefore we can perform edge detection by using 2D filtering as follows,

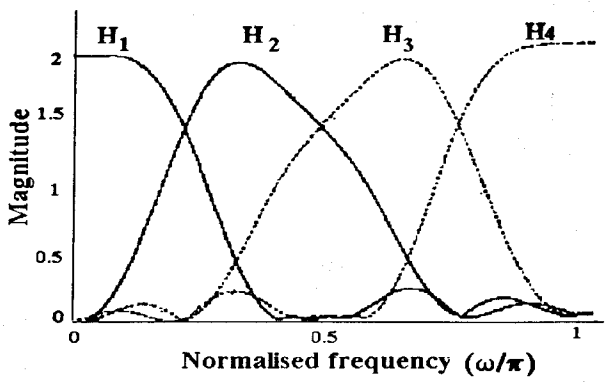


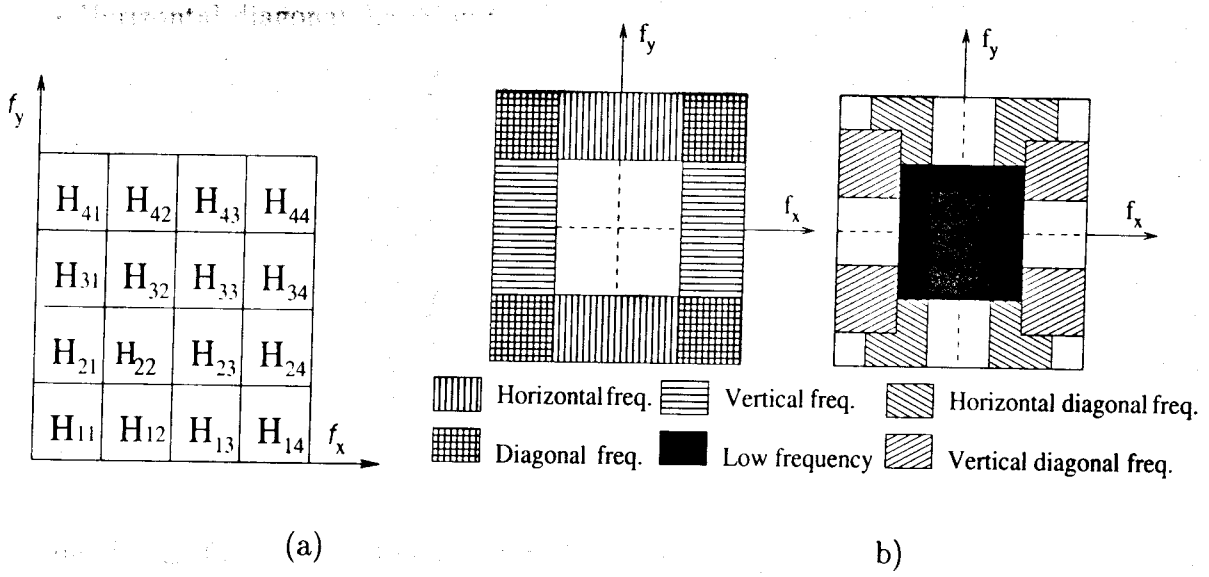
Figure 5.2: Frequency domain plot of the scaling and the wavelet functions

- *horizontal edges* are detected by high pass filtering on columns and low pass filtering on rows.
- *vertical edges* are detected by low pass filtering on columns and high pass filtering on rows.
- *diagonal edges* are detected by high pass filtering on columns and high pass filtering on rows.
- *horizontal-diagonal edges* are detected by high pass filtering on columns and low pass filtering on rows.
- *vertical-diagonal edges* are detected by low pass filtering on columns and high pass filtering on rows.

A typical edge detection filter corresponding to a particular direction covers a certain region in the 2D spatial frequency domain, this is illustrated in Fig. 5.3(b) where, f_x and f_y are the horizontal and vertical frequencies. Based on this concept several wavelet decomposition filters are possible which are given by, $\sum_{Reg} H_{\hat{m}_x,\hat{m}_y}$ for $\hat{m} = 1,\ldots,4$, where *Reg* denotes the frequency sector corresponding to a certain direction and scale.

Since the filter system we are using here is orthogonal and has QMF structure, therefore

$$\sum_{\hat{m}_x=1}^{4}\sum_{\hat{m}_y=1}^{4} H_{\hat{m}_x\hat{m}_y}H^{*}_{\hat{m}_x\hat{m}_y} = 1.$$



(a) b)

Figure 5.3: (a) Frequency bands corresponding to decomposition filters (b) Frequency sector representations for filtering in horizontal (H),vertical (V) and diagonal (D) directions

This signifies that the resulting 2D filters treats all the frequencies in a resolution cell equally. The number of channels as well as the number of possible filter combinations depend on the value of M. The decomposition filters $\sum_{Reg} H_{\hat{m}_x \hat{m}_y}$ are formed as follows for different directions in increasing level of resolutions.

⋆ Horizontal direction :

$$\begin{aligned} filt_{hor_1} &= H_{12} \\ filt_{hor_2} &= H_{12} + H_{13} \\ filt_{hor_3} &= H_{12} + H_{13} + H_{14} + H_{24} \end{aligned}$$

⋆ Vertical direction :

$$\begin{aligned} filt_{ver_1} &= H_{21} \\ filt_{ver_2} &= H_{21} + H_{31} \\ filt_{ver_3} &= H_{21} + H_{31} + H_{41} + H_{42} \end{aligned}$$

⋆ Diagonal direction :

$$\begin{aligned} filt_{diag_1} &= H_{22} \\ filt_{diag_2} &= H_{22} + H_{33} \\ filt_{diag_3} &= H_{22} + H_{33} + H_{44} \end{aligned}$$

⋆ Horizontal-diagonal direction :

$$\begin{aligned} filt_{hdiag_1} &= H_{12} \\ filt_{hdiag_2} &= H_{12} + H_{23} \\ filt_{hdiag_3} &= H_{12} + H_{23} + H_{34} \end{aligned}$$

⋆ Vertical-diagonal direction :

$$\begin{aligned} filt_{vdiag_1} &= H_{21} \\ filt_{vdiag_2} &= H_{21} + H_{32} \\ filt_{vdiag_3} &= H_{21} + H_{32} + H_{43} \end{aligned}$$

These filter responses basically give a measure of signal energies at different directions and scales, the corresponding filtered images are denoted by F_{hor_j}, F_{ver_j}, F_{diag_j}, F_{hdiag_j} and F_{vdiag_j} for $j = 1, 2, 3$ as shown in Fig. 5.4.

5.2.2 Local energy estimator

As already been discussed in the previous chapter (Section 4.3.2) that the objective of the local energy estimator is to estimate the energy of the filter output in a local region. Here we have used the most popular magnitude operation $|\cdot|$ as a nonlinear operator.

This is followed by a Gaussian low pass (smoothing) filter of the form,

$$h_G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x^2+y^2)}$$

where, σ defines the spatial extent of the averaging filter.

Formally, the feature image $Feat_b(x, y)$ corresponding to filtered image $F_b(x, y)$ is given by,

$$Feat_b(x, y) = \sum_{(x',y') \in G_{xy}} \Gamma(F_b(x', y') h_G(x - x', y - y')) \tag{5.3}$$

where $b = hor_i, ver_i$ etc., $\Gamma(\cdot)$ is the nonlinear function and G_{xy} is a $G \times G$ window centered at pixel with co-ordinates (x, y).

An important parameter in this regard is the size G of the Gaussian averaging window. More reliable measurement of texture feature demands larger window size. On the other hand, more accurate localization of region boundaries requires smaller window. This is because averaging blurs the boundaries between textured regions.

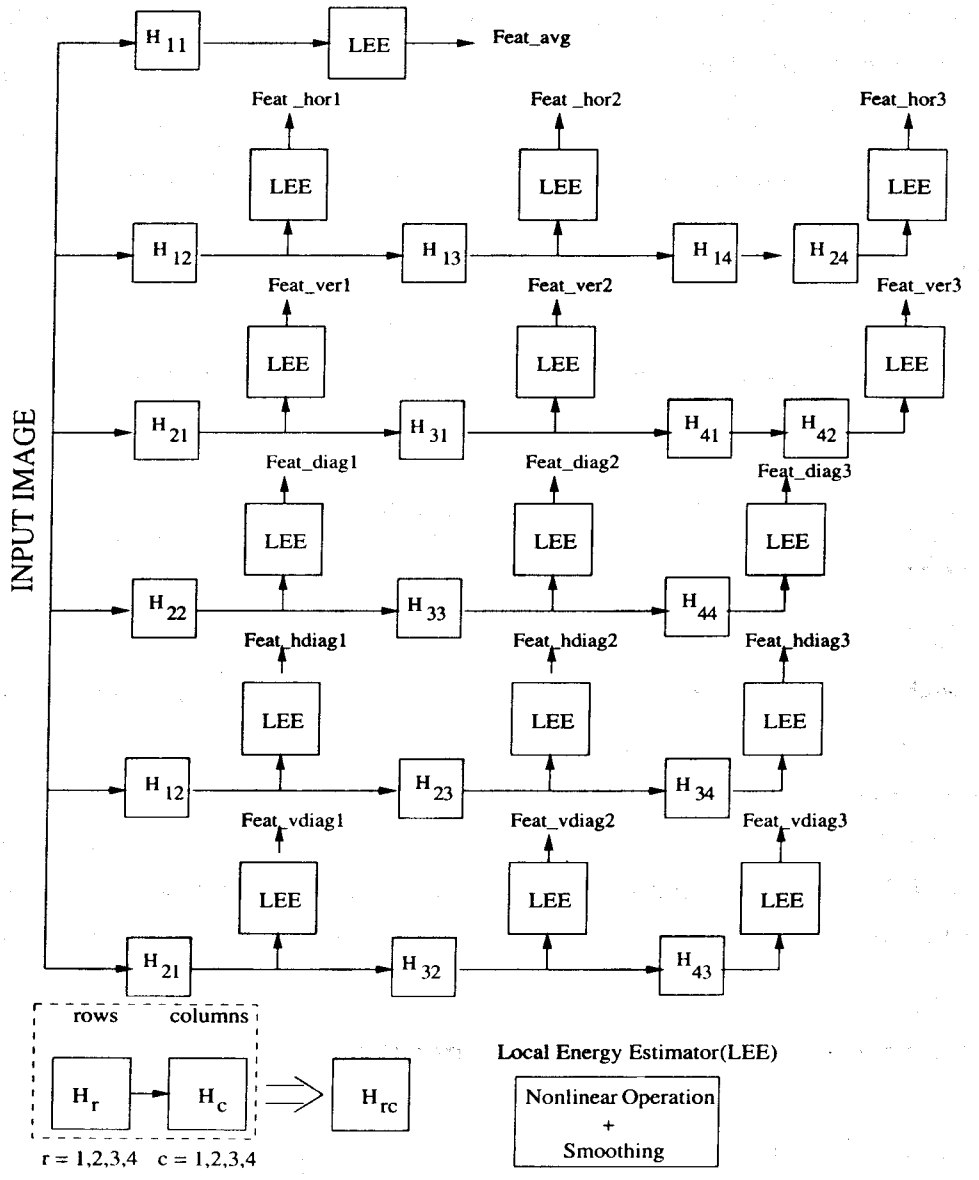


Figure 5.4: Block diagram illustrating the algorithm of our scheme

Choice of σ of the smoothing filter

The choice of the space constant σ of the averaging filter is very crucial. The problem is how to determine the size of the smoothing filter. If we want to estimate the local energy of an image with low spatial frequency the smoothing filter must have a wide unit impulse response, while narrower filter is to be used for higher frequency content image.

In the present work we set the smoothing filter size based on the measure of the spectral content of the image. *Spectral flatness measure* (*SFM*) gives a measure of the overall image activity. The spectral flatness of a digital image of size $N1 \times N2$ is defined as the ratio of the arithmetic and the geometric mean of the Fourier coefficients [87]. For two dimensional digital image this can be expressed as,

$$SFM = \frac{[\prod_{x=0}^{N1-1} \prod_{y=0}^{N2-1} \mid \hat{I}(x,y) \mid^2]^{\frac{1}{N1N2}}}{\frac{1}{N1N2} \sum_{x=0}^{N1-1} \sum_{y=0}^{N2-1} \mid \hat{I}(i,j) \mid^2} \tag{5.4}$$

$\hat{I}(i,j)$ is the $(x,y)^{th}$ Fourier coefficient of the two dimensional image. SFM has a dynamic range of $[0,1]$.

Highly active image means SFM close to 1, then the image has many edges or has predominantly high frequencies. So the image requires a smaller window for smoothing. Moderately active image has SFM of somewhat moderate value within 0 and 1. That means the image contains moderate range of frequencies and requires a moderate window size for good feature extraction. Finally an image with low SFM is lowly active and has low spectral content. It is evident that this type of image would require larger window size of the smoothing filter.

We have found experimentally that the spatial extent of the windows, for these three categories of image activities range from 11×11 to 31×31. With these choices we have worked successfully on all the test images that have been experimented. Thus we can adaptively select the size of the averaging window depending on the spectral content of the image. Our scheme is adaptive in the sense that we do not use any fixed windowing operation and hence it can accommodate diverse set of textured images as input.

5.3 Feature ranking and selection

Increasing the number of features in a certain data set does not necessarily increase the recognition/classification performance. This brings up the need for feature selection; *i.e.*, given a data set with N_F features, aim is to find the subset of n_F features for which recognition performance is (sub-)optimal. Since the number of possible subsets rises exponentially with the total number of features, evaluating all possible subsets is practically a tedious task. Therefore, we suggest a sub-optimal technique for feature selection.

The feature selection methodology comprises of two steps. Firstly, for every ith feature ($i \in N_F$) we compute a feature quality index FQI_i and then rank the features accordingly. In the second step we sequentially add one feature at a time according to their rank, starting with the best feature and compute the quality of segmentation based on some performance criterion PC.

The FQI_i used in this work is the percentage of correct classification. The β index (defined in Chapter 4, Section 4.4) is suggested as the performance criterion (PC) in this study. A plot of number of features (NOF) added according to their ranks versus the performance criterion is obtained. We expect a peak in the plot say at $NOF = n_F$, and we select the first n_F number of features from the feature set N_F for our segmentation purpose.

We also prove that the feature set so selected, indeed gives a good representation of the data by calculating the *representation entropy* of the selected feature set $\{n_F\}$ and the rest of the feature set $\{N_F - n_F\}$.

It is well known that a pattern vector $\vec{Y}$ can be approximated by n_F features from the n_F largest eigenvalues of the matrix $\mathbf{Y}$ [53]. The eigenvalues represent the variance of the features. It is evident that greater the variance of a feature more information it will contain. Ideally, we want to compress most of the information into a very few terms of the feature vector and the information conveyed by the remaining features can then be neglected. For this matter it is essential to define an information compression indicator. An obvious candidate for carrying out this task is an entropy function.

We make use of the *representation entropy* as a measure of compressibility of information [53]. Suppose we select the first n_F feature set as discussed above. We compute

the covariance matrix ($COV\{n_F\}$) of this feature set and calculate the eigenvalues λ_i of this matrix. Let $\tilde{\lambda_i}$ be a new variable,

$$\tilde{\lambda_i} = \frac{\lambda_i}{\sum_{i=1}^{n_F} \lambda_i} \tag{5.5}$$

such that $\tilde{\lambda_i}$, $i = 1, \cdots, n_F$ satisfy

$$0 \leq \tilde{\lambda_i} \leq 1 \tag{5.6}$$

and

$$\sum_{i=1}^{n_F} \tilde{\lambda_i} = 1 \tag{5.7}$$

The *representation entropy* function is defined as

$$H_R = -\sum_{i=1}^{n_F} \tilde{\lambda_i} log \tilde{\lambda_i} \tag{5.8}$$

H_R will attain a maximum when all λ_i are identical which means greatest uncertainty about the pattern vector $\vec{Y}$. While zero entropy implies that all information about $\vec{Y}$ is contained in one term of the expansion. So in our study, if the value of H_R calculated by the selected features is less than that calculated by the remaining features, we can infer that the selected feature set gives more or less a good approximation of the pattern. The feature set selected may not be the best set, but is sub-optimal. We make a trade off between optimality and complexity of computation.

5.3.1 Feature integration and Post processing

Having obtained the feature images, the main task is to integrate these feature images to produce a segmentation. We emphasize on the feature extraction (representation) part in this work. So we have used a *k-means* clustering algorithm.

After the class maps are obtained, segmentation results can be improved by post processing. The *k-means* clustering algorithm labels each pixel independently and does not take into account the high correlation between neighboring pixels. A more sophisticated algorithm would incorporate some neighborhood constraint into the segmentation process, such as relaxation labeling. So we have used median filtering to simulate the benefit of a local constraint.

5.3.2 Statistical significance of features

The performance of the different extracted features have been compared using a statistical significance test. We test whether two distributions have significantly different variances. The $F-test$ examines the hypothesis that two samples have different variances by trying to reject the null hypothesis that their variances are actually consistent [129]. The statistics F is the ratio of one variance to the other, so value either $\gg 1$ or $\ll 1$ will indicate very significant difference. The $F-test$ routine returns the value of f and its significance as *prob*. Small value of *prob* indicate that the two samples have significantly different variance.

$$prob = 2 * betai(0.5 * df2, 0.5 * df1, \frac{df2}{(df2 + df1 * f)})$$

where, $f = var1/var2$ if $var1 > var2$
$df1 = n1 - 1, \quad df2 = n2 - 1$
or, $f = var2/var1$ if $var2 > var1$
$df1 = n2 - 1, \quad df2 = n1 - 1$

$var1$ and $var2$ are the variances of the samples and $n1$ and $n2$ their sizes respectively. *betai* is the incomplete beta function.

We have tested the set of features corresponding to the two texture classes in each image for significance.

5.4 Algorithm

The texture segmentation algorithm based on the M-band wavelet decomposition is illustrated in the block diagram Fig. 5.4.

This algorithm consists of the following steps :

- The input image is first decomposed into $M \times M$ channels by wavelet analysis without downsampling as referred in section 5.2.1. In this work we have used an eight tap 4 band wavelet [17], so in all we get 16 decomposition channels as discussed in section 5.2.1, which means the image is decomposed into 16 subbands, i.e., in all 16 features can be computed. But we ignore the filtered

output corresponding to the low frequency channel H_{11}. Filtered outputs F_{hdiag_1} and F_{vdiag_1} are nothing but F_{hor_1} and F_{ver_1} respectively. So these two subbands are also ignored. In effect, now we are left with 13 frequency bands.

- These filter outputs are subjected to the nonlinear operation followed by smoothing as discussed in section 5.2.2, which then form the feature images $Feat_b$.
- The feature vector thus consist of N_F=13 feature elements, and might have redundancy. The features are ranked in order by a suitable criterion called the feature quality index FQI and then a subset $\{n_F\}$ of this ranked feature set is selected to reduce redundancy.
- We have a matrix of $n_F \times N_D$, where n_F is the number of feature elements (selected) in each vector and N_D is the total data size (the total number of pixels in the input image). The features are normalized between [0,1] along each column of the feature matrix and subjected to the clustering algorithm. This step gives us the class map corresponding to the composite texture image.

5.5 Experimental results

We have applied our texture segmentation algorithm to several two texture images, in order to demonstrate the performance of our algorithm.

5.5.1 Test Images

Here we have considered several 2-texture images $D6D17$ (Fig. 5.5(a)) and $D55D56$ (Fig. 5.6(a)) $D17D55$ (Fig. 5.9(a)) of size 256×256 each, texture images $D5D92$ (Fig. 5.11(a)), $D8D84$ (Fig. 5.10(a)), $D12D17$ (Fig. 5.12(a)) and $D9D24$ (Fig. 5.13(a)), $D17D77$ (Fig. 5.14(a)) of size 256×512 each.

All the test images discussed so far have simple texture boundaries. It is to be noted that the feature extractors extract features using some window or neighborhood, so subsequently the complexity and shape of the boundary is of major interest. The ability to cope with texture boundaries is of course an important feature and the test images should be appropriate in this respect.

We have selected some test images of size 256×256 which have complex class boundaries. Several texture images have been created by collaging sub-images extracted from various natural texture images like:
$Test-image1$ (Fig. 5.15) consisting of $D9$, $D92$, $Test-image2$ (Fig. 5.16) consisting of $D77$, $D84$, $Test-image3$ (Fig. 5.17) consisting of $D77$, $D24$, $Test-image4$ (Fig. 5.18) consisting of $D77$, $D84$, $D55$, $D24$ and $Test-image5$ (Fig. 5.19) consisting of $D77$, $D84$, $D24$.

The 256×256 natural scene image *Scene* (Fig. 5.20) consisting two classes categorized as grass/plant and sky [114].

All these images are obtained from common sources like, Brodazt album [27] and MIT Vision Texture database [114], and are considered as benchmark images for texture analysis. These images were captured using different equipment and under different conditions. All these textures are gray-scale images, and the dynamic ranges are represented by eight bits per pixel. In order to make the textures indiscriminable from the local mean gray level or local variance, *i.e.*, to abolish any variation in illumination, the images have separately been globally histogram equalized prior to being used.

5.5.2 Performance evaluation of the proposed method

Here we demonstrate the effectiveness of our methodology over several texture mosaics as described in the previous sections. We have experimented with energy and entropy features, and have found that the former shows better sensitivity over the later for these texture images. So, all the results that we have presented in this work are by using energy as textural feature. The percentage of correctly classified pixels has been used as the segmentation quality measure. Simple median filtering can be applied to the class maps as a post-processing step to improve the segmentation results. This is because median filtering is an edge preserving smoothing operation. It tries to remove small variations within a region but at the same time tries to keep the boundaries between regions intact. Several such edge preserving smoothing filtering techniques exist, but since median filtering is easy to implement we have used it for our purpose.

In case of $D6D17$ (Fig. 5.5(a)) and $D55D56$ (Fig. 5.6(a)) texture pairs, only 5 features were sufficient for successful segmentation of the images. The segmented outputs are

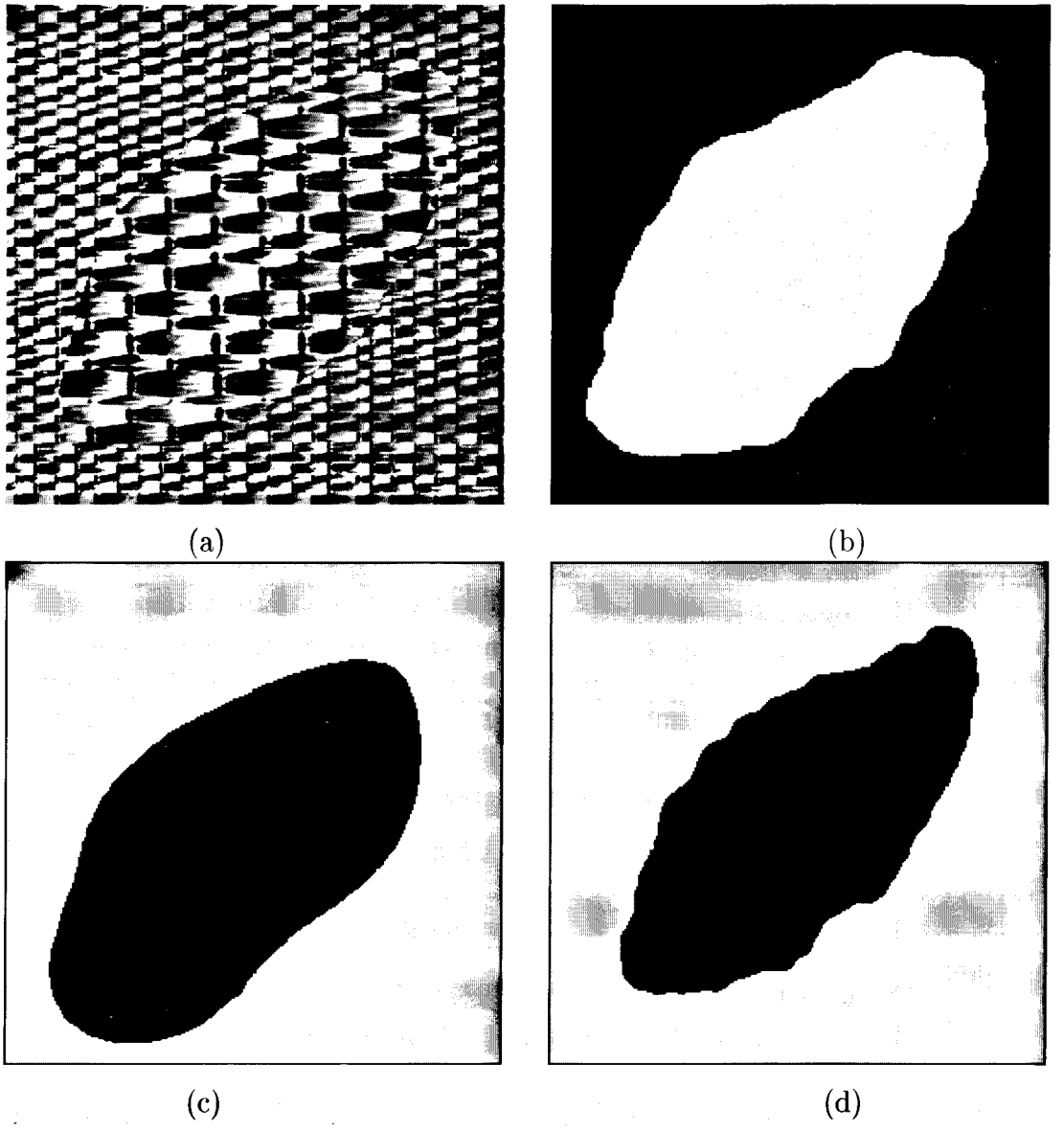


Figure 5.5: (a)Texture $D6D17$, segmented output using (b) our method (c) optimal Gabor filters by Teuner *et al.* [159] (d) optimal filters by Randen *et al.* [131]

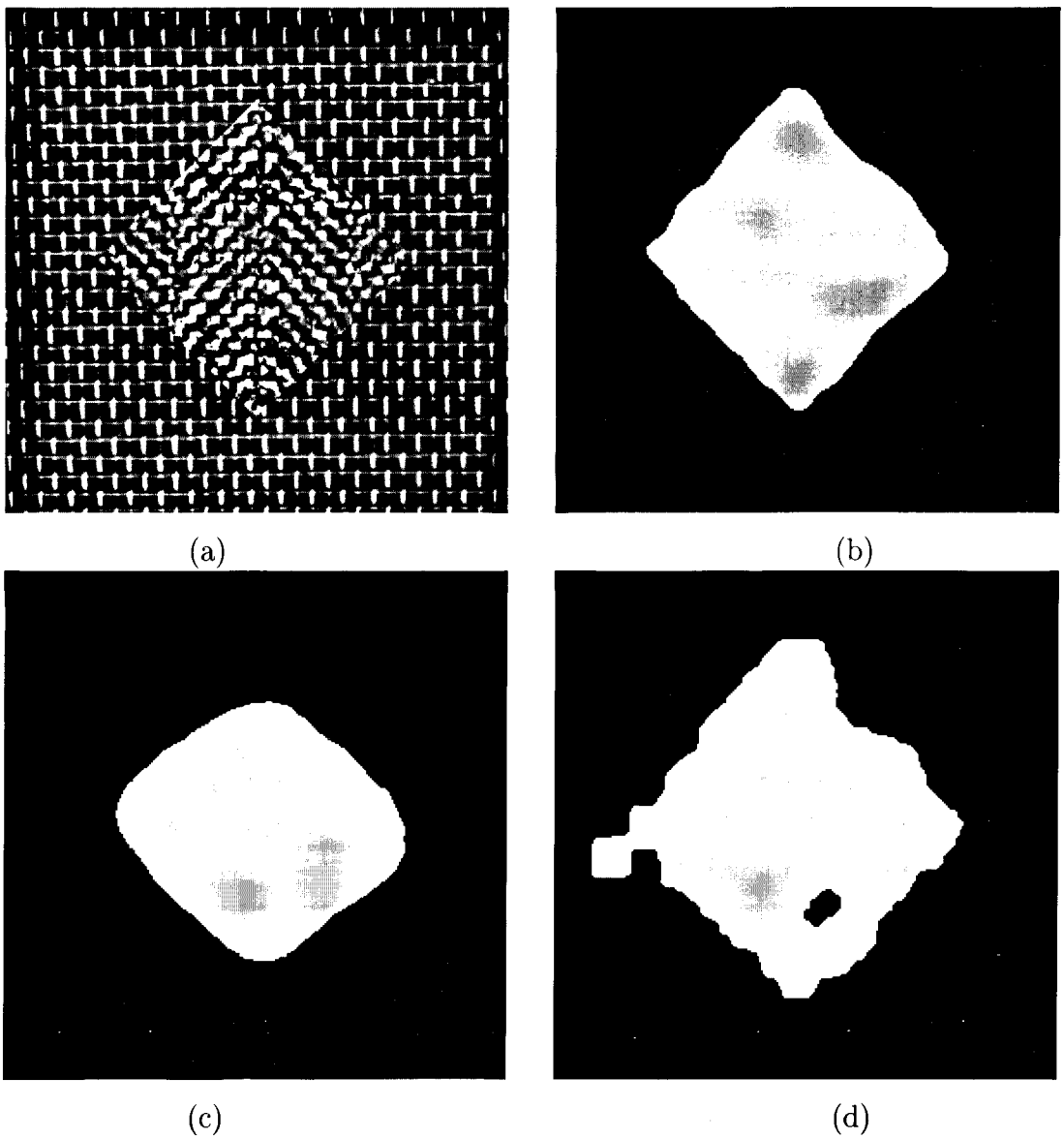


Figure 5.6: (a)Texture $D55D56$, segmented output using (b) our method (c) optimal Gabor filters by Teuner *et al.* [159] (d) optimal filters by Randen *et al.* [131]

illustrated in Fig. 5.5(b) and Fig. 5.6(b) respectively. It is to be noted that the results shown are without any post processing of the class maps. We present the results obtained by Randen *et al.* [131] and Teuner *et al.* [159] on the above two images for a qualitative comparative study. Randen *et al.* [131] proposed a technique for unsupervised optimal feature extraction and segmentation of texture images. The image was first divided into cells of equal sizes and similarity measure on the autocorrelation function for the cells were estimated. The similarity measures were used for clustering the image into clusters of cells with similar textures. Figs. 5.5(c) and 5.6(c) show the results obtained by Randen *et al.* [131] applied to the texture images $D6D17$ and $D55D56$ respectively. While, Fig. 5.5(d) and Fig. 5.6(d) illustrates the segmentation results that had been obtained by applying the approach of Teuner *et al.* [159]. The performance measures are summarized in Table 5.2. Two examples are by far not enough experimental data to judge which approach is best, but the figures clearly demonstrate that the approach we have proposed is better for these two images.

Fig. 5.9(b) shows the segmented image for the texture pair $D17D55$ (Fig. 5.9(b)). For this image we selected only 3 features, corresponding to the vertical-diagonal frequency, at the first level of resolution ($Feat_{VD_1}$), and the diagonal frequencies corresponding to the first and second level of resolutions ($Feat_{D_1}$ and $Feat_{D_2}$). As an example we show the plot of NOF versus PC (Fig. 5.7).

The distribution of features considering these 3 selected features is shown in Fig. 5.8(a). While Fig. 5.8(b) graphically represents the 3 worst features. Looking into these figures it is evident that the feature distribution of the selected ones are somewhat elongated implying presence of a large λ_i (eigenvalue) and a distinctive feature that approximates the image well. The plot of the 3 worst features on the other hand is somewhat spherical signifying that all the λ_i's are more or less identical and the features do not provide good discriminatory information.

The value of H_R considering the 3 selected features is 0.188645, while that considering the remaining 10 features is 1.44926. The percentage of correctly classified pixels using these 3 features has been found to be 98.1%. For the texture pair $D12D17$ (Fig. 5.10(a)) the segmented output is presented in Fig. 5.10(b) and for which 99.7% of pixels are being correctly classified. Fig. 5.11(a) show another texture pair $D5D92$ that we have worked on. Although the image looks simple, it was very difficult to segment using different techniques that are available in the literature. The segmented output

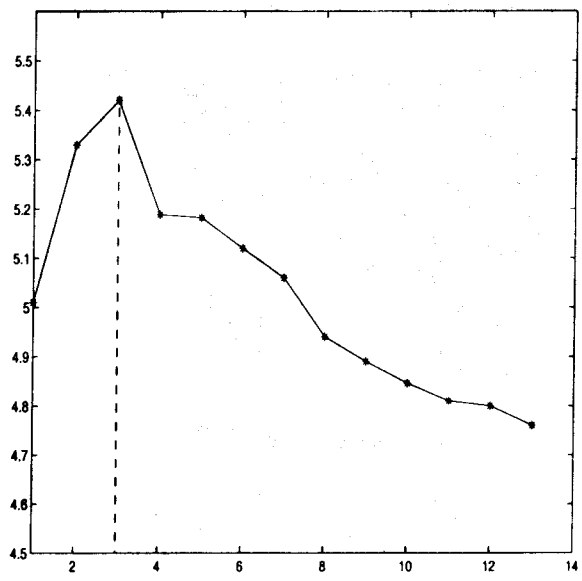
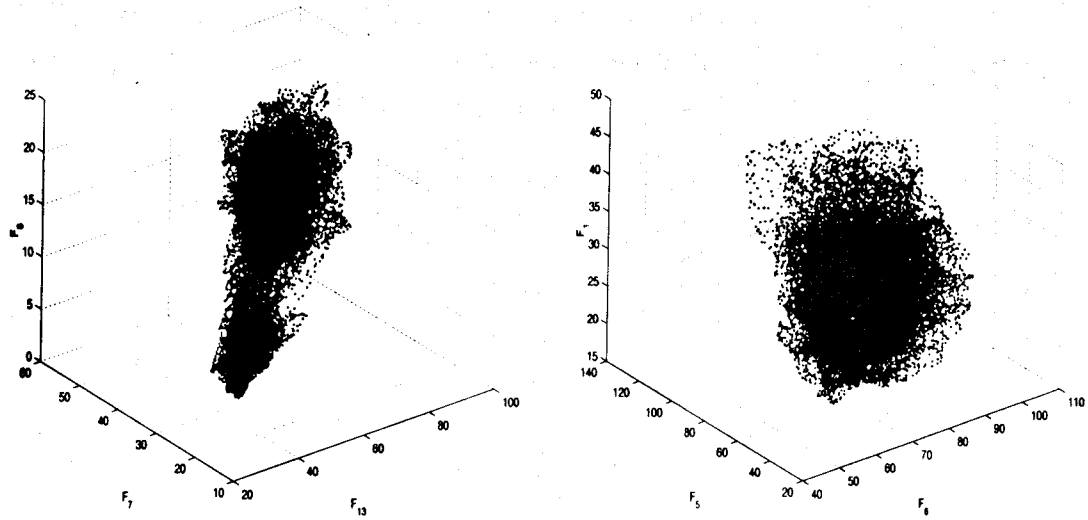
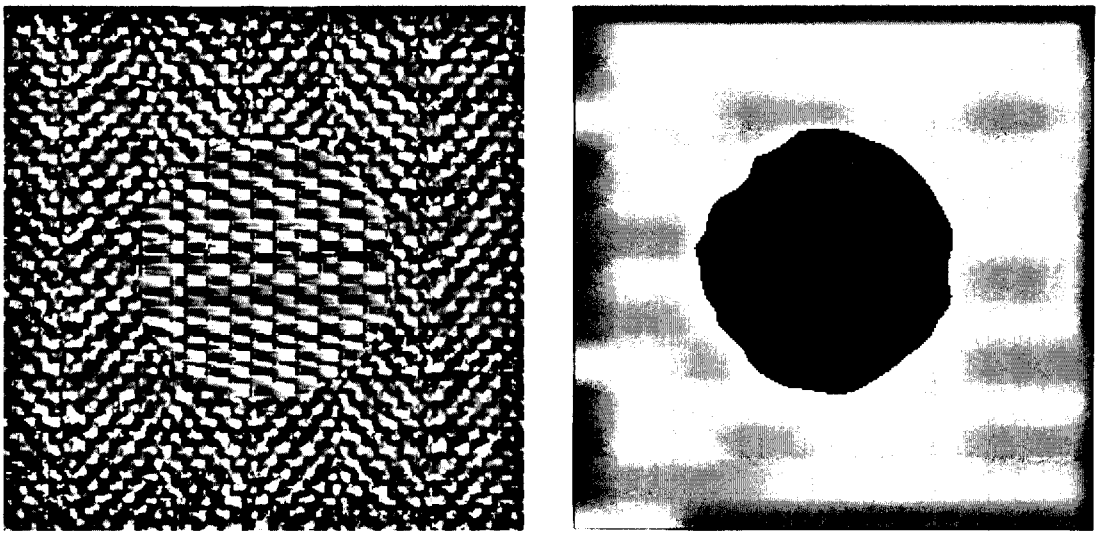


Figure 5.7: Plot of number of features and the β value



(a) (b)

Figure 5.8: Feature distribution of the (a) best set (b) worst set

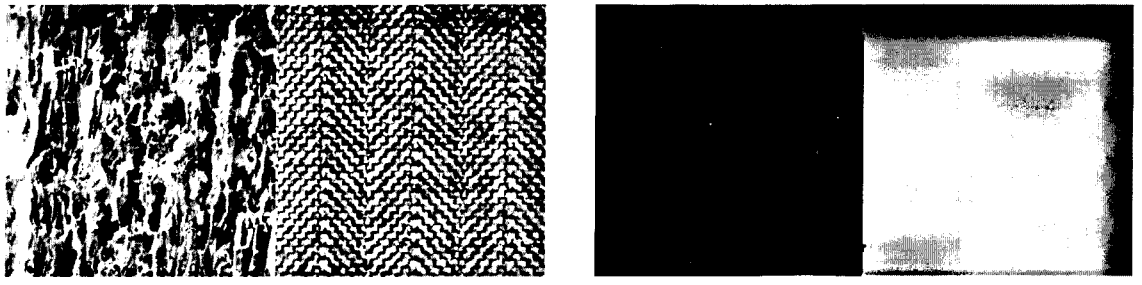


(a) (b)

Figure 5.9: (a)Texture $D17D55$ (b) Corresponding class map

that we have obtained is presented in Fig. 5.11(b) with 99.0% of correct classification.

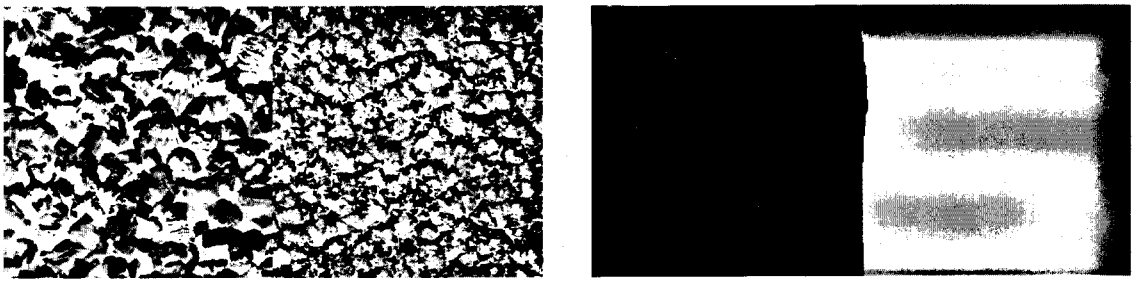
In order to prove the efficacy of our algorithm we have tested it over another texture pair $D8D84$ depicted in Fig. 5.12(a). The segmentation accuracy for this image is 99.3%, the segmented result is shown in Fig. 5.12(b). This image was tested by Jain and Karu by the back propagation designed mask. The result (Fig. 5.12(c)) obtained by their method is taken from [131] and given for a qualitative comparative study with our result.



(a) (b)

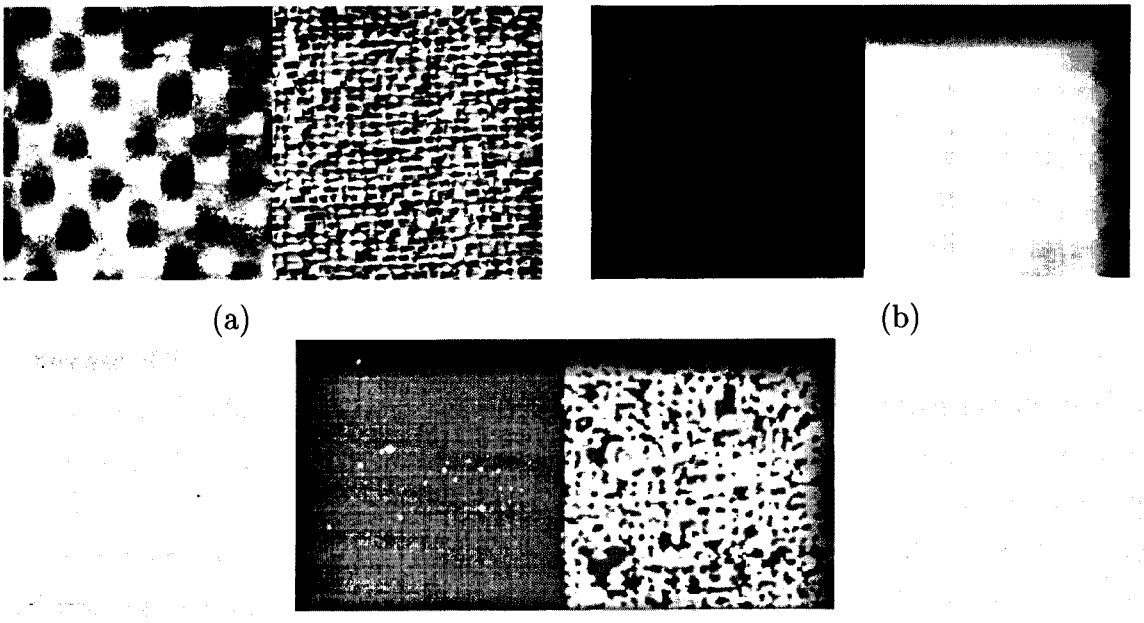
Figure 5.10: (a) Texture image $D12D17$ (b) corresponding class map

The texture mosaic $D9D24$ shown in Fig. 5.13(a) although visually difficult to dis-



(a) (b)

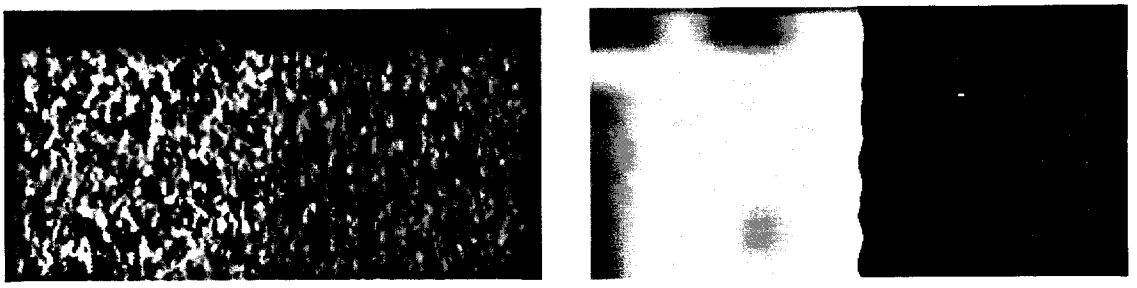
Figure 5.11: (a) Texture image $D5D92$ (b) corresponding class map



(a) (b)

(c)

Figure 5.12: (a) Texture image $D8D84$ corresponding class map using (b) our method (c) by Jain and Karu's method [131]



(a) (b)

Figure 5.13: (a) Texture image $D9D24$ (b) corresponding class map

criminate, could be identified quite correctly by our method. The result is shown in Fig. 5.13(b) with a segmentation accuracy of 99.5%.

In Fig. 5.14(b) we present the experimental result of the texture pair $D17D77$ (Fig. 5.14(a)), which is difficult to discriminate visually. Very good result have been achieved in this case also which is evident from the class map. The percentage of correctly classified pixels is found to be 99.2%. We also give the texture feature images (Figs. 5.14(c), 5.14(e) and 5.14(g)) that have been used for this particular texture pair and it is quite evident from the figures that the two texture regions are appreciably discernible on the basis of features extracted. This is also reflected from the corresponding graphs Figs. 5.14(d), 5.14(f) and 5.14(h), which give the features averaged per column of the feature images. Median filtering of size (5×5) has been applied to the class map as a post-processing step.

In the case of $Test-image1$ (Fig. 5.15(a)) and $Test-image2$ (Fig. 5.16(a)), there are 2 different texture classes. Here a particular class is not only confined to a specific region, but also distributed in various regions of the image. Moreover the boundaries between various classes, like other images, are not easily discernible. Even then the classes in the segmented image (Fig. 5.15(b) and 5.16(b)) can be identified satisfactorily. Performance measure has been found to be 98.8% and 93.8% respectively and tabulated in Table 5.2) as in the previous examples. The proposed scheme has been examined over composite textures having rotated versions of the textures $Test - image3$ (Fig. 5.17(a)). The segmentation result (Fig. 5.17(b)) is appealing in the sense that our algorithm is rotation invariant, the segmentation accuracy being found to be 97.1%. We have particularly focussed our studies on two texture images, because there are several real world images comprising of two textures, one such example being document images.

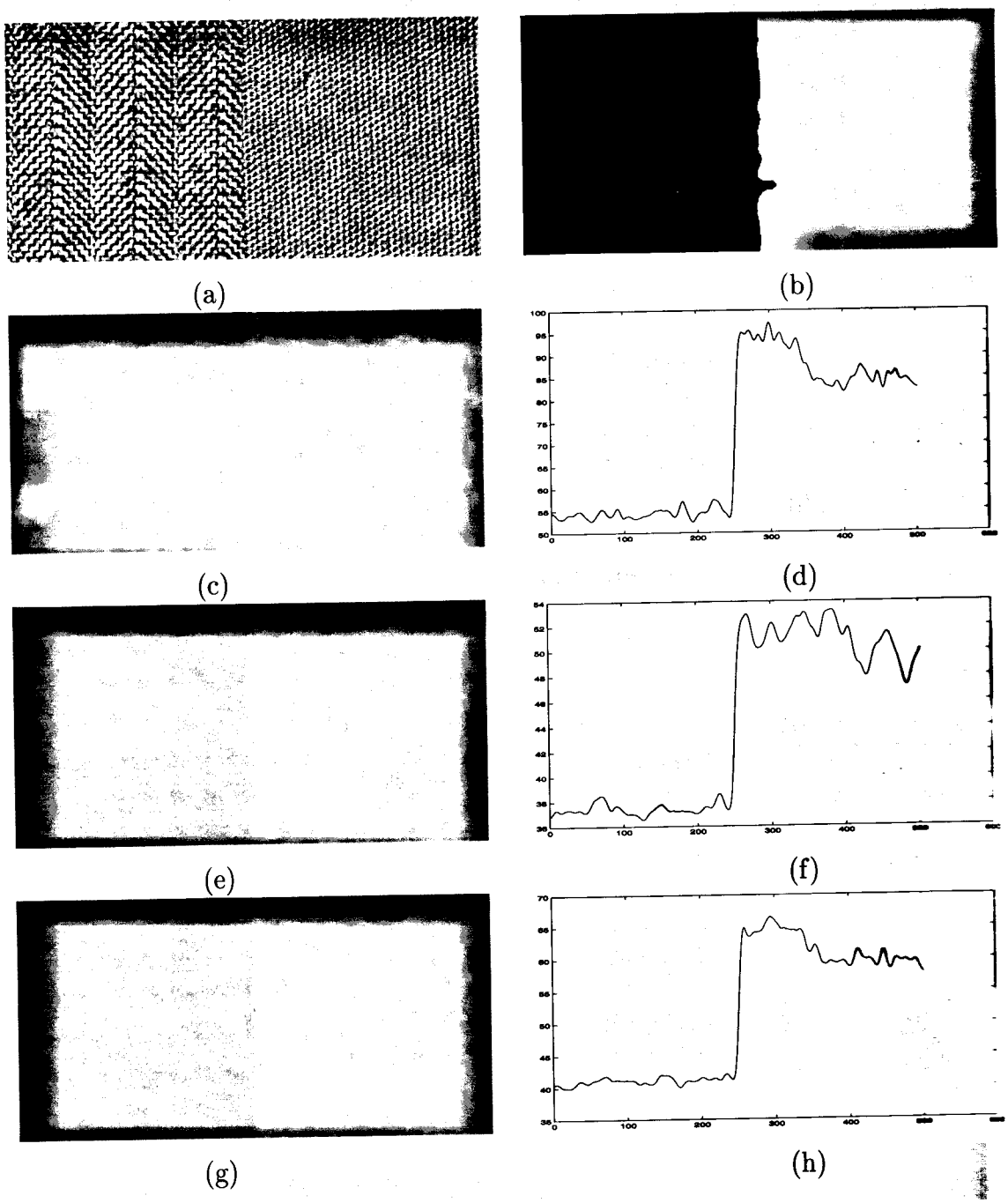
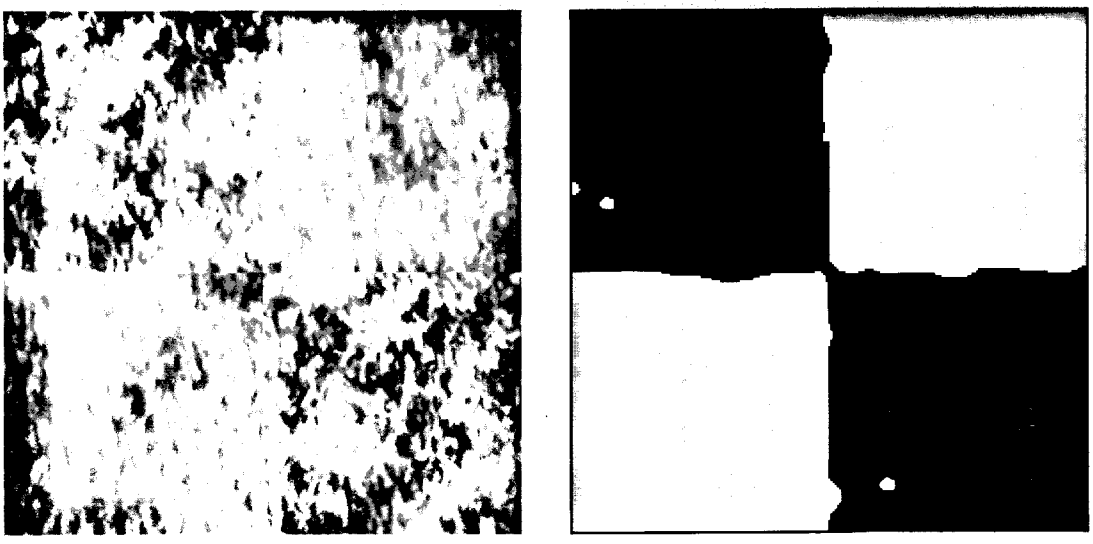
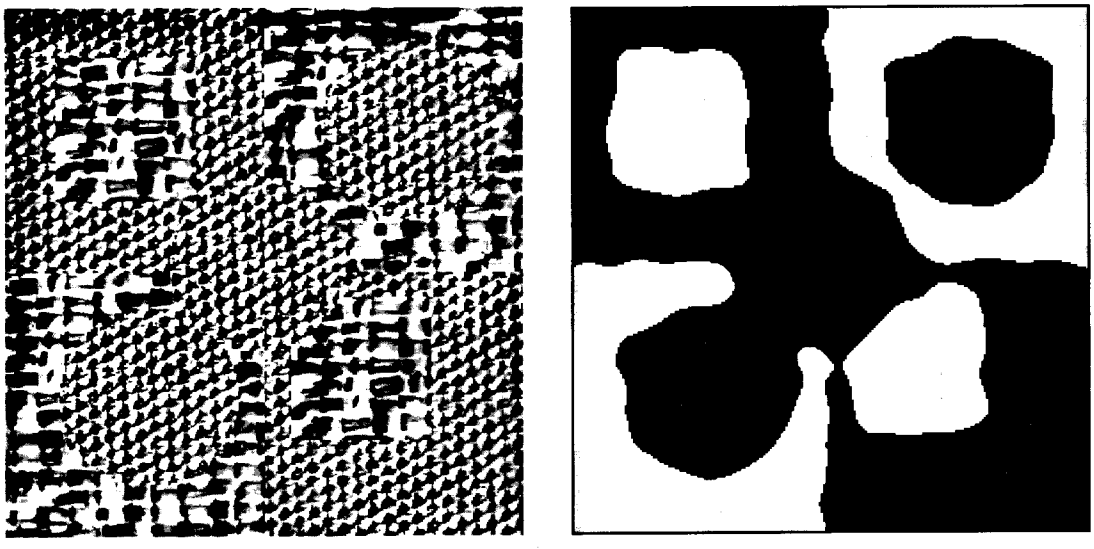


Figure 5.14: (a) Texture image $D17D77$ (b) corresponding class map, Feature image and features averaged along columns in (c)-(d) $Feat_{H_2}$ (e)-(f) correspond to $Feat_{D_2}$ and **(g)-(h)** correspond to $Feat_{HD_2}$



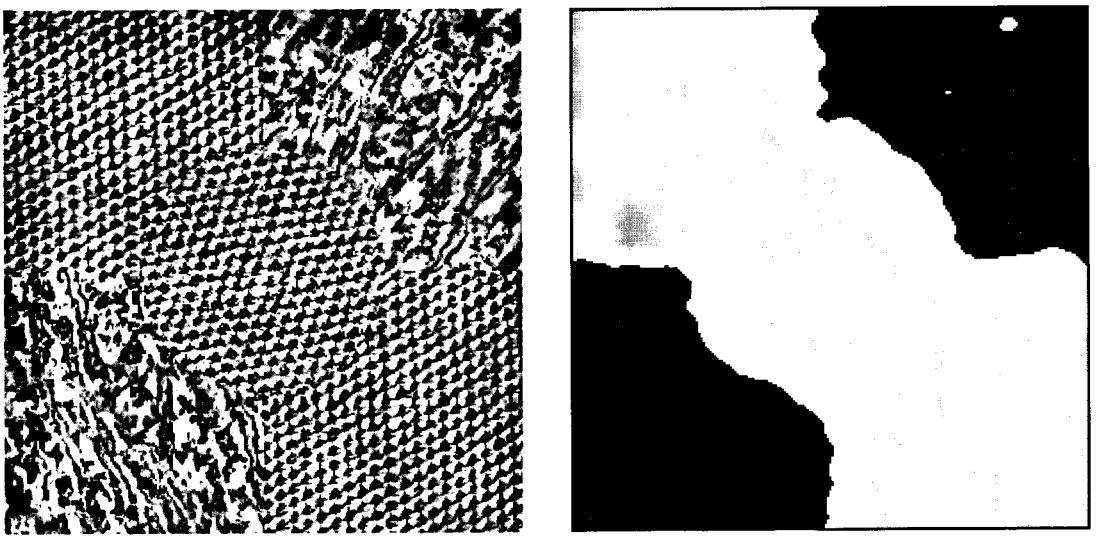
(a) (b)

Figure 5.15: (a) $Test - image1$ (b) corresponding class map



(a) (b)

Figure 5.16: (a) $Test - image2$ (b) corresponding class map



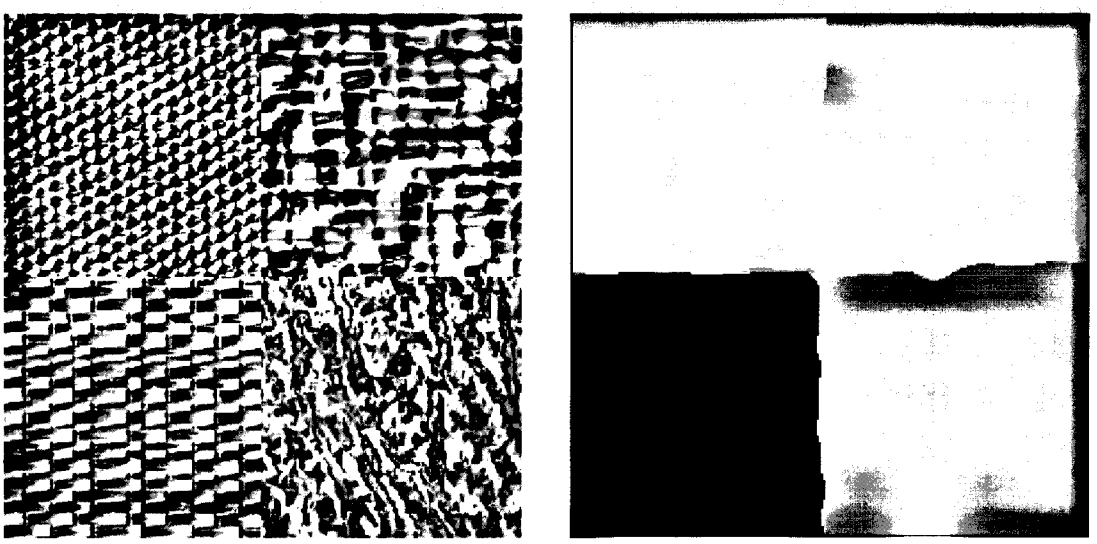
(a) (b)

Figure 5.17: (a) $Test - image3$ (b) corresponding class map

But we have also examined our algorithm over composite texture images comprising of a moderate number of textures. For texture mosaic $Test - image4$ (Fig. 5.18(a)), the segmented output is presented in Fig. 5.18(b). Note that although the image contains 4 different Brodazt textures, some of them are not distinctly identifiable visually as depicted in Fig. 5.18(a). Interestingly, the proposed methodology has been able to identify more or less all the classes. For $Test - image5$ (Fig. 5.19(a)) the classification error is more in the segmented output (Fig. 5.19(b)) as compared to Fig. 5.18(a), due to wider within class variation. The percentages of correctly classified pixels has been found to be 97.4% for $Test - image4$ and 94.8% for $Test - image5$. This ensures that our algorithm works appreciably well for moderate number of textures also.

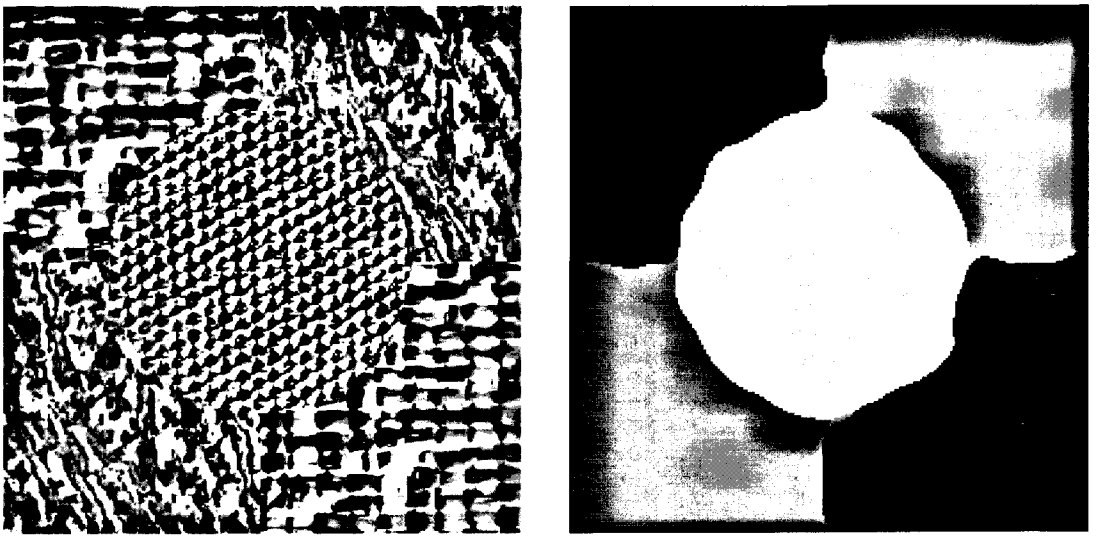
It is to be noted that for test Figs. 5.15, 5.17 and 5.18 we have not done any post processing. While post processing like median filtering or morphological operation like dilation would have definitely improved our classification performance we opted to present our result without it to show how robust our scheme is under various conditions.

Segmented result on natural scene *Scene* (Fig. 5.20(a)), is given in (Fig. 5.20(b)) considering the two regions of sky and grass/plant as two different textures.



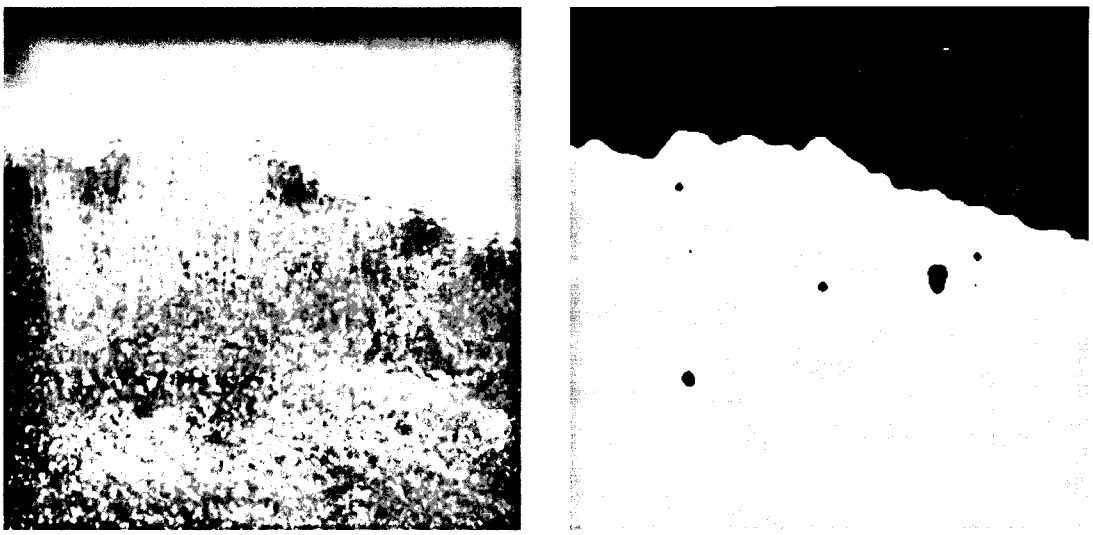
(a) (b)

Figure 5.18: (a) $Test - image4$ (b) corresponding class map



(a) (b)

Figure 5.19: (a) $Test - image5$ (b) corresponding class map



a) b)

Figure 5.20: A natural scene and the corresponding class map

Table 5.2: Performance of different test images

Test figure	Classification	Number of features
$D17D55$	98.1%	3
$D12D17$	99.7%	3
$D5D92$	99.0%	4
$D8D84$	99.3%	3
$D9D24$	99.5%	3
$Test - image1$	98.8%	3
$Test - image2$	93.8%	4
$Test - image3$	97.1%	5
$Test - image4$	97.4%	5
$Test - image5$	94.8%	5

Table 5.3: Performance evaluation for texture mosaic $D17D55$

method	*% classification*	No. of Features
Farrokhnia [60]	98.3 %	28
Dunn [56]	91.6 %	1
Randen [131] ($f32d$)	97.4 %	10
Randen [131] ($F\_2\_1\_09$))	97.0 %	10
Proposed method	98.1 %	3

5.5.3 Comparison

As part of the investigation, an extensive comparison has been made to show the superiority of the proposed methodology over various existing related algorithms. Note that several approaches to multichannel filtering for texture segmentation have been studied and a comparative performance evaluation is presented by Randen and Husφy in [132].

Farrokhnia [60] used a bank of Gabor filters with even symmetry and an octave band decomposition, the segmentation result of which is presented in Table5.5.3. Although this method gives a good result, but it applies a large set of fixed filters, and consequently results in a high computational complexity. Dunn *et al.* [56] introduced a method for finding the optimum parameters for the Gabor filters to be used for a given pair of textures and achieved good computational savings. Randen and Husφy [131] proposed a set of critically sampled FIR filter '$f32d$' [88] and IIR filter '$F\_2\_1\_09$' [85] for the purpose of segmentation. All the segmentation results for texture mosaic $D17D55$ are summarized in table 5.5.3. In this chapter, the comparison is made with respect to the number of features used for texture segmentation and the percentage of correctly classified pixels. Although so far the best result is still obtained using the bank of Gabor filters in [60], the result we have obtained is very close to it. When comparing the performance results, complexity issues should also be taken into account. The method presented in [60] gives good segmentation, but the most important drawback is its high computational complexity. An image of size 256×256 requires 28 feature images of the same size as the input image. The method introduced in [56] gives a large savings in computational cost, but with a modest degradation of performance compared to [60]. Instead of the 28 features used in [60], only 1 feature is required in [56], but this method has its weaknesses,

- The computational cost for calculating the optimal Gabor filter parameters is very high.
- To find out the optimal parameters an *a priori* knowledge about the input textures is required. So for an unsupervised approach to texture segmentation this method can not be applied.

For critically sampled filtering method in [131], there is a good savings in computational cost. But the feature dimension is 10, and also there is a degradation of the segmentation result. One major advantage of our scheme over other methods is that even though we have made use of overcomplete wavelet representation of images, which imply large feature space (i.e. 13 feature images of the same dimension as the input image), we see that only 3 features suffice for the desired segmentation and hence dimensionality of the feature space can be greatly reduced. So compared to the methods in [60] [56] [131] our method has reduced feature space and is computationally simple, while still maintaining the segmentation quality comparable to [60].

Table 3 in [132] summarizes the performance of several heuristically designed filter banks used as the feature extractors. Such as Laws filter masks [101], Dyadic Gabor filter banks [60], wavelet transform, wavelet packets [32] [98] [144], wavelet frames [164] and quadrature mirror filters (QMF) [131], used as the feature extractors. If the performance results for the texture mosaics mentioned above are taken into consideration then discrete cosine transform (DCT) appears to have produced the best result. On the other hand if the average performance for several other texture mosaics is considered then the quadrature mirror filter (QMF) and wavelet frame approaches produce better results. Most of the heuristically designed filter banks have been reported to yield successful segmentation on several test images. But most of the filter banks imply large number of features and consequently high computational complexity. Therefore optimization of the filtering operation with respect to some explicit criterion related to texture classification is desirable. Randen and Husφy [132] made extensive study on the performance of several optimized filter banks like, Gabor filter and finite impulse response (FIR) filters [88]. These were optimized following Mahalanobis and Singh (J_{MS}) [132], Unser (J_U) [132] and Fisher (J_F) [132] criterion. Table 6 in [132] gives the results of various approaches using several optimized filters, while Tables 8 and 9 summarize the performance results using full rate and critically sampled wavelet filters. In

Table 5.4: Performance evaluation for texture pair $D12D17$ and $D5D92$

Methods/filters	*Test Figures*			
	$D12D17$		$D5D92$	
	Classification	Number of features	Classification	Number of features
Heuristic	97.8%	8	97.5%	8
DCT Optimized w.r.t	97.9%	1	94.9%	1
J_U / J_F Full rate	99.5%	40	93.2%	40
QMF f32d (d) Critically sampled	99.1%	40	93.6%	40
QMF f32d (d) Proposed method M-band wavelet	99.7%	3	99.0%	3

Table 5.4 below we summarize the results achieved for the texture pairs $D12D17$ and $D5D92$ using the present method and those reported to have given the best results in [132], for a comparative study.

It is quite clear from the above discussion that we have been able to obtain high quality segmentation. For the filtering approaches, filtering and classification are the main contributors to the total complexity of the system. The heuristically designed filter (DCT) approach has low filtering complexity (short separable filter masks with fast implementation schemes) and low feature dimensionality with number of features equaling 8. The optimized filtering approach has low feature count and consequently low computational complexity. But it is not possible for an unsupervised system, because for finding out the optimal parameters a first hand knowledge about the input images have to be known. The QMF filter banks have high filtering complexities. A 40-dimensional feature extractor puts high requirements on the system [132]. Whereas in our method for both of these texture pairs the number of features were limited to 3.

5.6 Conclusion

In this work we have presented a texture segmentation algorithm using M-band wavelet transform. In the case of standard dyadic wavelet decomposition, the low frequency

band of an image is recursively split as the level of decomposition increases. Since the textures have majority of their characteristic features in the higher frequency bands, we have developed an M-band decomposition of the textured images which splits the lower as well as the higher frequency bands.

It is shown that the use of M-band wavelet decomposition of the texture image gives an efficient representation of the image in terms of frequencies in different directions and orientations at different resolutions. This representation thus facilitates an improved segmentation of the different texture regions and is also rotation invariant. The filtering and the feature extraction operations account for major computational cost, however our methodology is very simple and computationally less expensive in comparison to some related existing techniques. It has been experimentally found that for the given set of experimental data, 3 to 5 features out of the maximum of 13 (Section 5.4) features are sufficient for good quality segmentation. It is to be mentioned here that, for some methodologies reported in [132] the feature dimensionality is as high as 40. So dimensionality of the feature space is greatly reduced. Also since we have used an overcomplete wavelet representation of the textured images translational invariance can also be achieved.

Another important point to be mentioned is that **except** for the knowledge about the total number of classes present in a composite image no other *a priori* information (spatial information about the pixel labeling) about the test images are required. While all the other approaches reported in [132] are supervised and requires more *a priori* information (ground truth).

Although the major impetus in this work has been the segmentation of two textured images, we have also shown that the algorithm can work fairly well in case of images consisting of a moderate number of texture classes. The algorithm is also tested on natural scenes, the result shows its effectiveness in segmenting real life data.

We have demonstrated the effectiveness of our algorithm on composite images with a maximum of four classes. But the algorithm failed to produce satisfactory results when the number of texture classes in image is more. In the next chapter we would be discussing a multi-class texture segmentation scheme (where the number of classes can be many) using an improved methodology which is once again based on M-band wavelets.

Chapter 6

Feature Extraction using M-band Wavelet Packet Frame and their Neuro-fuzzy Selection for Multi-class texture Segmentation

6.1 Introduction

This chapter presents a multi-class texture segmentation technique using M-band wavelet packet frames [12, 6]. We have shown in the previous chapter that the decomposition scheme based on M-band ($M > 2$) wavelets, gives a better representation of a signal in terms of its constituent energies. Although the M-band wavelet decomposition results in a combination of linear and logarithmic frequency (scale) resolution, we conjecture that a further recursive decomposition of the high frequency regions would characterize the textures better. M-band wavelet packet transform recursively decomposes the high frequency bands as well besides the low frequency band. Also most of the significant information of a texture often appears in the middle frequency channels. Considering the above facts, the present study investigates the potentiality of an *M-band wavelet packet* transform for this purpose. Besides this, *frame* analysis is employed to accomplish translation invariance. This transform corresponds to a general tree-structured filter bank. Features are derived from these filter outputs. The decomposition scheme employing M-band wavelet packet leads to a large number of independent bases. It is usually unnecessary to decompose all subbands in each scale to achieve the full decomposition tree. Hence, an appropriate means of selecting the significant bases is required.

Even if decomposition of a full tree is avoided by selection of appropriate bases, the number of features generated are still large, which incurs redundancy, so it is necessary to reduce the number of redundant features. Different useful classical techniques for feature selection are available in [53]. Several algorithms for multiscale basis and feature selection are reported in the literature [84, 35, 39, 60]. The idea of designing local basis using class separability criteria has been studied by Saito and Coifman [144]. Other techniques for best basis selection are given in [47, 108]. Some recent attempts made for feature selection in the framework of artificial neural networks (ANN) include [151, 52, 128, 145]. Depending on whether the class information of the samples is known or not, these methods are classified under supervised or unsupervised modes. Recently, Etemad and Chelappa [59] have employed a class separability measure for selection of a suitable set of basis functions, in connectionist framework, from a tree structured wavelet packet basis set.

Therefore, the proposed methodology has two parts. In the first part, we develop a

computationally efficient and adaptive technique for finding out an appropriate choice of a tree-structured M-band wavelet packet basis, to avoid a full decomposition. The selection of basis is based on some criterion of a maximum measure of textural cue, in order to locate only those frequency channels (subbands) which convey dominant information, and to decide whether further sub-division of a particular subband is needed or not. With this transform, we are able to zoom into any desired frequency channels for further decomposition. The textural measure based on the statistical parameters, *viz.*, energy and entropy, are then extracted from each of the subbands. The energy measures textural uniformity, *i.e.*, extent of repetitions of pixel pairs while entropy measures the disorder of an image.

In the second part, we focus on a neuro-fuzzy approach under unsupervised learning, for selecting an optimal set of features, from those obtained after the selection of the basis. For this purpose, we have developed a neuro-fuzzy algorithm which is a modified version of Pal *et al.* [122]. This modification enables us to select the textural features more efficiently as compared to the method in [122].

Note that neuro-fuzzy computing [125, 123] which integrates the merits of fuzzy set theory and ANN, enables the feature selection process artificially more intelligent. Incorporation of fuzzy set theory helps us to deal with uncertainties in different tasks of pattern recognition system, arising from deficiencies (*e.g.*, vagueness, incompleteness etc.) in information, in an efficient manner. ANNs, having the capability of fault tolerance, adaptivity and generalization, and scope for massive parallelism, are widely used in dealing with learning and optimization tasks.

The methodology for neuro-fuzzy feature selection involves formulation of a layered network for minimization of a fuzzy feature evaluation index. The index for a set of features is defined in terms of membership values denoting the degree of similarity between two patterns both in the original and the transformed spaces. It is such that, for a set of features, the lower is its value, the higher is the importance of that set in characterizing/discriminating various clusters. A set of weighting coefficients is used to denote the degree of importance of the individual features in characterizing/discriminating different clusters and to provide flexibility in modeling various clusters. Minimization of the evaluation index through unsupervised learning of the network determines the optimum weighting coefficients providing an order of importance of the features individually.

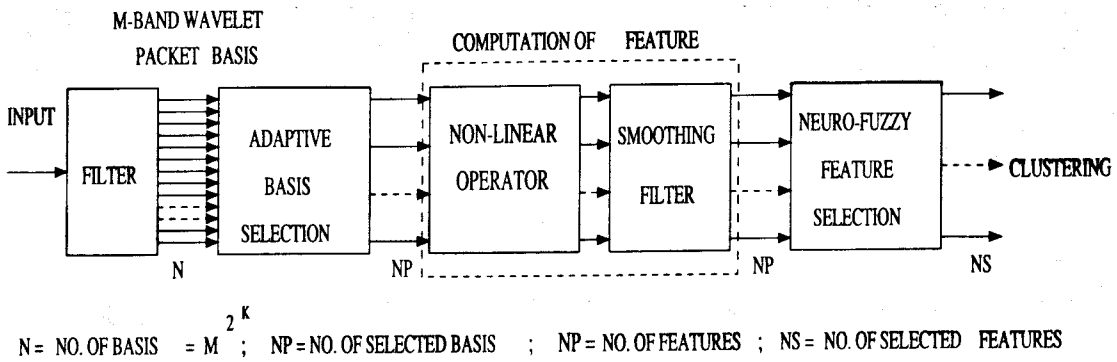


Figure 6.1: Experimental setup

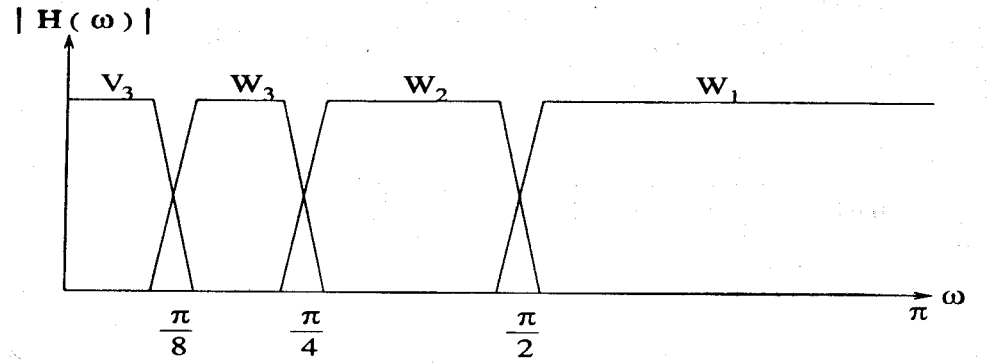
The next section discusses about the filtering technique for extraction of features used in our investigation.

6.2 Extraction of Multiscale Wavelet Features

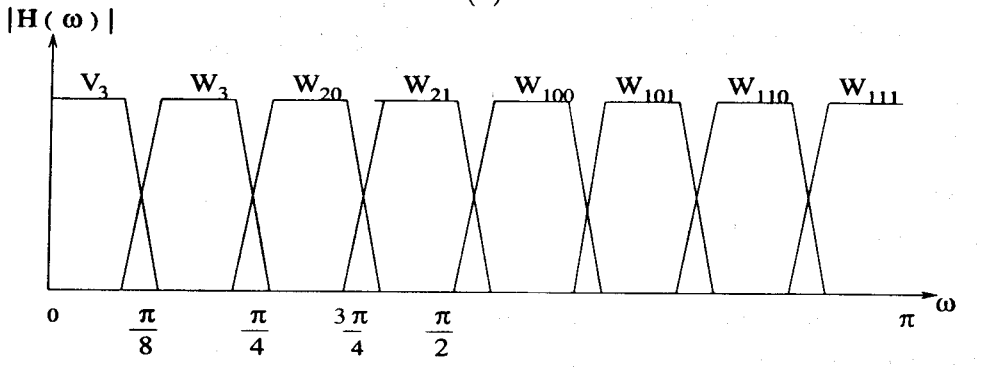
This section provides the methodology for extracting multiscale wavelet features of a texture image. It involves M-band wavelet packet filtering of an input image followed by adaptive basis selection. Subsequently, features are computed from this set of selected basis by using a nonlinear operator and smoothing filter. These features are then evaluated and selected using a neuro-fuzzy algorithm (described in Section 6.3). The entire methodology is depicted in Fig. 6.1.

6.2.1 M-band wavelet packet filters

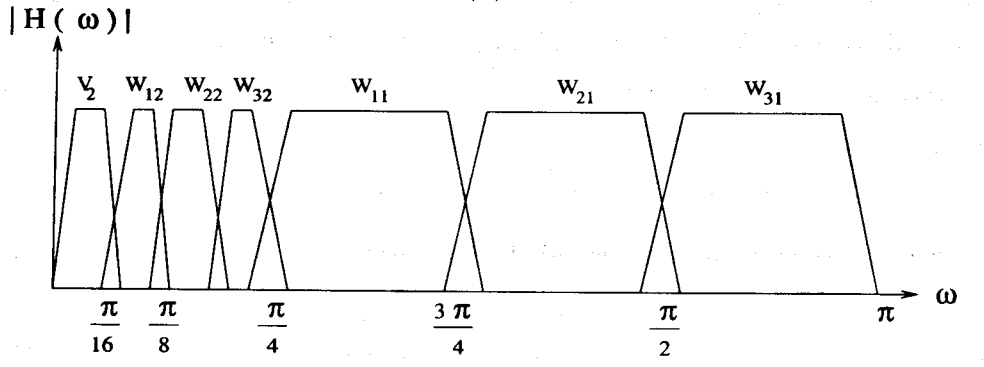
The filter bank in essence is a set of bandpass filters which select frequency and orientation. In the filtering stage, we make use of the same M-band (M=4) wavelet used in the previous chapter. The discrete M-band wavelet packet transform (DMbWPT) used in this work results in a tree structured multi-band extension of the M-band wavelet transform. Thus a finer and adjustable resolution at high frequencies is allowed as compared to the 2-band case. Fig. 6.2 depicts the frequency responses for 2-band wavelet, 2-band wavelet packet, 4-band wavelet, and 4-band wavelet packet filter banks for a better understanding of the various decomposition schemes.



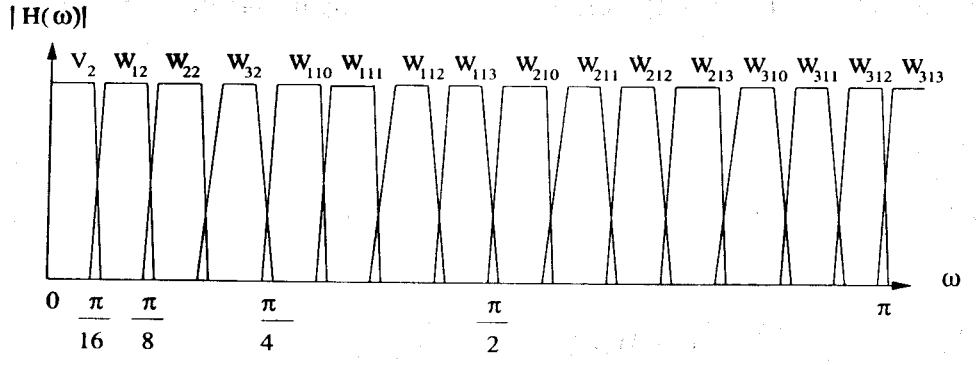
(a)



(b)



(c)



(d)

Figure 6.2: Frequency responses of (a) classic (M=2) band wavelet (b) 2-band wavelet packet (c) $M = 4$-band wavelet (d) $M = 4$-band wavelet packet filter banks

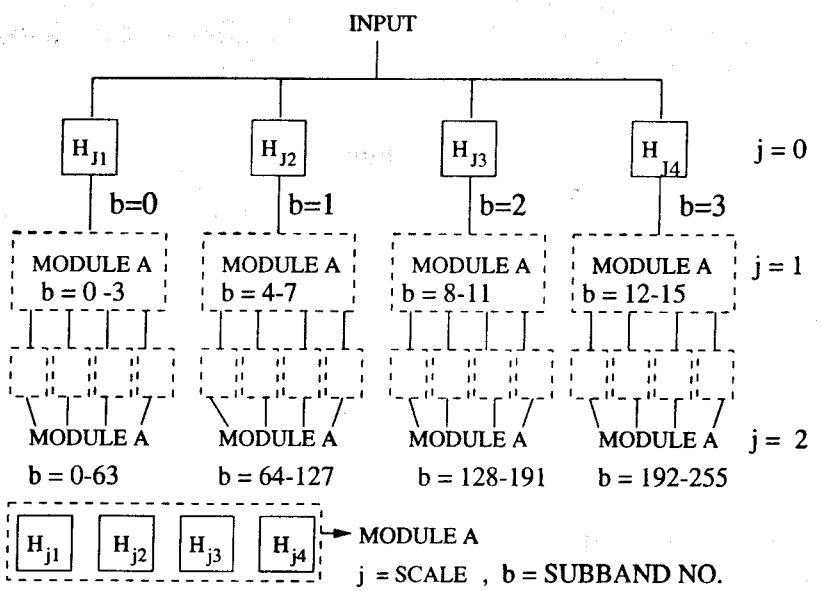


Figure 6.3: Tree structure of DM-bWPF transform and related indices

All the classical decomposition methods described in Section 2.9, require a downsampling by a factor M (where $M = 2$ for standard wavelet) at each scale. But these decompositions are not translation invariant. As mentioned earlier, a discrete M-band wavelet packet frame (DMbWPF) is used, which is similar to DMbWPT, except that no downsampling occurs between scales (levels of decomposition).

Fig. 6.3. shows a general tree structure of discrete M-band wavelet packet frame decomposition. Scale $j = 0$ corresponds to the highest resolution of the image, that is the image itself before decomposition.

The filter responses in the frequency domain $H_{j,r}(\omega)$ (for $r = 1, \ldots, 4$) and at level j are generated as follows

$$H_{j,r} = H_{0,r}(M^j\omega) \quad \text{for} \quad r = 1, \ldots, 4. \tag{6.1}$$

Let, $\hat{I}_b^j(\omega)$ be the Fourier transform of the input signal $I(x)$ for subband (frequency channel) b at decomposition level j. For $0 \leq b \leq 4^j - 1$ and $r = 1, 2, 3, 4$, we have

$$\hat{I}^{j+1}_{4b+(r-1),r}(\omega) \;=\; H_{j,r}(\omega)\hat{I}^j_{b,r}(\omega). \tag{6.2}$$

From the filter bank theoretic point of view [167], this corresponds to a filter bank with channel filters $\{filt^j_{b,r}(\omega) \quad | r = 1, \ldots, 4.\}$. $filt^j_{b,r}(\omega)$ are given by the following

recursive relation

$$\begin{aligned} filt^{0}_{0,r}(\omega) &= H_{0,r}(\omega), \\ &\text{and} \\ filt^{j+1}_{4b+(r-1),l}(\omega) &= H_{j,r}(\omega) filt^{j}_{b,r}(\omega), \\ &= H_{0,r}(M^{j}\omega) filt^{j}_{b,r}(\omega). \end{aligned} \tag{6.3}$$

For images we simply use tensor product extension for which the channel filters are written as

$$filt^{j}_{b,r\ [x\times y]}(\omega_x, \omega_y) = filt^{j}_{b,r\ [x]}(\omega_x) filt^{j}_{b,r\ [y]}(\omega_y)$$

At scale $j = 1$, the image is first decomposed into $M \times M$ channels using all the filters $H_{0,r}$ with $r = 1, 2, 3, 4$, and without downsampling. The process is repeated for each of the subbands for subsequent scales. Fig. 6.3 shows a general tree structure of discrete M-band wavelet packet frame decomposition. Module-A in Fig. 6.3 comprises of all the filters $H_{0,r}$ with $r = 1, 2, 3, 4$.

6.2.2 Adaptive basis selection

The objective of wavelet decomposition of a signal/image is to compact most of the energy in a fewer number of subbands. An appropriate way of obtaining the wavelet transform for texture feature extraction is to detect the most significant frequency channels and then to decompose them further. This leads naturally to a tree structured wavelet transform. It is also usually redundant to decompose all the subbands in each scale to achieve the full tree of decomposition. Moreover, an M-band wavelet packet decomposition gives M^{2^J} number of bases, for a decomposition depth of J. It is quite evident that an exhaustive search to determine the optimal basis from this large set is computationally very expensive.

In order to find out a suitable basis without going for a full decomposition, we propose an adaptive decomposition algorithm using a criterion of maximum textural measures extracted from each of the subbands. Then the most significant subbands are identified and it is decided whether further decomposition of the particular channel would generate more information or not. This search is computationally efficient and enables one to zoom into any desired frequency channel for further decomposition [6].

For this purpose, the image is first decomposed into $M \times M$ channels using the 2D

M-band wavelet transform without downsampling (oversampled) as shown in Fig. 6.3. Energy for each subband is then computed. Among these subbands, those for which energy values exceed $\epsilon_1\%$ of the energy of the parent band, are considered and decomposed further. We further decompose a subband if its energy value is more than some $\epsilon_2\%$ of the total energy of all the subbands at the current scale. The analysis is performed upto the second level of decomposition and this results in a set of wavelet packet bases. These bases corresponding to different resolutions are assumed to capture and characterize effectively different scales of texture of the input image. Empirically we have seen that a value of $\epsilon_1 = 2 - 5\%$ and $\epsilon_2 = 50\%$ are good choices for the images we have considered here. This simple top-down splitting technique performs well for most images.

6.2.3 Local estimator of textural measure

After selection of the significant bases, a local estimator is applied to each subbands as discussed earlier (Section 4.3.2). A textural feature is derived from the statistical parameter from each of the subbands. In this study two different measures for texture segmentation are examined and compared. These measures are *energy* and *entropy*. We have used modulus operator as the nonlinearity. *Average absolute deviation* from the mean has been used as a generalized energy definition in this work.

For a subband image $F_b(x, y)$of subband number b, the local energy $Eng_b(x, y)$ around the x, y^{th} pixel can be formally defined as

$$Eng_b(x, y) = \frac{1}{R} \sum_{x'=1}^{w_l} \sum_{y'=1}^{w_l} \mid (F_b(x', y') - \overline{F}_b(x, y)) \mid, \tag{6.4}$$

and the local entropy $Ent_b(x, y)$

$$Ent_b(x, y) = \sum_{x'=1}^{w_l} \sum_{y'=1}^{w_l} (F_b(x', y') log(F_b(x', y')) \tag{6.5}$$

where w_l is the window size and $R = w_l \times w_l$. The term $\overline{F}_b(x, y)$ stands for the mean around the $(x, y)^{th}$ pixel and $F_b(x, y)$ is a subband image.

This step is succeeded by a Gaussian low pass (smoothing) filter $h_G(x, y)$ to get a feature image. Formally, the feature image $Feat_b(x, y)$ corresponding to subband image

$F_b(x, y)$ is given by

$$Feat_b(x, y) = \sum_{(x',y') \in G_{xy}} \Gamma(F_b(x', y') h_G(x - x', y - y'))$$

where $\Gamma(\cdot)$ gives the energy (entropy) measure and $G_{x,y}$ is a $G \times G$ window centered at pixel with coordinates (x, y).

An important issue in this regard is the size w_l of the local window. After a number of experiments we have found that the choice of the local window size is very crucial to find proper features. In an image with patterns of different texel sizes we have to choose a suitable window size. If we choose a larger window for the largest texel size we introduce more uncertainty on the boundaries regions. This problem can be solved if the effective window size changes with the level of resolution. These various window sizes will catch textures with different texel sizes. For larger texels it is better to choose larger local window size and for smaller texels it should be kept smaller.

Adaptive window for assigning feature values

To improve the reliability of the feature values around texture boundaries we have utilized an adaptive window to assign features. The adaptability is in terms of a simple decision rule on the area of which we should calculate the features. We consider the variance of the wavelet features in five neighborhood windows of the pixel under study. One of the areas is the area which is covered by the local window centered at the pixel under study. Four other areas are the areas covered by the local windows in which the concerned pixel is on the top right corner, top left corner, bottom right corner and bottom left corner. The window size in all five cases is fixed. We find the window with the minimum variance and assign the local statistics of that window as the feature of the pixel under consideration.

6.3 Selection of Wavelet Features using a Neuro-fuzzy Method and Segmentation

The wavelet features extracted in the previous section are evaluated and then selected using a neuro-fuzzy feature selection under unsupervised learning. The method is a

modification of an earlier one [122]. This modification enables one to handle large data sets in an efficient manner. Note that for an image a large number of pattern vectors are generated (as described in Section 6.2). These selected features are then used for the purpose of segmenting multi-texture images.

6.3.1 Fuzzy feature evaluation index and membership function

The neuro-fuzzy feature selection method [122] involves formulation of a fuzzy feature evaluation index followed by its minimization in connectionist framework. The feature evaluation index for a set of transformed features is defined as

$$E = \frac{2}{s(s-1)} \sum_{p} \sum_{q \neq p} \frac{1}{2} [\mu_{pq}^{T}(1 - \mu_{pq}^{O}) + \mu_{pq}^{O}(1 - \mu_{pq}^{T})]. \tag{6.6}$$

Here μ_{pq}^{O} and μ_{pq}^{T} are the degree that both the pth and qth patterns belong to the same cluster in the n-dimensional original feature space, and in the n'-dimensional ($n' \leq n$) transformed feature space respectively. μ values determine how similar a pair of patterns are in the respective features spaces. That is, μ may be interpreted as the membership value of a pair of patterns belonging to the fuzzy set "similar". s is the number of samples on which the feature evaluation index is computed.

The index has the following characteristics.
(i) For $\mu_{pq}^{O} < 0.5$ as $\mu_{pq}^{T} \rightarrow 0$, E decreases. For $\mu_{pq}^{O} > 0.5$ as $\mu_{pq}^{T} \rightarrow 1$, E decreases. In both the cases, the contribution of the pair of patterns to the evaluation index E becomes minimum (= 0) when $\mu_{pq}^{O} = \mu_{pq}^{T} = 0$ or 1. (ii) For $\mu_{pq}^{O} < 0.5$ as $\mu_{pq}^{T} \rightarrow 1$, E increases. For $\mu_{pq}^{O} > 0.5$ as $\mu_{pq}^{T} \rightarrow 0$, E increases. In both the cases, the contribution of the pair of patterns to E becomes maximum (= 0.5) when $\mu_{pq}^{O} = 0$ and $\mu_{pq}^{T} = 1$, or $\mu_{pq}^{O} = 1$ and $\mu_{pq}^{T} = 0$. (iii) If $\mu_{pq}^{O} = 0.5$, the contribution of the pair of patterns to E becomes constant (= 0.25), *i.e.*, independent of μ_{pq}^{T}.

Therefore, the feature evaluation index decreases as the membership value representing the degree of belonging of pth and qth patterns to the same cluster in the transformed feature space tends to either 0 (when $\mu^{O} < 0.5$) or 1 (when $\mu^{O} > 0.5$). In other words, the feature evaluation index decreases as the decision on the similarity between a pair of patterns (*i.e.*, whether they lie in the same cluster or not) becomes more and more

crisp. This means, if the inter-cluster/intra-cluster distances in the transformed space increase/decrease, the feature evaluation index of the corresponding set of features decreases. Therefore, our objective is to select those features for which the evaluation index becomes minimum; thereby optimizing the decision on the similarity of a pair of patterns with respect to their belonging to a cluster.

The membership function μ_{pq} in a feature space, satisfying the characteristics of E (6.6), may be defined as [122]

$$\begin{aligned} \mu_{pq} &= 1 - \frac{d_{pq}}{D} \quad if\ d_{pq} \leq D, \\ &= 0, \qquad\quad otherwise. \end{aligned} \tag{6.7}$$

d_{pq} is a distance measure which provides similarity (in terms of proximity) between the pth and qth patterns in the feature space. Note that, the higher the value of d_{pq}, the lower is the similarity between pth and qth patterns, and *vice versa*. D is a parameter which reflects the minimum separation between a pair of patterns belonging to two different clusters. When $d_{pq} = 0$ and $d_{pq} = D$, we have $\mu_{pq} = 1$ and 0, respectively. If $d_{pq} = \frac{D}{2}$, $\mu_{pq} = 0.5$. That is, when the distance between the patterns is just half the value of D, the difficulty in making a decision, whether both the patterns are in the same cluster or not, becomes maximum; thereby making the situation most ambiguous. Note that the computation of μ_{pq} in (6.7) does not require the information on class label of the patterns.

The term D in (6.7) may be expressed as $D = \alpha d_{max}$ where d_{max} is the maximum separation between a pair of patterns in the entire feature space, and $0 < \alpha \leq 1$ is a user defined constant. α determines the degree of flattening of the membership function (6.7). The higher the value of α, more will be the degree, and *vice-versa*.

The distance d_{pq} (6.7) can be defined in many ways. Considering weighted Euclidean distance, we have

$$\begin{aligned} d_{pq} &= [\sum_i w_i^2 (x_{pi} - x_{qi})^2]^{\frac{1}{2}}, \\ &= [\sum_i w_i^2 \chi_i^2]^{\frac{1}{2}}, \quad \chi_i = (x_{pi} - x_{qi}), \end{aligned} \tag{6.8}$$

where $w_i \in [0, 1]$ represents weighting coefficient corresponding to ith feature. The terms x_{pi} and x_{qi} are values of ith feature (in the corresponding feature space) of pth and qth patterns, respectively. d_{max} is defined as

$$d_{max} = [\sum_i (x_{maxi} - x_{mini})^2]^{\frac{1}{2}}, \tag{6.9}$$

where x_{maxi} and x_{mini} are the maximum and minimum values of the ith feature in the corresponding feature space.

The membership value μ_{pq} is dependent on w_i. The values of w_i (< 1) make the μ_{pq} function of (6.7) flattened along the axis of d_{pq}. The lower the value of w_i, the higher is extent of flattening. In the extreme case, when $w_i = 0$, $\forall i$, $d_{pq} = 0$ and $\mu_{pq} = 1$ for all pair of patterns, *i.e.*, all the patterns lie on the same point making them indiscriminable.

The weight w_i in (6.8) reflects the relative importance of the feature x_i in measuring the similarity (in terms of distance) of a pair of patterns. The higher the value of w_i, the more is the importance of x_i in characterizing a cluster or discriminating various clusters. $w_i = 1$ (0) indicates most (least) importance of x_i.

As mentioned earlier, our objective is to minimize the evaluation index E (6.6) which involves the terms μ^O and μ^T. The computation of μ^T requires (6.7)–(6.9), while μ^O needs these equations with $w_i = 1$, $\forall i$. Therefore, the evaluation index E (6.6) is a function of $\mathbf{w}$, if we consider ranking of n features in a set. The problem of feature selection/ranking thus reduces to finding a set of w_is for which E becomes minimum; w_is indicating the relative importance of x_is. The task of minimization as in [122], is performed using gradient-descent technique in a connectionist framework under unsupervised mode. This is described below.

6.3.2 Connectionist model

The network (Fig. 6.4) consists of an input, a hidden and an output layer [122]. The input layer consists of a pair of nodes corresponding to each feature, *i.e.*, the number of nodes in the input layer is $2n$, for n-dimensional (original) feature space. The hidden layer consists of n number of nodes which compute the part χ_i^2 of (6.8) for each pair of patterns. The output layer consists of two nodes. One of them computes μ^O, and the other μ^T. The feature evaluation index E (A.4) is computed from these μ-values off the network.

Input nodes receive activations corresponding to feature values of each pair of patterns. A jth node in the hidden layer is connected only to an ith and $(i+n)$th input nodes via connection weights $+1$ and -1, respectively, where $j, i = 1, 2, \ldots, n$ and $j = i$. The

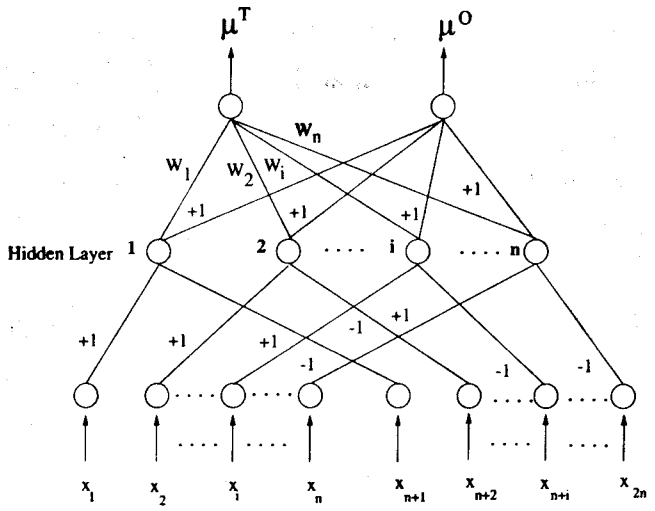


Figure 6.4: A schematic diagram of the neural network model for feature selection.

output node computing μ^T-values is connected to a jth node in the hidden layer via connection weight W_j $(= w_j^2)$, whereas that computing μ^O-values is connected to all the nodes in the hidden layer via connection weights $+1$ each.

During learning, each pair of patterns are presented at the input layer and the evaluation index is computed. The weights W_js are updated in order to minimize the index E. The task of minimization of $E(\mathbf{W})$ (A.4) with respect to $\mathbf{W}$ is performed using gradient-descent technique, where the change in W_j $(\triangle W_j)$ is computed as

$$\triangle W_j = -\eta \frac{\partial E}{\partial W_j}, \forall j, \tag{6.10}$$

where η is the learning rate. Note that, d_{max} is directly computed from the unlabeled training set. The values of d_{max} and β are stored in both the output nodes for the computation of D. The details concerning the operation of the network are given in Appendix.

6.3.3 Proposed modification of the neuro-fuzzy algorithm for handling large data

As we have seen in Section 6.2, the number of patterns generated for an $N \times N$ image is $N^2 = s$. Each of these patterns corresponds to a pixel and has all the multiscale wavelet feature. Therefore, for selecting an optimal set of features, the

number of patterns to be presented to the connectionist system in one epoch, during its training, is $\frac{s(s-1)}{2} = \frac{N^2(N^2-1)}{2}$, which is a very large number. This requires a very high computational time. In order to avoid such situation, *i.e.*, in order to make the neuro-fuzzy algorithm computationally more efficient, we, first of all, apply a clustering algorithm (e.g., k-means clustering algorithm) on the entire feature space, for grouping the data, and the cluster centers $\boldsymbol{cen}_q$'s are noted. Then two sets of samples, namely, $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_p, \cdots, \boldsymbol{x}_{N^2}\}$ and $S_c = \{\boldsymbol{cen}_1, \boldsymbol{cen}_2, \cdots, \boldsymbol{cen}_c\}$ are formed. That is, S is the entire training set, and S_c is the set of c cluster centers (for c clusters) obtained by the clustering algorithm. Now the similarity between the patterns and these cluster centers are computed, instead of computing it for every pair of patterns. These cluster centers are considered as representatives (prototypes) of all the points belonging to the respective clusters. Thus, the number of patterns to be presented to the network in one epoch becomes $\frac{s(s_c-1)}{2} = \frac{N^2(c-1)}{2}$, where $s = |S|$ and $s_c = |S_c| << s$. This is a way of making the algorithm scalable.

Note that the clustering algorithm at this stage provides a first hand knowledge about the various objects in an image. Based on this knowledge, neuro-fuzzy feature selection algorithm is applied, in order to remove the redundant and confusing features, and thereby improving the segmentation result. The algorithm for finding optimal $\vec{\boldsymbol{W}}$ is as follows.

Algorithm for learning $\boldsymbol{W}$

- Cluster the patterns using an appropriate algorithm (*e.g. k-means* clustering algorithm) and the cluster centers $\boldsymbol{cen}_q$'s are noted.
- Calculate d_{max} from the unlabeled training set and store it in both the output nodes. Store β (user specified) in both the output nodes.
- Initialize W_j with small random values in $[0, 1]$.
- Repeat until convergence, *i.e.*, until the value of E becomes less than or equal to certain predefined small quantity, or number of iterations attains certain predefined number of iterations:
  - For each pair of patterns $\vec{\boldsymbol{x}}_p$ and $\vec{\boldsymbol{cen}}_q, \forall p, q$:
    - Present the pattern pair to the input layer.

* Compute $\triangle W_j$ for each j using the updating rule in (6.10).

– Update W_j for each j with $\triangle W_j$ averaged over all the pattern pairs.

$E(\mathbf{W})$, after convergence, attains a local minimum and then the weights ($W_j = w_j^2$) of the links connecting hidden nodes and the output node computing μ^T-values, indicate the order of importance of the features. Note that this unsupervised method performs the task of feature selection without clustering the feature space explicitly and does not need to know the number of clusters present in the feature space.

6.3.4 Segmentation

The features thus selected are used for segmenting a texture image. For this purpose, the patterns corresponding to the pixels, generated in Section 6.2, undergo clustering using *k-means* clustering algorithms, where each of the clusters correspond to a constituent texture class of the image. Therefore, the number of clusters is the same as the number of texture classes. In this work we have used the *k-means* clustering algorithm twice. It has been used for the purpose of getting a preliminary segmentation using the features prior to neuro-fuzzy feature evaluation, and again for the final segmented output with features after the neuro-fuzzy feature evaluation step.

6.4 Experimental Results

In this section, we demonstrate the effectiveness of the proposed methodology on various multi-texture images, for segmentation. The superiority of this methodology, over other existing methods studied in [132] is also established.

6.4.1 Data sets

The proposed feature extractor (Section 4.3.2) uses some window, which is shifted from one pixel to the other. So, the complexity of textures and shape of the borders between different texture regions, in a multi-texture test image is of major interest. The ability to identify and discriminate between these boundaries is of course an important aspect. Therefore, we have chosen test images that are appropriate for this study.

Here we have considered two 5-texture images $Nat5b$ (Fig. 6.5(a)) and $Nat5v$ (Fig. 6.6(a)) of size 256 × 256 each, a 10-texture image $Nat10a$ (Fig. 6.7(a)) of size 256 × 512, and a 16-texture image $Nat16c$ (Fig. 6.8(a)) of size 512 × 512 [132]. All these images are obtained from common sources, like Brodazt album [27] and MIT Vision Texture database [114], and are considered as benchmark images for texture analysis. Moreover, we have considered three natural scenes ($ROCK$, $MB10$ and $MB433$). These images were captured using different equipment and under different conditions. All these textures and natural scenes are gray-scale images, and the dynamic ranges are represented by eight bits per pixel. In order to make the textures indiscriminable for the local mean gray level or local variance, *i.e.*, to abolish any variation in illumination, the images have separately been globally histogram equalized prior to being used.

The 256 × 256 5-texture images are:

- $Nat5b$, consisting of D77, D84, D55, D53, and D24 [27], and
- $Nat5v$, a complicated image consisting of Fabric.0000, Fabric.0017, Flowers.0002, Leaves.0006, and Leaves. 0013 [114].

The 256 × 512 image with simple borders but with as many as 10 texture classes is:

- $Nat10a$, consisting of D4, D9, D19, D21, D24, D28, D29, D36, D37, and D38 [27]

The 512 × 512 image with complex boundaries and as many as 16 texture classes is:

- $Nat16c$, consisting of D3, D4, D5, D6, D9, D21, D24, D29, D32, D33, D54, D55, D57, D68, D77, and D84 [27]

We have also worked on another test image $Patch5$ of size 256 × 256 comprising of seven classes (Fig. 6.9(a)). Note that this image also seems to be visually difficult to identify the different texture regions.

Several natural scenes of size 512 × 512 were also analyzed. These are:

- *Rock* (Fig. 6.10(a)), a complicated image consisting of six classes, *viz.*, dark rocks, light rocks, pebbles, dust, sky and horizon. This is an image taken from Sol-3 of the Mars Pathfinder Mission [143].

- $MB10$ (Fig. 6.11(a)), consisting of 3 classes, *e.g.*, sky, clouds and trees.
- $MB433$ (Fig. 6.12(a)), consisting of five classes, *e.g.*, leaves1, leaves2, grass, road and miscellaneous.

The images $MB10$ and $MB433$ are obtained from the MINERVA (Machine Intelligence for Natural Environment Recognition and Visual Analysis) [113] benchmark images.

Throughout the study, the parameters are chosen to be, ϵ_1=2-5%, ϵ_2=50%, local window sizes $w_l \times w_l$ are 9×9 and 17×17 for the first, second level of resolutions. The averaging window sizes $G \times G$ are also chosen in commensuration with the local windows.

6.4.2 Performance evaluation of the proposed methodology

Here we demonstrate the effectiveness of our methodology over several texture mosaics as described in the previous section. We have experimented with energy and entropy features, and have found that the former shows better sensitivity over the later for these texture images. So, all the results that we have presented in this work are by using energy as textural feature. The percentage of correctly classified pixels has been used as the segmentation quality measure. In order to validate the importance of neuro-fuzzy feature evaluation, we present here the segmented outputs of all the test images prior to this step. Also we show how the feature dimensionality can be greatly reduced after feature evaluation. Simple median filtering can be applied to the class maps as a post-processing step to improve the segmentation results. This is because median filtering is an edge preserving smoothing operation. It tries to remove small variations within a region but at the same time tries to keep the boundaries between regions intact. Several such edge preserving smoothing filtering techniques exist, but since median filtering is easy to implement we have used it for our purpose.

Fig. 6.5(b)) shows the segmented image of $Nat5b$ (Fig. 6.5(a)). The 5 texture classes can easily be identified here. For $Nat5v$ (Fig. 6.6(a)), the classification error is more in the segmented output (Fig. 6.6(b)) as compared to Fig. 6.5(a), due to wider within class variation. The percentages of correctly classified pixels has been found to be for $Nat5b$ 97.9% and 84.9% for $Nat5v$. The segmented outputs for $Nat5b$ and $Nat5v$ prior to the neuro-fuzzy feature evaluation are also depicted in Figs. 6.5(c) and 6.6(c). Note

Table 6.1: Performance of different test images

Test figure	*Steps*				
	With feature evaluation		*Without feature evaluation*		*With Post-processing*
	Classification	Number of features	Classification	Number of features	Classification
Nat5b	97.9%	5	94.6%	21	98.7%
Nat5v	84.9%	5	84.4%	32	86.2%
Nat10a	79.5%	7	71.7%	19	84.0%
Nat16c	79.0%	11	67.3%	29	80.4%
Patch5	89.9%	5	88.1%	12	92.2%

that the neuro-fuzzy feature evaluation method has enhanced the quality of output images in terms of percentage classification (Table 6.1). Moreover, number of features have been greatly reduced from 21 to 5 for *Nat5b* and 32 to 5 for *Nat5v*, by using neuro-fuzzy feature evaluation. The segmented outputs after a 5×5 median filtering are shown in Figs. 6.5(d) and 6.6(d) and found a further improvement in quality of segmentation (Table 6.1).

For texture mosaic *Nat10a* (Fig. 6.7(a)), the segmented output is presented in Fig. 6.7(b). Note that although the image contains 10 different Brodazt textures, some of them are not distinctly identifiable visually as depicted in Fig. 6.7(a). Interestingly, the proposed methodology has been able to identify more or less all the classes. Here the percentage of correctly classified pixels has been found to be 79.55%. Like *Nat5b* and *Nat5v*, the quality of segmentation has been improved by neuro-fuzzy feature evaluation (Fig. 6.7(c)). A great reduction in feature dimensionality is achieved via this feature evaluation step where the number of features has been reduced from 19 to 7 (Table 6.1). After the post processing step, a further improvement is noted as shown in Fig. 6.7(d) and Table 6.1.

The texture mosaic *Nat16c* (Fig. 6.8(a)), which comprises of as many as 16 Brodazt textures, also shows very complex boundaries between various constituent textures. The segmented image (Fig. 6.8(b)), shows 15 different classes, where two of the classes

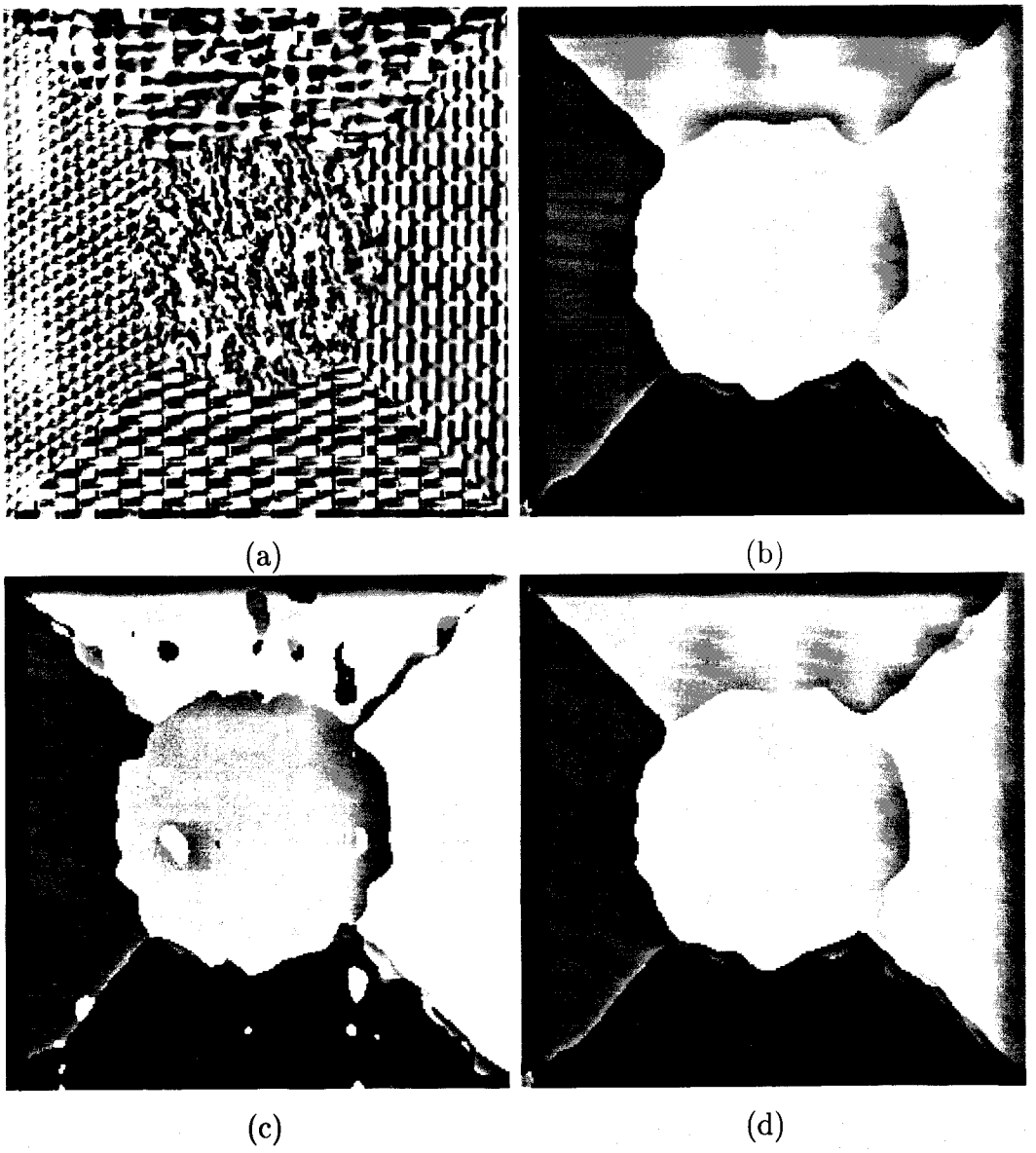


Figure 6.5: (a) $Nat5b$ segmented output (b) after and (c) before neuro-fuzzy feature evaluation (d) after post processing class = 5

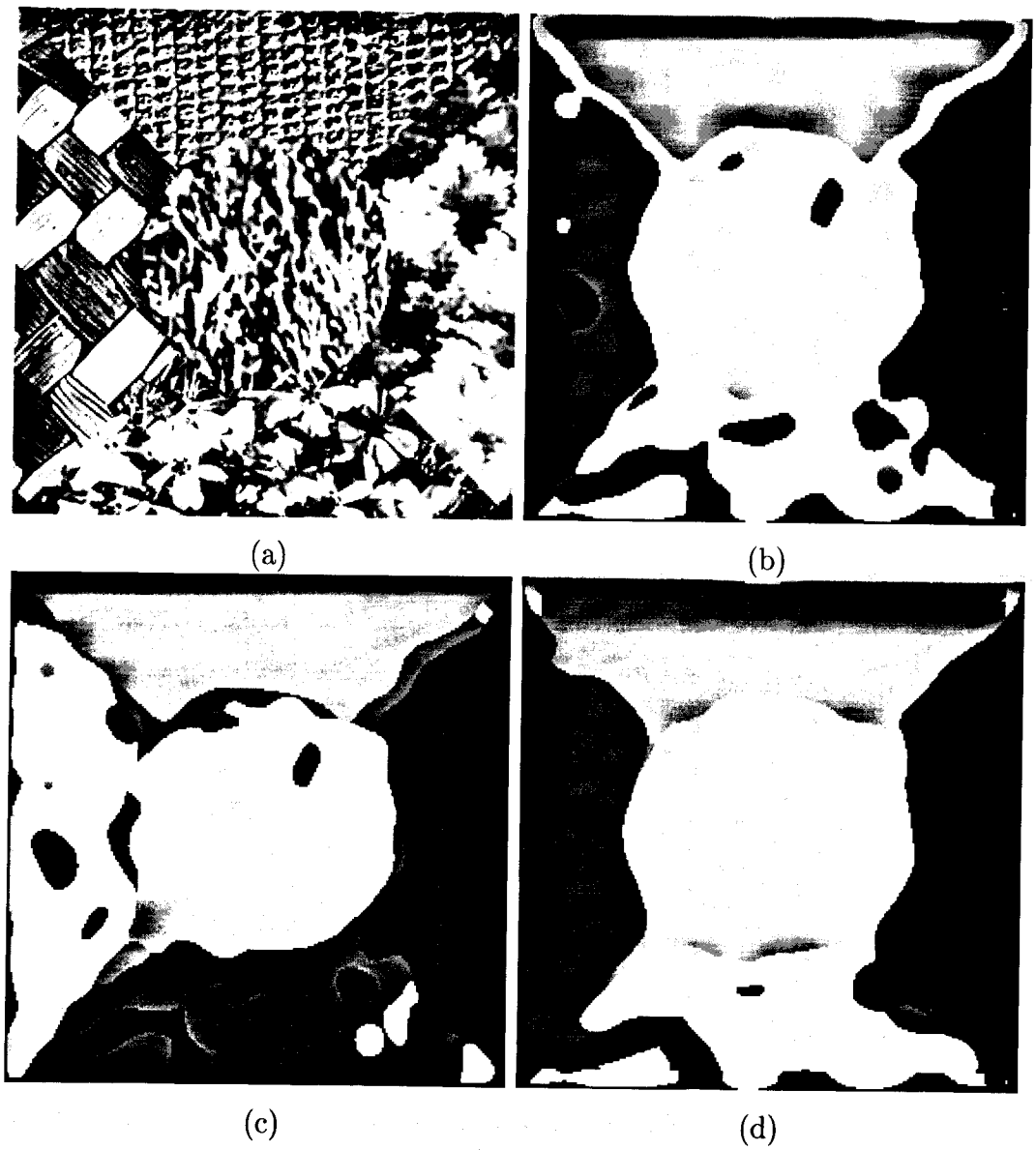


Figure 6.6: (a) $Nat5v$, segmented output (b) after and (c) before neuro-fuzzy feature evaluation (d) after post processing class = 5

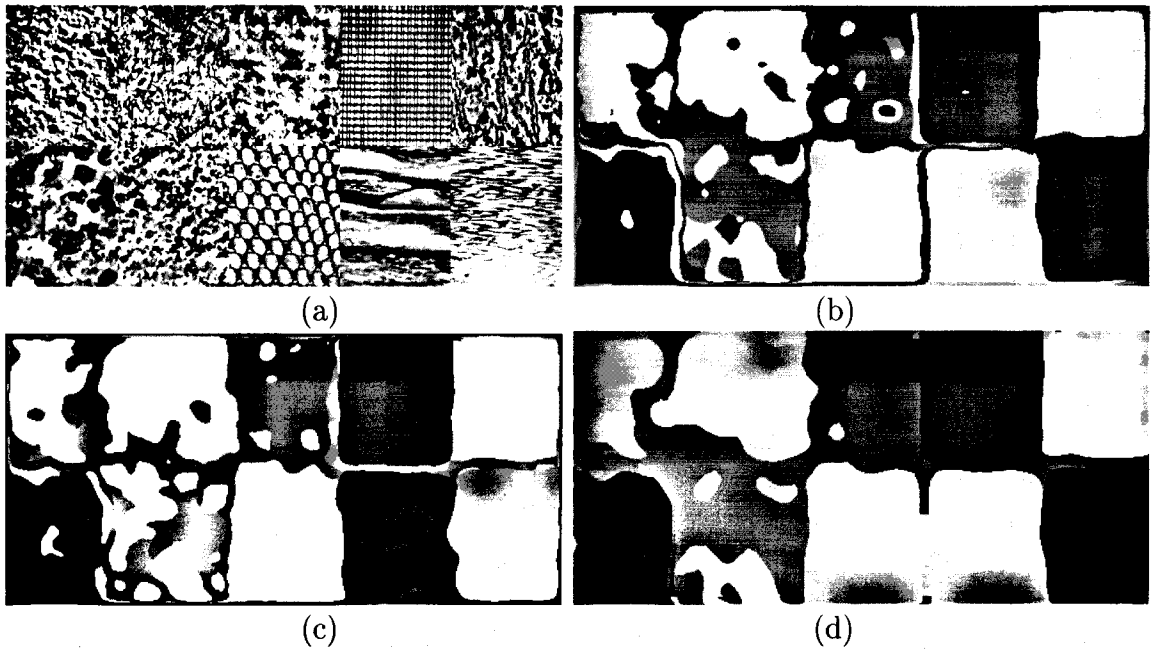


Figure 6.7: (a) *Nat*10*a*, segmented output (b) after and (c) before neuro-fuzzy feature evaluation (d) after post processing class = 10

have been merged. Note that these two classes can not be visually identified from the original image (Fig. 6.8(a)) also. The percentage of correctly classified pixels is 79.0% for this image. The segmented output for *Nat*16*c* without feature evaluation is given in Figure 6.8(c). In this case also, the classification performance increases and there is a great reduction in feature dimensionality (from 29 to 11) after the neuro-fuzzy feature evaluation step. The improved segmented output of *Nat*16*c* after median filtering is given in Fig. 6.8(d) and Table 6.1.

In the case of *Patch*5 (Fig. 6.9(a)), there are 7 different texture classes. Here a particular class is not only confined to a specific region, but also distributed in various regions of the image. Moreover the boundaries between various classes (*e.g.*, classes A and B), like other images, are not easily discernible. Even then the classes in the segmented image (Fig. 6.9(b)) can be identified satisfactorily. The percentage of correct classification has been found to be 89.9%. Here also the neuro-fuzzy feature evaluation method improves the quality of segmentation (Fig. 6.9(c) and Table 6.1) and reduces the number of features from 12 to 5. The post processing step leads to a further improvement of the segmented output (Fig. 6.9(d) and Table 6.1) as in the previous examples.

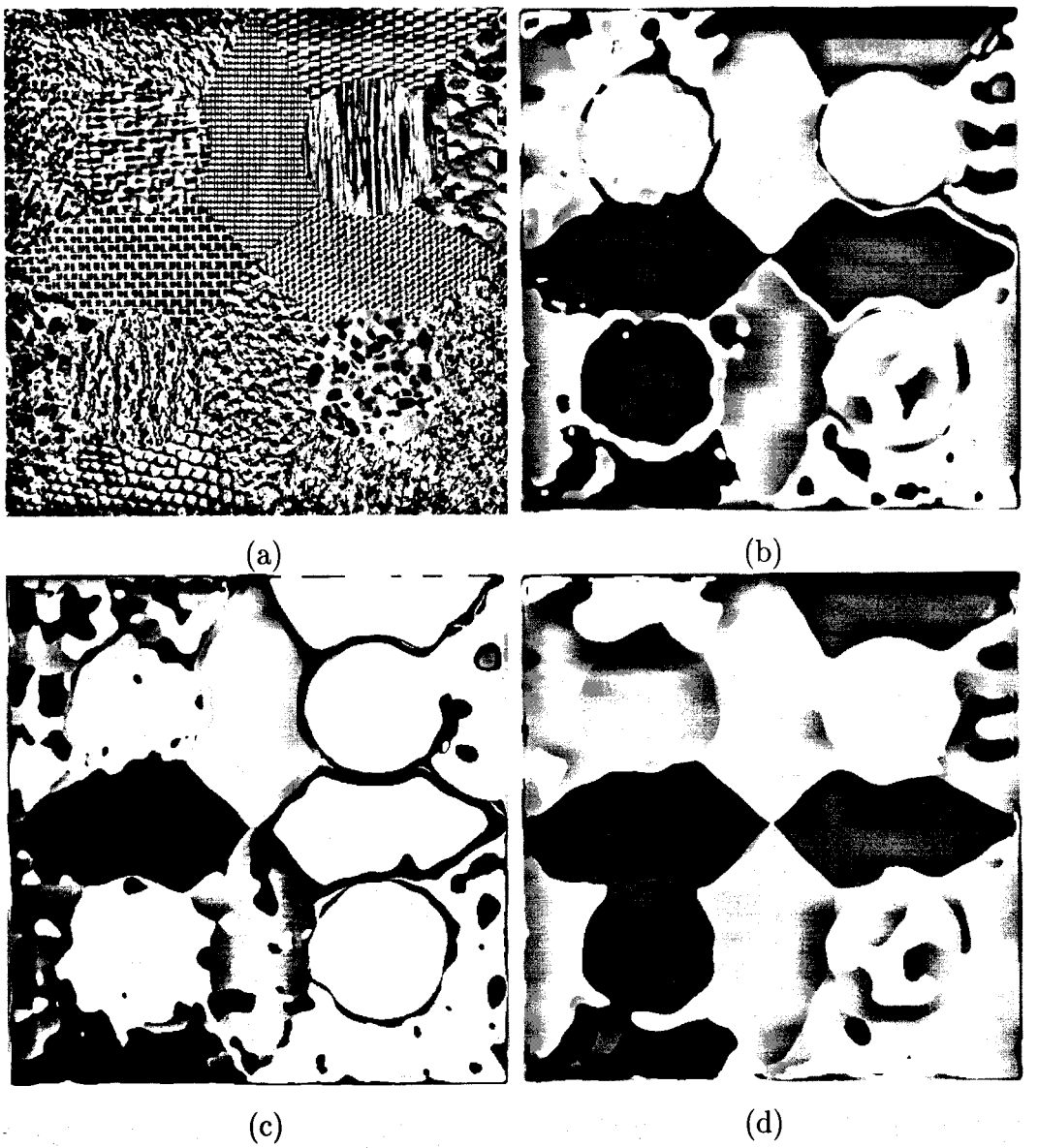


Figure 6.8: (a) *Nat*16*c*, segmented output (b) after and (c) before neuro-fuzzy feature evaluation (d) after post processing class = 16

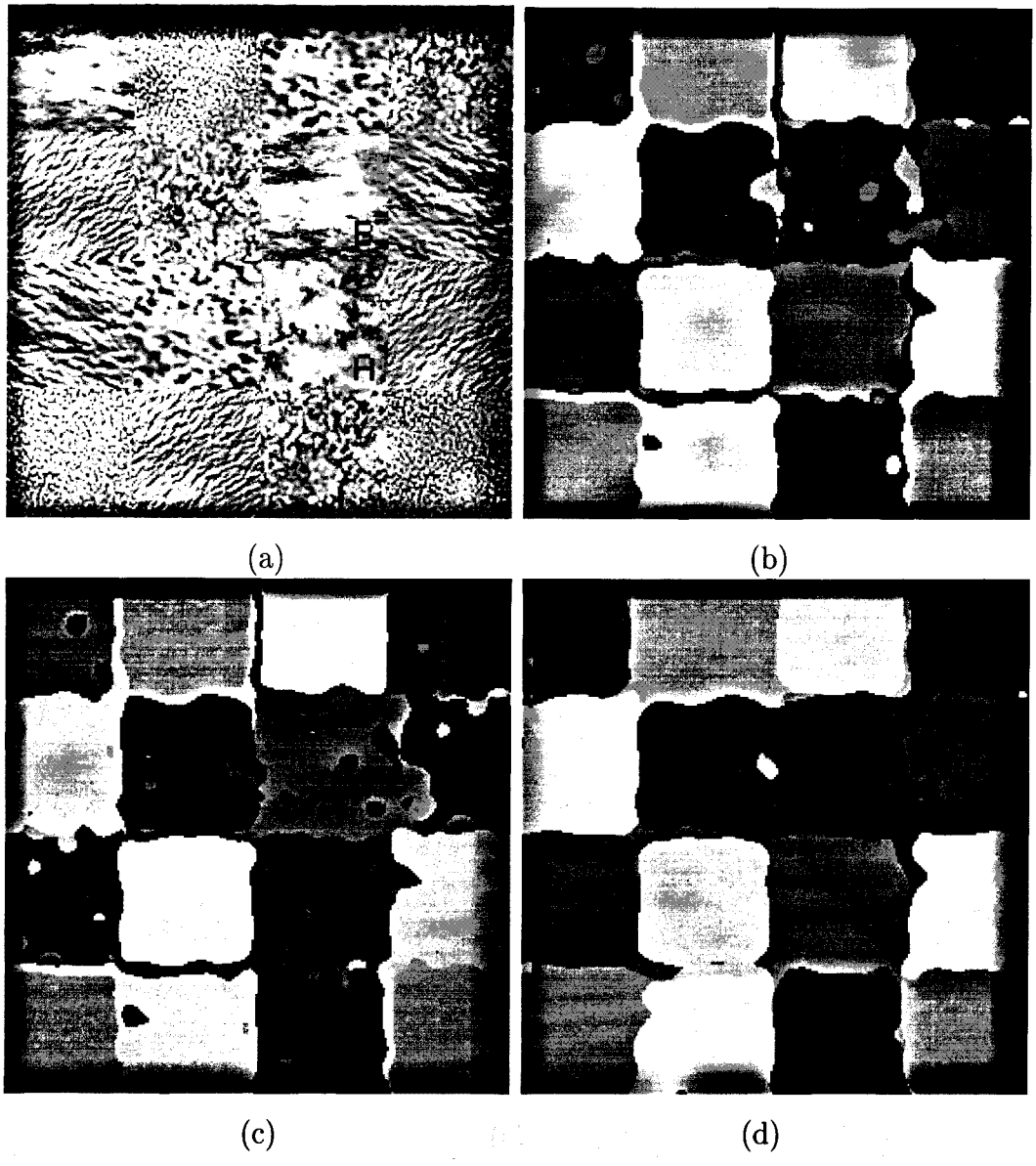


Figure 6.9: (a) *patch*5, segmented output (b) after and (c) before neuro-fuzzy feature evaluation (d) after post processing class = 7

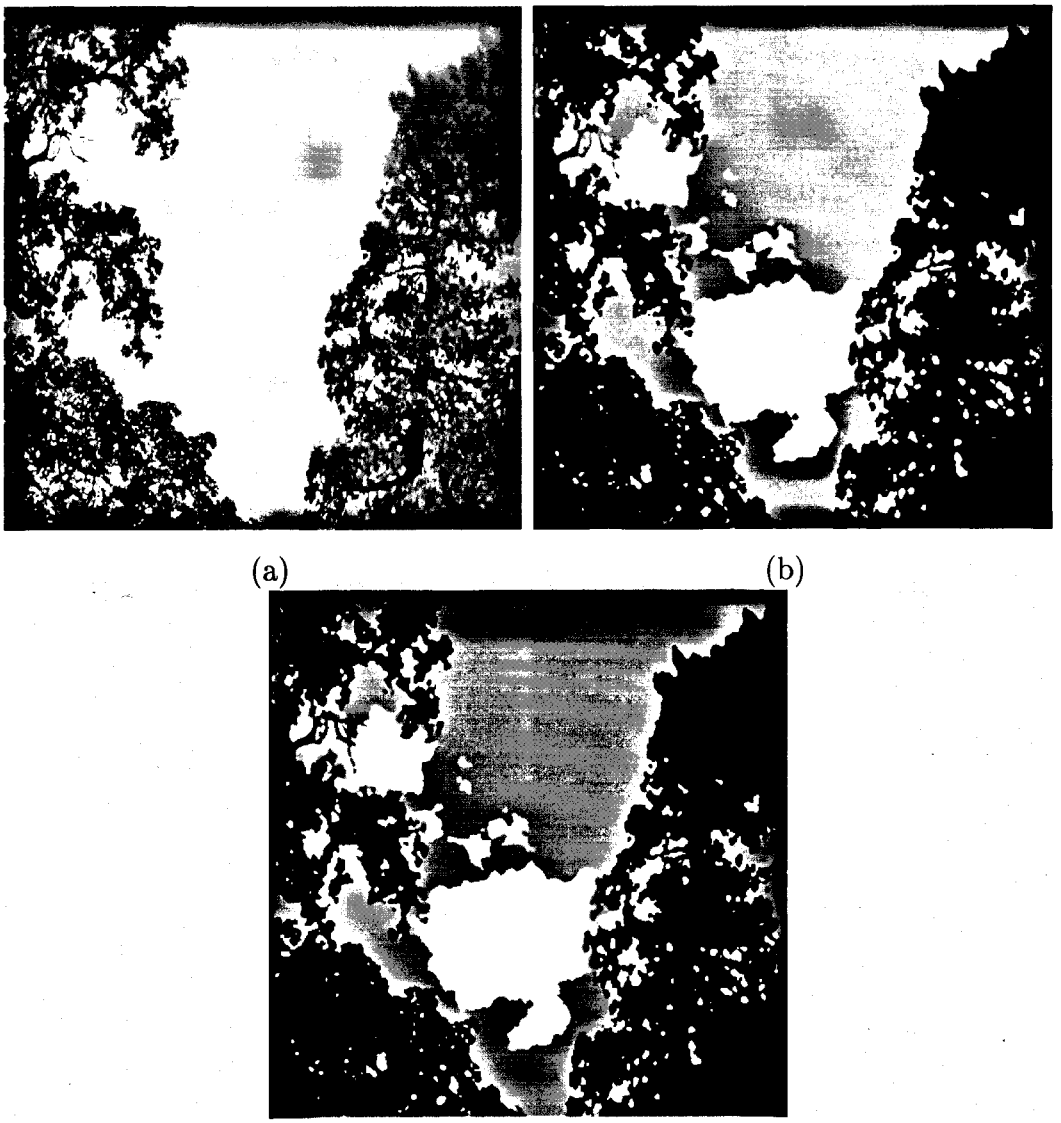
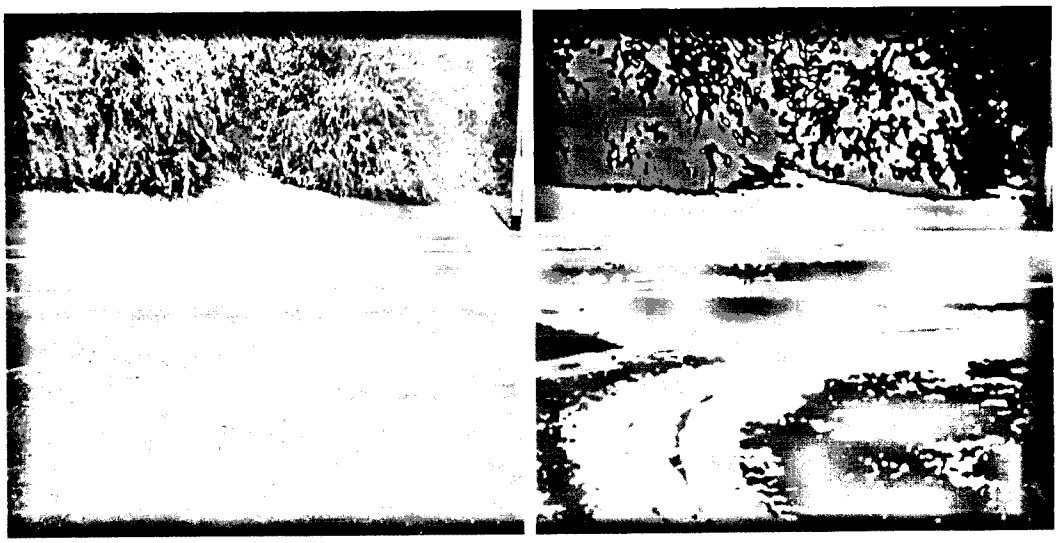


Figure 6.10: (a) Natural scene - 1 (b) before (c) after neuro-fuzzy feature evaluation class = 3



(a) (b)

(c)

Figure 6.11: (a) Natural scene - 2 (b) before (c) after neuro-fuzzy feature evaluation class = 5

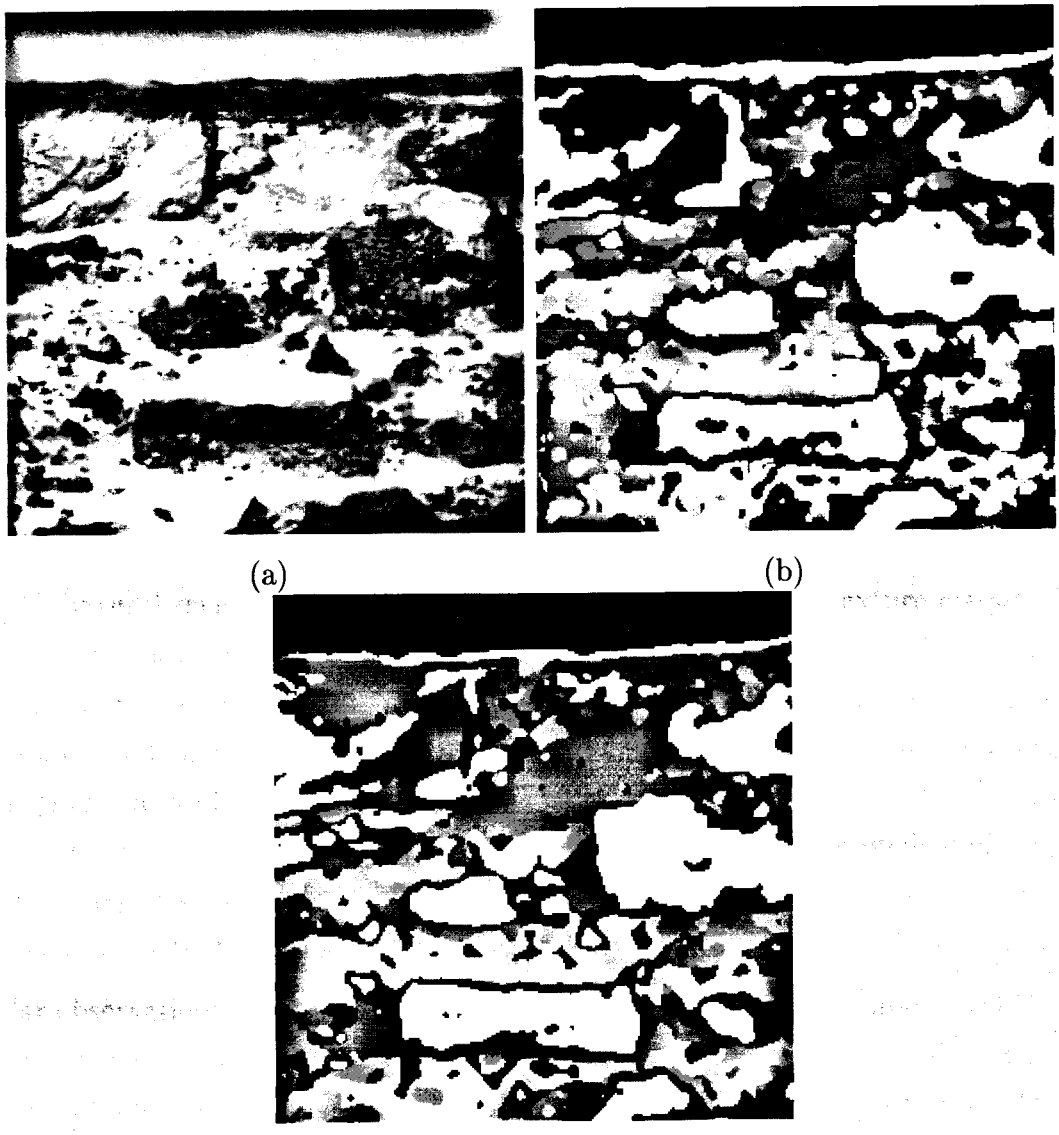


Figure 6.12: (a) Natural scene - 3 (b) before (c) after neuro-fuzzy feature evaluation class = 6

The real life natural scene images contain natural objects like roads, brick walls, etc. having uniform texture, and snow, leaves, trees, etc. more of a fractal nature. In our natural object classification task, the following objects have been considered: trees, grass, leaves, clouds, sky, roads, pebbles and rocks. A single image can contain more than one of these objects. Some of these classes are fairly homogeneous in their texture, e.g. sky, and road, whereas the others are more varied, e.g. trees, grass, leaves and clouds.

Since we do not know the exact boundaries between different objects/classes in the natural scene images, the classification score has not been used in these cases as segmentation quality measure. Instead the quantitative index β [124] defined in Chapter 4 (Section 4.4) is considered for this purpose. In equation (4.10), the numerator is constant for a given image and number of classes present in the image. Therefore, the value of β is dependent only on the denominator. The denominator, on the other hand, decreases with homogeneity of a region. So higher the value of β, better is the segmentation.

For the natural scene *Rock* (Fig. 6.10(a)), there are 6 different texture classes. Here a particular class is not only confined to a specific region, but also distributed in various regions of the image. Moreover the boundaries between various classes, like other images, are not easily discernible. Even then the classes in the segmented image (Fig. 6.10(b)) has been identified satisfactorily. The value of β is found to be 2.991870. Here also the neuro-fuzzy feature evaluation method has improved the quality of segmentation as depicted by Fig. 6.10(c) and the value of $\beta = 3.559690$, and has reduced number of features from 5 to 3 (Table 6.2).

Similar observations are noted for the images $MB10$ (Fig. 6.11(a)) and $MB433$ (Fig. 6.12(a)), where β values for the segmented images (Fig. 6.11(b) and Fig. 6.12(b)) have been found to be 5.404067 and 4.272661 respectively. The the neuro-fuzzy feature evaluation method has reduced the number of features from 5 to 3 for $MB10$ and 5 to 1 for $MB433$ (Table 6.2).

Table 6.2: Performance of different scenes

Test figure	*Steps*			
	With feature evaluation		*Without feature evaluation*	
	β	Number of features	β	Number of features
Rock	2.931870	3	3.559690	5
$MB10$	5.404067	3	5.409503	5
$MB433$	4.272661	1	6.2240	5

6.4.3 Comparison

As part of the investigation, an extensive comparison has been made to show the superiority of the proposed methodology over various existing related algorithms. Note that several approaches to multichannel filtering for texture segmentation have been studied and a comparative performance evaluation is presented by Randen and Husφy in [132].

They have compared the performance of several heuristically designed filter banks, such as Laws filter masks [101], Dyadic Gabor filter banks [60], wavelet transform, wavelet packets [32, 98, 144], wavelet frames [164] and quadrature mirror filters (QMF) [131], used as the feature extractors. Considering the average performance over the texture mosaic $Nat5b$, a non-dyadic bank of Gabor filters [131] have been found to produce better result, while for $Nat5v$ and $Nat16c$ the FIR filter $f16b$ [131, 88] gives a better performance over other filters. For $Nat10a$, the best result is obtained by a dyadic Gabor filter bank [60].

Most of the heuristically designed filters use large number of features which result in high computational complexity. Therefore, the optimization of the filtering operation with respect to some explicit criterion related to texture classification is desirable. Randen and Husφy [132] made extensive study on the performance of several optimized filter banks like, Gabor filter and finite impulse response (FIR) filters [88]. These were optimized following Mahalanobis and Singh (J_{MS}) [132], Unser (J_U) [132] and Fisher (J_F) [132] criterion. The *optimized representation Gabor* filter bank [26, 25] produced the best result as far as texture mosaic $Nat5b$ is concerned, while the *10-filter optimized*

Gabor filter bank [170] gives the best result for $Nat5v$. The best result for the texture mosaic $Nat16c$ is produced by the *prediction error filter* [131] and that for $Nat10a$ is given by FIR filters optimized with respect to J_U [131].

Randen and Husφy [131] studied another class of filters, the QMF, incorporating both finite impulse response (FIR) [88] and infinite impulse response (IIR) filters [85]. In this study they have analyzed several IIR and FIR filters and arbitrary Gabor filter banks, applied to various subband decompositions which are dyadic as well as non-dyadic [132]. Apart from the above category of filters, they have also examined several wavelet filters, applied to octave band decomposition, as well as wavelet frame and wavelet packet decompositions, for the purpose of segmentation. All the above filters have been studied for both critical rate (subsampled) as well as full rate (non-subsampled). For the full rate (non-subsampled) case, FIR filter $f8a$ [131, 88] gives best result for $Nat5b$. For $Nat10a$, an arbitrary Gabor filter bank [131] gives the best performance. While, an FIR filter $f16b$ is found to give the best results for $Nat5v$ and $Nat16c$. All the above filters use a non-dyadic decomposition scheme. For the critical rate (subsampled) case, the FIR filter $f8a$ produces the best result for $Nat5b$. Filter $f32d$ [131, 88] gives best performance for $Nat5v$ with a non-dyadic decomposition and that for $Nat10a$ with a dyadic decomposition. While, for $Nat16c$ the filter $f16b$ gives the best result with a non-dyadic decomposition.

In this chapter, the comparison is made with respect to the number of features used for texture segmentation and the percentage of correctly classified pixels. Tables 6.3 and 6.4 show that only 5 M-band wavelet features have been able to classify 97.9% and 84.9% pixels of $Nat5b$ and $Nat5v$ images respectively. On the other hand, the other algorithms could not achieve this performance even with a large number of features. Similar is the case for $Nat16c$ (Table 6.5) and $Nat10a$ (Table 6.6) images, where the best classification performances are achieved with the least number of features obtained by the proposed methodology. The differences between classification score obtained by our methodology and those by others are also significant for these images.

Table 6.3: Performance evaluation of *Nat5b*

Methods/filters		Classification	Number of features
Proposed method *M*-band wavelet	Without feature evaluation	94.6%	21
	With feature evaluation	97.9%	5
Heuristic Gabor filter bank (d) [131]		91.8%	40
Optimized Optimized representation Gabor filter [26] [25]		92.8%	10
Full rate QMF FIR filter *f8a* (d) [131]		92.8%	40
Critically sampled QMF FIR *f8a* (d) [131]		90.6%	40

Table 6.4: Performance evaluation of *Nat5v*

Methods/filters		Classification	Number of features
Proposed method *M*-band wavelet	without feature evaluation	84.4%	32
	with feature evaluation	84.9%	5
Heuristic Gabor filter bank (d) [131]		81.1%	40
Optimized Optimized representation Gabor filter [26] [25]		70.6%	10
Full rate QMF FIR filter *f8a* (d) [131]		81.1%	40
Critically sampled QMF FIR *f8a* (d) [131]		79.5%	40

Table 6.5: Performance evaluation of *Nat16c*

Methods/filters		Classification	Number of features
Proposed method *M*-band wavelet	without feature evaluation	67.3%	29
	with feature evaluation	79.0%	11
Heuristic Gabor filter bank (d) [131]		63.6%	40
Optimized Optimized representation Gabor filter [26] [25]		65.3%	16
Full rate QMF FIR filter *f8a* (d) [131]		63.6%	40
Critically sampled QMF FIR *f8a* (d) [131]		60.9%	40

Table 6.6: Performance evaluation of *Nat10a*

Methods/filters		Classification	Number of features
Proposed method M-band wavelet	without feature evaluation	71.7%	19
	with feature evaluation	79.5%	7
Heuristic Gabor filter bank (d) [131]		67.7%	20
Optimized Optimized representation Gabor filter [26] [25]		64.1%	15
Full rate QMF FIR filter *f8a* (d) [131]		60.3%	40
Critically sampled QMF FIR *f8a* (d) [131]		57.7%	13

6.5 Conclusion

In this chapter, we have described a feature extraction method based on M-band wavelet packet frames followed by neuro-fuzzy evaluation of the extracted features for texture segmentation. The use of M-band wavelet decomposition of the texture image provides an efficient representation of the image in terms of frequencies in different directions and orientations at different resolutions. This representation thus facilitates an improved segmentation of the different texture regions. The feature extraction method splits both the lower as well as the higher frequency bands, and results in a tree structure. The neuro-fuzzy feature evaluation method helps in finding out important features efficiently from a texture image where the various texture classes are overlapping in nature. Both the feature extraction and neuro-fuzzy feature evaluation schemes are unsupervised and do not require any knowledge about the number of classes. Moreover, the methodology also does not require *a priori* knowledge about the spatial relationship of different texture classes in the test images.

The features obtained by the feature extraction method have been able to segment different benchmark test images satisfactorily. The neuro-fuzzy feature evaluation method has reduced the number of extracted features greatly and thereby enhances the quality of segmentation with only a few number of features. The superiority of our methodology in segmenting the texture images over that of others obtained by various existing methods has been shown both qualitatively and quantitatively. This implicates that in our method, dimensionality of the feature space can be greatly reduced, while still maintaining a high segmentation quality. It has been however found for certain cases that, better segmentation results are obtained even without the neuro-fuzzy evaluation step. The next chapter discusses about the segmentation of two real life data.

Chapter 7

Segmentation of Document and Remotely Sensed Images

7.1 Introduction

This chapter describes two examples of real life applications of texture segmentation. In the first part of the chapter, an efficient and computationally fast method for segmenting text and graphics part of document images based on textural cues is presented [7, 8, 11]. It is logical to assume that the graphics part has different textural properties than the non-graphics (text) part. We have used M-band wavelet feature extraction scheme developed in Chapter 5, for this purpose. The later part of the chapter describes how the features extracted in Chapter 6, can be used for segmenting real life data such as remotely sensed images. Different quasi-homogeneous regions in the image can be treated to have different texture properties. Based on this assumption the multi-class texture segmentation scheme is applied for this purpose [9, 13].

7.2 Document Segmentation

With the advent of modern publishing technologies, the layout of today's documents has been a very complex affair. Most of them contain not only text and background regions, but also graphics, tables and pictures. Therefore scanned documents must often be segmented before other document processing techniques, such as compression or rendering, can be applied. Efforts have been there to store the documents in digitized form but that too requires an enormous storing space even after compressing, using modern techniques. Documents can be more effectively represented by separating out the text and the graphics/image part and storing the text as ASCII (character) set and the graphics/image part as bit-maps. So research in the area of document image analysis has become an important issue to many researchers.

7.2.1 Previous research

Several useful techniques for text-graphics segmentation are given in [149], the most popular amongst these being the *top-down* and *bottom-up* approaches. The most common *top-down* techniques are *run length smoothing* [34, 168] and the *projection profiles* [169, 95]. *Top-down* approaches first split the document into blocks which are then identified and subdivided appropriately, in terms of columns first and then split them

into paragraphs, text lines and may be words also. Early works of the block-based approaches have been mainly designed for binary document images. For example, Wong *et al.* [168] have proposed a technique called the run length smoothing algorithm (RLSA) to partition a binary document image into blocks. Each block is then classified as text or picture according to some statistical features. A similar algorithm has also been investigated by Wang *et al.* [169] for newspaper layout analysis. Chauvet *et al.* [34] have presented a recursive block partition algorithm based on RLSA. A more detailed survey of these approaches can be found in [79]. Recent block-based segmentation algorithms are developed mostly for gray-scale or color document images. Among these algorithms, some use features extracted from the discrete cosine transform (DCT) coefficients to separate text blocks from picture blocks. For example Konstantinides and Tretter [93] use a DCT block activity measure. Some assume these blocks to be only rectangular [153]. The *top-down*/block-based methods are not suitable for skewed texts as these methods are restricted to rectangular blocks. Whereas the *bottom-up* methods are typically variants of the *connected components* [102] which iteratively group together components of the same type starting from the pixel level and form higher level descriptions of the printed regions of the document (words, text lines, paragraphs etc.) [154]. The drawbacks with the connected components method is that it is sensitive to character size, scanning resolution, inter-line and inter-character spacings.

Several other approaches use the contours of the white space to delineate the text and non-text regions [18]. These methods can only be applied to low noise document images which are highly structured, that is all objects are separated by white background and objects do not touch each other.

Each of the above methods relies to an extent on *a priori* knowledge about the rectangularity of major blocks, consistency in horizontal and vertical spacings, independence of text, graphic and image blocks and/or assumptions about textual and graphical attributes like font size, text line orientation etc. So these methods can not work in a generic environment. It is desirable to have segmentation techniques which do not require any *a priori* knowledge about the content and attributes of the document image or rather any such knowledge might not be available, in some applications. Additionally these methods operate on thresholded images. So, for a degraded image due to poor image capturing conditions, an appropriate threshold choice is a difficult task.

Jain and Bhattacharjee's [86] method has been able to overcome these restrictions and does not require an *a priori* knowledge of the document to be processed. The document segmentation has been achieved by a texture segmentation scheme using Gabor filter as the feature extractor. One major drawback of this approach is that, the use of Gabor filter makes it computationally very expensive. Randen and Husϕy [130] have proposed a method using critically sampled infinite impulse response (IIR) QMF banks for extracting features. Both the aforementioned methods do not take into consideration the possibility of overlapped/mixed classes. Etemad *et al.* [58] have developed an algorithm for document segmentation using multiscale wavelet packet feature vectors and fuzzy local decision information.

More recent research on document segmentation include works by Choi and Baraniuk [42] which is based on wavelet-domain hidden Markov tree (HMT). Tang *et al.* [156] have introduced a mixture-state document segmentation method based on wavelet and the hidden Markov tree (HMT) models. Li and Gray [104] have developed an algorithm for segmenting document images into four classes: background, photograph, text, and graph. Features used for classification are based on the distribution patterns of wavelet coefficients in high frequency bands. Harit *et al.* [81] present a new model-based document image segmentation scheme that uses XML-DTDs (eXtensible Markup Language Document Type Definitions).

Some of the common difficulties that occur in document segmentation are,

- Differences in font size, column layout, orientation and other textual attributes.
- Document skewed with text regions having different orientations.
- Document degraded due to improper scanning.
- Combinations of varying text and background gray level.
- Text regions touching or overlapping with non-text regions.
- Irregular layout structures with non-convex or overlapping object boundaries.
- Multi-column document with misaligned text lines and different languages.

We develop a texture based document image segmentation scheme, which takes care of all the above observations. The present work is based on the assumption that the

text part in the document image comprise of one texture class and the non-text part as the other. Basically this is a two texture segmentation problem. It is already well known that textures can be characterized by their energies. A composite texture can be discriminated if it is possible to obtain information about the texture signal energies. The basic idea is to decompose the composite image into different frequency bands at different scales. The objective is to transform the edges between textures into detectable discontinuities. This creates the feature maps which give a measure of local energy around each pixel over small windows.

One of the salient features of document segmentation, as compared to other texture segmentation problems, is that there are large intra-class as well as inter-class variations in the textural features. So the multiscale nature of document constituents which are implicitly there (*e.g.*, characters, lines) justifies the need of multiscale representation scheme. In the present work we apply the two texture segmentation scheme developed in Chapter 5.

7.2.2 Unsupervised segmentation

In this section, the results on implementation of our algorithm on several document images are provided. The images considered are both structured and unstructured.

Test images

The test documents are scanned from parts of pages of the "Times of India" (TOI) and "Hindustan Times" both of which are popular news dailies in India.

- Structured document image with non-overlapping text and non-text regions scanned from "Times of India", which are as follows:
  - Fig. 7.1(a) shows a document image of size 512×512.
  - Fig. 7.2(a) shows the same image rotated by an angle of 22.5^0.
  - Fig. 7.3(a) shows the same image rotated by an angle of 90^0.
  - Fig. 7.4(a) shows the same image skewed by an angle of 25^0.
  - Fig. 7.5(a) shows another image of size 512×512.
  - Fig. 7.6(a) shows the same image scanned at half the resolution.

- Highly unstructured images with overlapped/mixed classes of size 512×512 scanned from "Hindustan Times", these are:
  - Fig. 7.7(a) shows a test image with document skewed and text regions with different orientations.
  - Fig. 7.8(a) shows a test image with non-convex and overlapping object boundaries.
  - Fig. 7.9(a) shows a document image with irregular non-text region and multi-column document with misaligned text lines and different languages.
  - Fig. 7.10(a) the text portions have different orientations as well as gray values and have different font sizes.
  - Fig. 7.11(a), the text regions overlap with non-text regions, combinations of varying text and background gray level and text regions have widely varying font size.
  - Fig. 7.13(a) shows a test image of size 512×512 that has been used by Randen and Husφy [130].
  - Fig. 7.14(a) shows a test image of size 512×512 that has been used by Jain and Bhattacharjee [86].

Experimental results

Out of the total 16 features possible in our decomposition scheme we have found that in most of the cases the number of features are limited between 3 to 5. The features are selected based on the scheme discussed in Chapter 5 (Section 5.3).

Fig. 7.1(a) shows a 512×512 pixel scanned image of a portion of a typical page of TOI, and the successful two-class segmentation of the image (Fig. 7.1(b)-(d)).

In order to prove the efficacy of our algorithm we apply this technique to segment the same image rotated by 22.5^0 (Fig. 7.2(a)). Since integer rotations are distortion free, we use fractional rotation deliberately to see how our algorithm behaves in such cases. The image is rotated about an axis through the center of it in a clockwise direction. The transformed coordinates in the rotated plane for a rotation of θ are given by,

$$x' = x\cos\theta + y\sin\theta \quad \text{and} \quad y' = -x\sin\theta + y\cos\theta$$

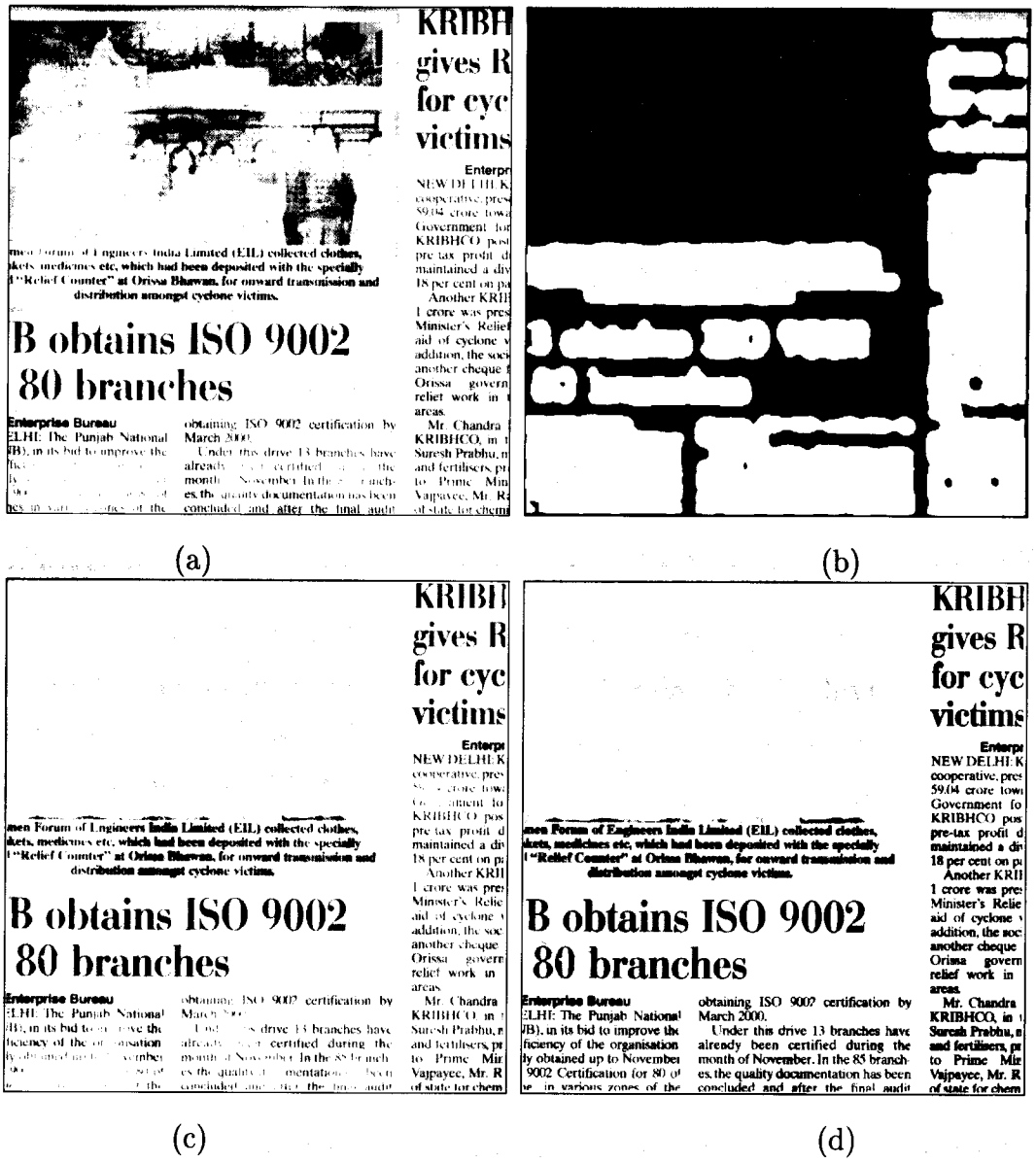
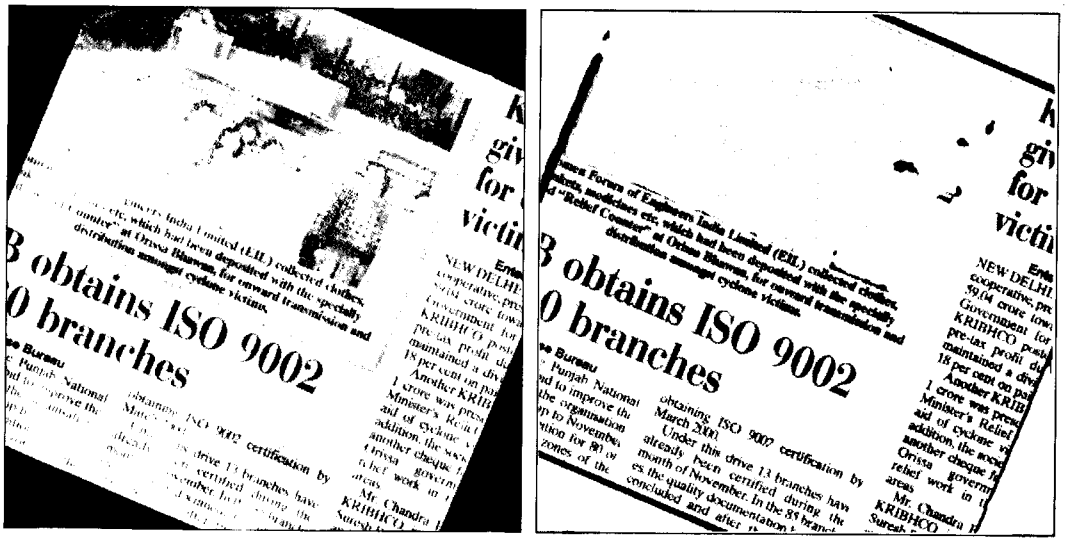


Figure 7.1: (a) A portion of a typical page of TOI (b) the two class segments from unsupervised segmentation (c) segmentation with regions classified excluding the picture (d) image segmentation after histogram thresholding



(a) (b)

Figure 7.2: (a) Same image rotated by 22.5^0 (b) segmented result

The segmentation results in Fig. 7.2(b) show that our method is almost invariant to rotation.

We also rotate the document by 90^0 (Fig. 7.3(a)) and find that in this case also we get excellent segmentation result Fig. 7.3(b), thus we can say that the scheme is almost independent of any specific layout of the document.

The same image is also given a skewed transform about an axis through the center and the transformed coordinates for a skewness of ϕ are given by,

$$x' = x + y\tan\phi \quad \text{and} \quad y' = y$$

The image is given a skewness by an angle of 25^0 (Fig. 7.4(a)) and the result is shown in Fig. 7.4(b), so we can also say that our algorithm is somewhat independent of skewness.

Fig. 7.5(a) shows a 512×512 image scanned from a typical page in the TOI and the same image scanned at half the resolution (Fig. 7.6(a)). The corresponding results are provided in Fig. 7.5(b) and Fig. 7.6(b). From the results we can infer that our scheme is invariant to scale also.

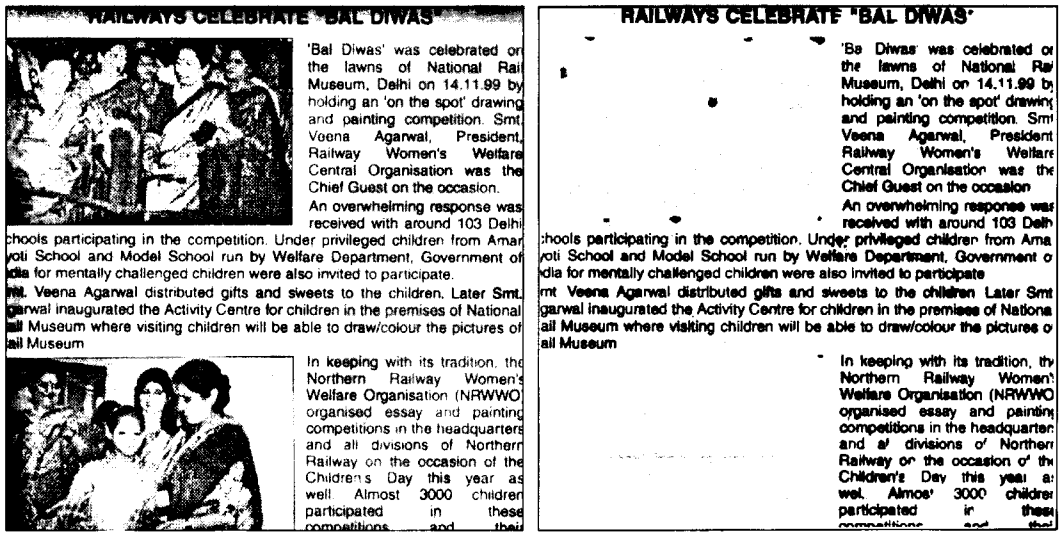
So long we have been concentrating on structured data with non-overlapping text and non-text regions. But there are several instances of documents which are highly

(a) (b)

Figure 7.3: (a) Same image rotated by 90^0 (b) two class segmentation

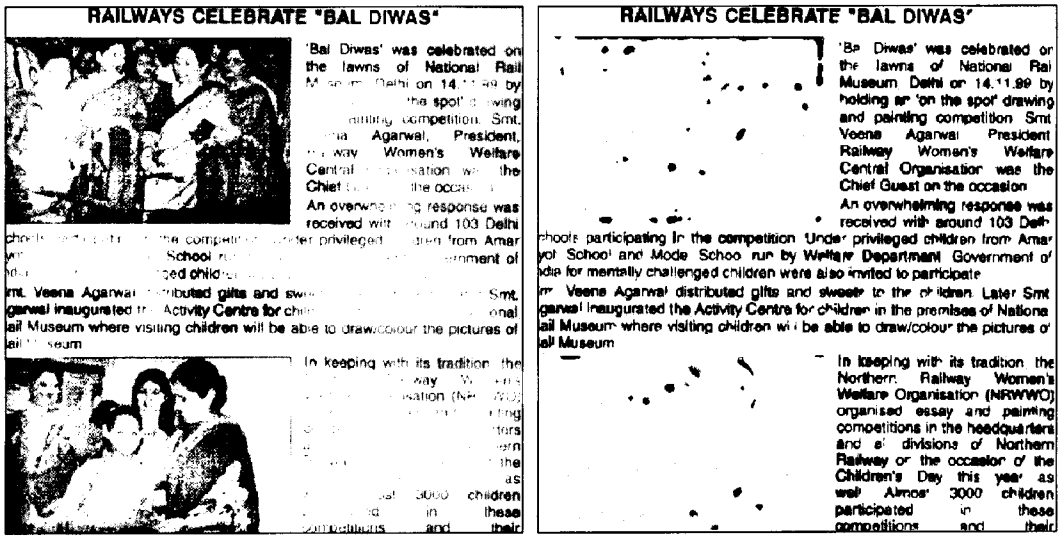
(a) (b)

Figure 7.4: (a) Image skewed by an angle 25^0 (b) corresponding image segmentation



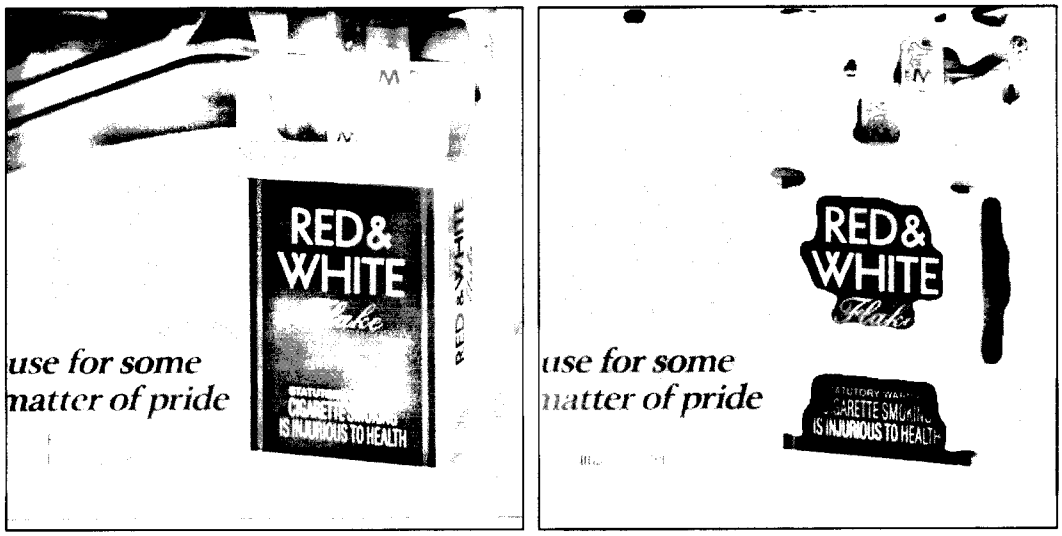
(a) (b)

Figure 7.5: (a) Image of a portion of a typical page of TOI (b) the two class segmentation



(a) (b)

Figure 7.6: (a) The same image scanned at half the resolution (b) image segmentation



(a) (b)

Figure 7.7: (a) Test image with document skewed and text regions with different orientations (b) segmented result

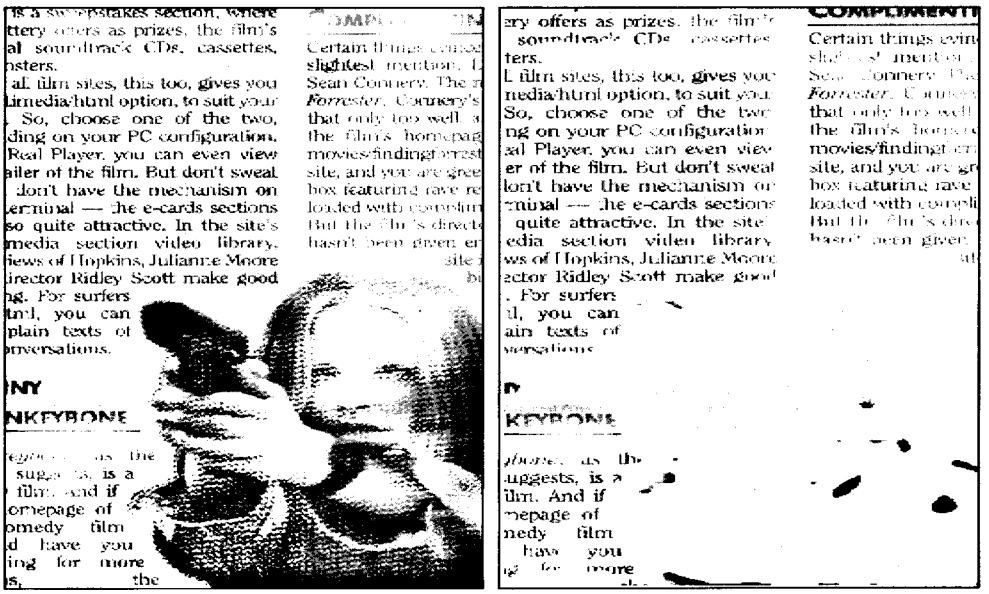
unstructured. We have experimented on several such data for an extensive study.

Fig. 7.7(a) shows a test image with document skewed and text regions with different orientations and the segmentation result is shown in Fig. 7.7(b). To prove the efficacy and robustness of our algorithm over different and diverse type of documents we have applied it on another test image with non-convex and overlapping object boundaries (Fig. 7.8(a)). The segmentation result (Fig. 7.8(b)) shows that the algorithm can efficiently identify the text and non-text regions in the documents.

While Fig. 7.9(a) shows a document image with irregular non-text region and multi-column document with misaligned text lines and different languages. The segmentation result given by Fig. 7.9(b) clearly reveals that our scheme is quite capable of identifying the text and non-text regions in such complicated documents also.

In Fig. 7.10(a) the text portions have different orientations as well as gray values and have different font sizes. The corresponding text segmentation result is shown in Fig. 7.10(b). While in Fig. 7.11(a), the text regions overlap with non-text regions, combinations of varying text and background gray level and text regions have widely varying font size. Fig. 7.11(b) shows the text portion which is segmented out.

In Fig. 7.12 we present the feature maps corresponding to test image in Fig. 7.11(a).

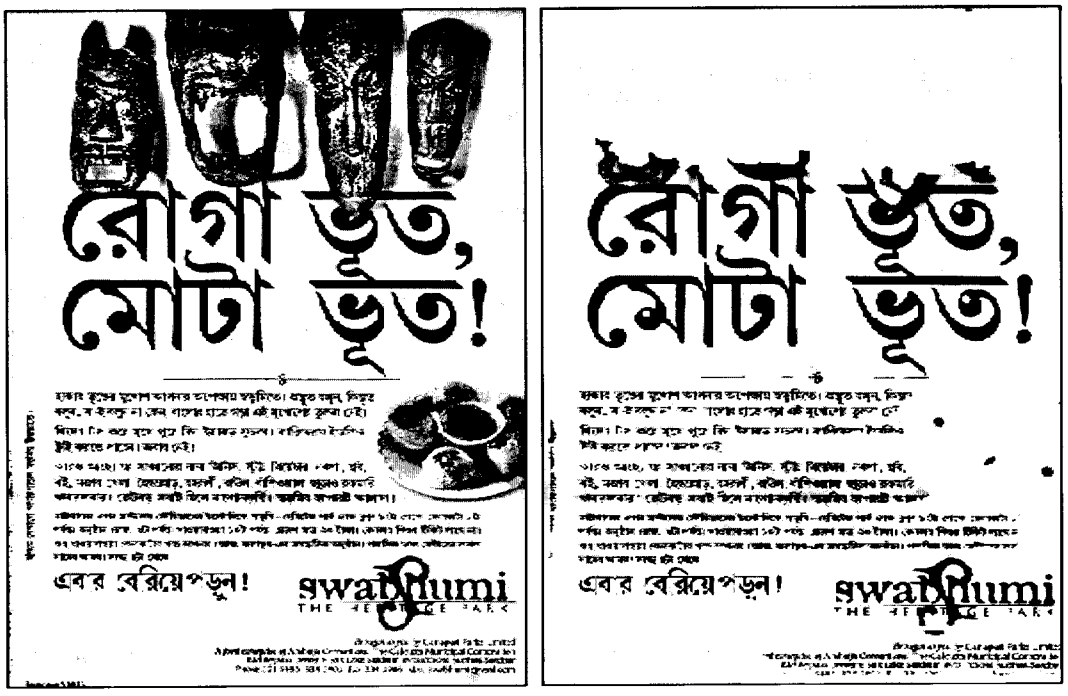


(a) (b)

Figure 7.8: (a) Test image with non-convex and overlapping object boundaries (b) segmented result

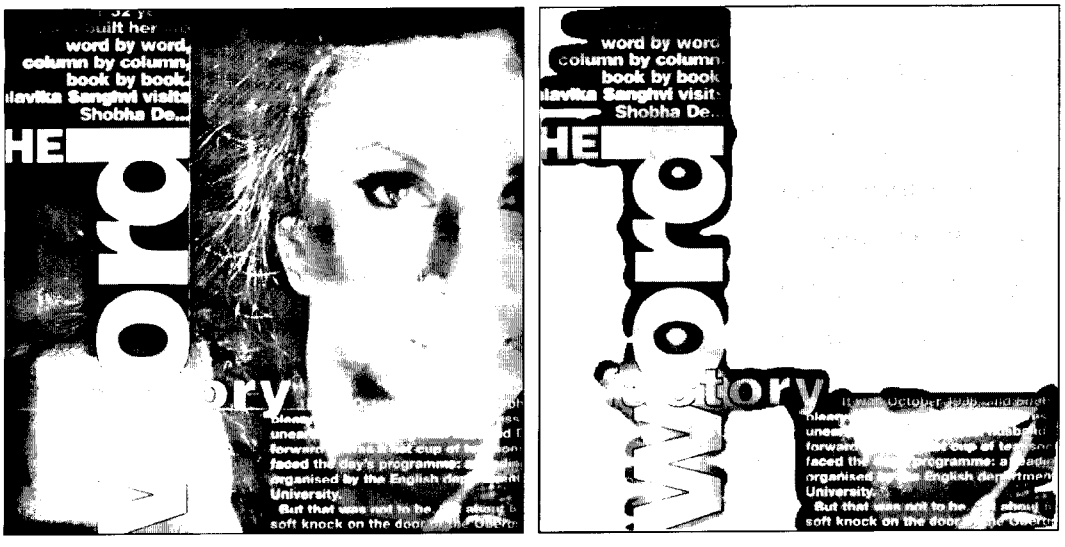
This is to give a feeling how the image is decomposed into it's constituent energies corresponding to different frequency bands and orientations. From the figure it is well understood that the signal energies are discernibly distributed amongst different frequency bands and orientations such as horizontal, vertical, diagonal, horizontal diagonal and vertical diagonal directions (here we present the histogram equalized versions of the original images for a better visibility and understanding of the readers although all the operations are done on the original images).

Throughout the experiment our effort has been to segment out the text part from the graphics part as accurately as possible. To compare our method with other methods we have used the same data that has been used by Randen and Husφy [130] (Fig. 7.13(a)). Using our algorithm we find that although some of the graphics part of Fig. 7.13(b) are misclassified as text data, we on the contrary get excellent results as far as text identification is concerned. The headings of two different font sizes could not be identified very accurately by Randen's method [130] (Fig. 7.13(c)), but have been possible by our method. The segmentation result obtained using classical wavelet packets and features suggested by Etemad *et al.* [58] (Fig. 7.13(d)) is also presented here for a comparative study.



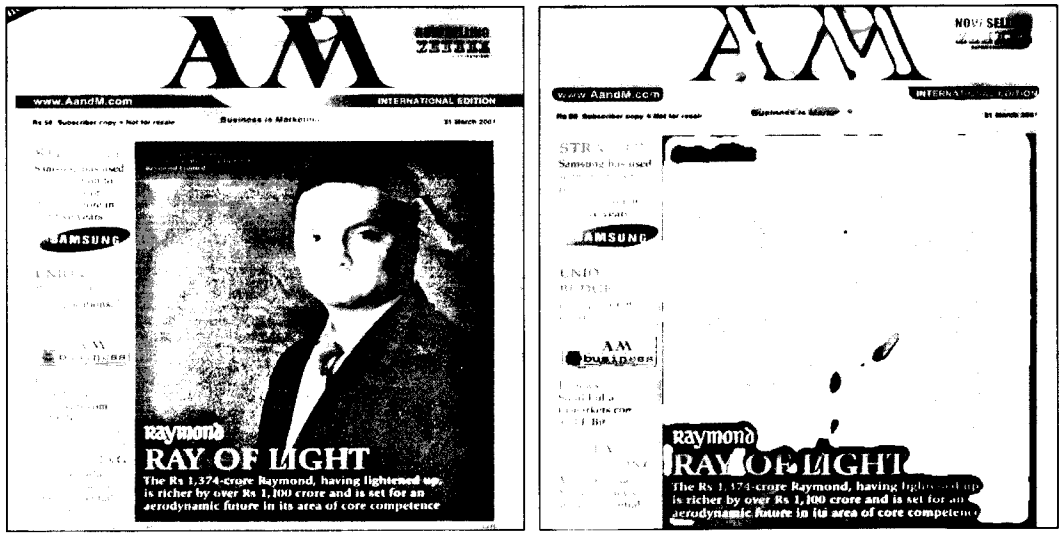
(a) (b)

Figure 7.9: (a) Test image with irregular non-text region, multi-column document with misaligned text lines and different languages (b) segmented result



(a) (b)

Figure 7.10: (a) Test image with text portions having different orientations and gray values as well as different font sizes (b) segmented result



(a) (b)

Figure 7.11: (a) Images with overlapping text and non-text regions (b) segmented result

For a comparative study we have also used the same data that has been used by Jain and Bhattacharjee [86] (Fig. 7.14(a)). Using our algorithm we find that the segmentation results (Fig. 7.14(c)) are more or less the same as in [130] (Fig. 7.14(c)). While Randen and Husϕy have verified that their method gives comparable segmentation results to that of the method used by Jain and Bhattacharjee. The result obtained using classical wavelet packets (Daubechies 8 wavelet has been used) and features given by Etemad *et al.* is also shown in Fig. 7.14(d).

It is seen that there is a significant improvement in segmentation result using $M(M > 2)$-band wavelets compared to the classical wavelet packets, where M=2. This may be explained by the additional information obtained by decomposing the image into a higher number of subbands for larger values of M. Also high frequency signals with relatively narrow bandwidth are better resolved using wavelets with higher values of M. This may also be attributed to the bandpass nature of higher bands that capture additional texture information. Moreover, in our method we have not subsampled the image, whereas in both the other methods used for the purpose of comparison, the image has been subsampled. Suitability of our method over the subsampled methods may be explained by the fact that subsampling reduces the size of the subbands at higher levels of decomposition and can possibly bias the decomposition.

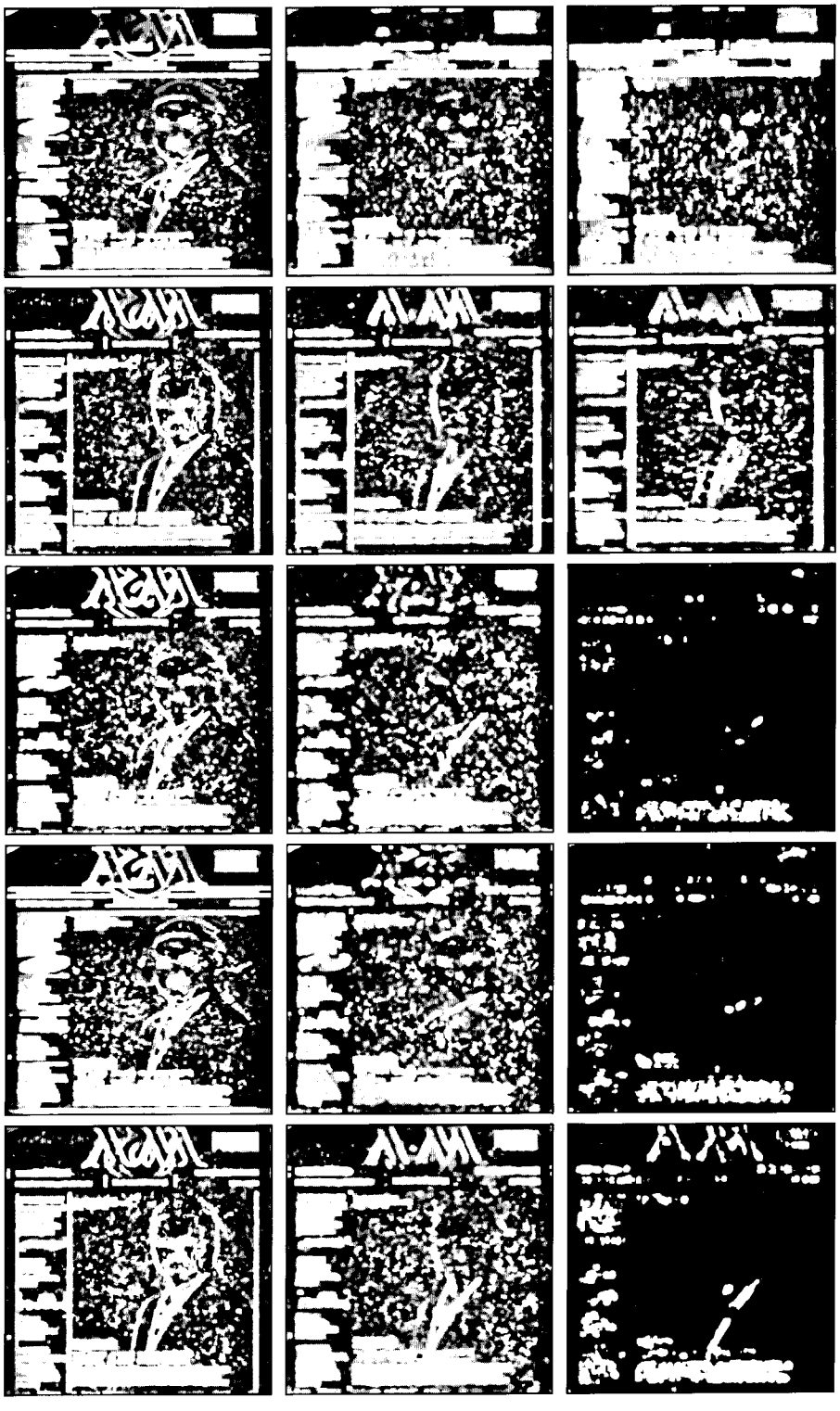


Figure 7.12: Feature images for the document image in Fig. 7.11. Top row - $Feat_{hor_j}$; second row - $Feat_{ver_j}$; third row - $Feat_{diag_j}$; fourth row - $Feat_{hdiag_j}$ and last row - $Feat_{vdiag_j}$, for $j = 1, 2, 3$.

Figure 7.13: (a) Segmentation results of the test image used in [130] using (b) our algorithm (c) Randen and Husøy [130] (d) Etemad *et. al.* [58]

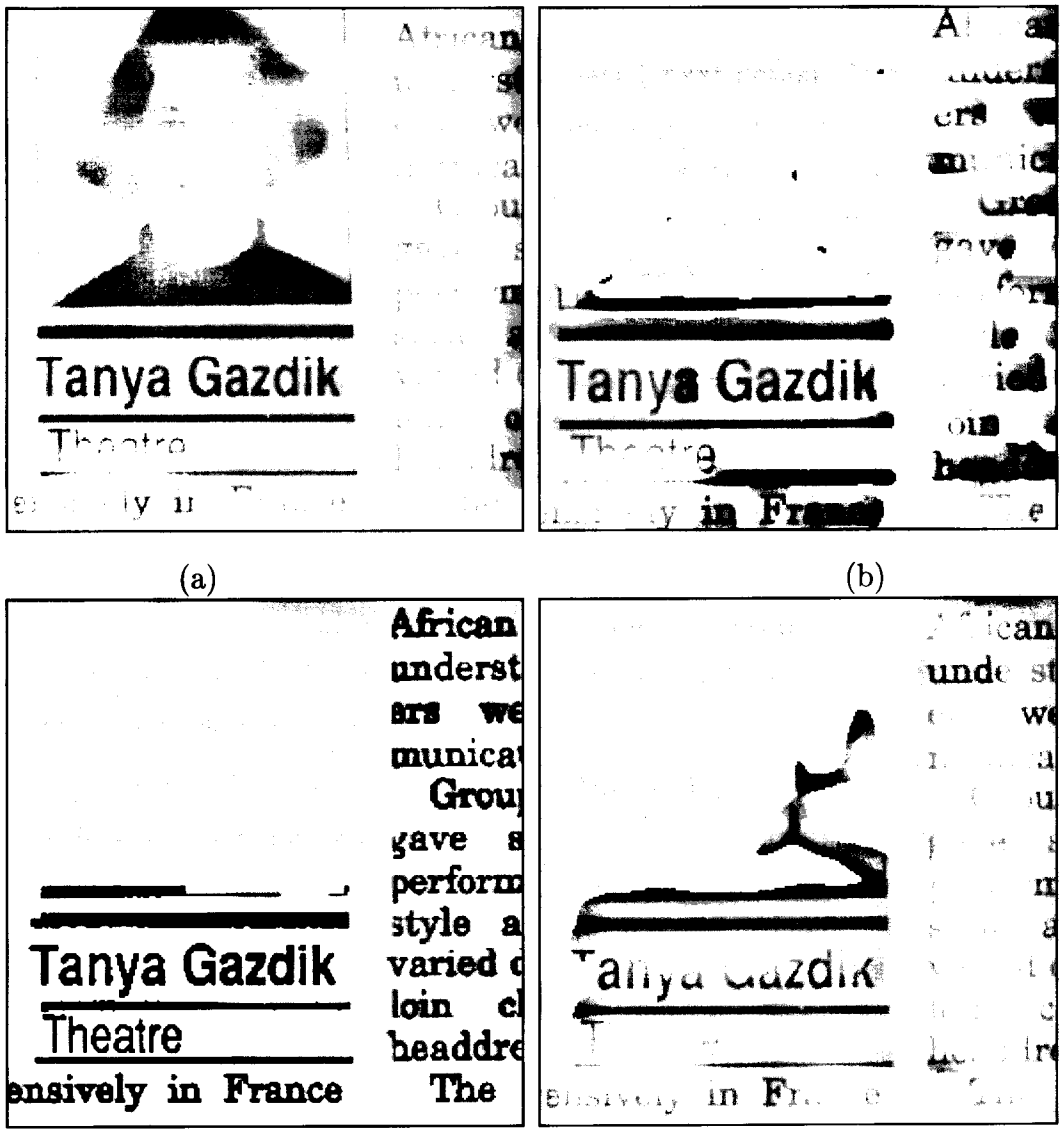


Figure 7.14: (a) Segmentation results of the test image used by Jain and Bhattacharjee in [86] using (b) our algorithm (c) Randen and Husϕy [130] (d) Etemad *et al.* [58]

We have experimented on a variety of data having different combinations of font sizes of the text, we have also considered a diverse variety of test images which are highly unstructured. In all of these data it has been possible to reliably classify the text parts from the non-text parts.

Unlike [130] which uses a subsampling of the image for extraction of features and which introduces some edge inaccuracies, since our technique uses overcomplete representation of the image meaning we have decomposed the image without downsampling, we get accurate edge localization and translation invariance can also be achieved [164].

Although the method used in [130] has considerable computational savings than the approach of Jain and Bhattacharjee [86], none of the above mentioned works have considered test data that are unstructured, or have overlapping text and non/text regions. Where as the work of Etemad *et al.* [58] have considered all these types of data, but they have used fuzzy local decision information for classification. This step clearly reduces ambiguity between the various classes and hence gives better performance. But we have shown that the features that we have extracted are appreciably authentic and gives good enough segmentation. We however conjecture that incorporation of fuzziness in the features that we have extracted would have produced much better results than that reported in this chapter.

It is to be noted that all our experiments are done with no *a priori* knowledge about the input image. We do not have any information about the font size, or format of the text. While a knowledge about these can definitely improve the segmentation results, and for this we can make use of supervised segmentation.

7.3 Segmentation of Remotely Sensed Image

The segmentation of different land cover regions of a remotely sensed image has been recognized as a complex problem. These images usually have poor illumination and are highly dependent on the environmental conditions. Spatial resolution of these images are also very low, *e.g.*, of the order of $20m \times 20m$. The scene mostly contains many objects (regions), *e.g.*, vegetation, water bodies, habitation, concrete structures, open spaces etc., and these regions are not very well defined because of spatial ambiguities. Moreover, the gray value assigned to a pixel is the average reflectance of different types

of land covers present in the corresponding pixel area. Assigning unique class levels with certainty is thus a problem of remotely sensed images.

Remotely sensed images contain information on a large range of scales and the frequency structure changes throughout the signal (*i.e.,* non-periodic signal). In remote sensing perspective, the resolution of the imagery may be different in many cases, and so it is important to understand how information changes through different scales of imagery. This problem leads naturally to multiresolution type analysis which are most effective using wavelets. Moreover, wavelet theory is well suited in this area of study where signals are complex and non-periodic. Furthermore, wavelets are particularly good at describing a scene in terms of the scale of the textures in it.

In a remotely sensed image, texture is considered to be the visual impression of coarseness or smoothness caused by the variability or uniformity of image tone. These textural properties of a remotely sensed image provide valuable information for segmentation, where different object regions are treated as different texture classes, so this basically boils down to a multi-texture segmentation problem. Note that segmentation of these images is necessary in order to identify regions of vegetation, habitation, water bodies, city area etc. Here we incorporate wavelet based texture segmentation algorithm for identifying the various object regions in a remotely sensed image, relevance of which are described above.

7.3.1 Previous works

Haralick *et al.* [80] have used gray level co-occurrence features to analyze remotely sensed images. They have computed gray level co-occurrence matrices for a distance of one with four directions($0^0, 45^0, 90^0$ and 135^0). For a seven class problem they have achieved 80% classification accuracy. Rignot and Kwok [137] have analyzed SAR (Synthetic Aperture Radar) images using texture features computed from gray level co-occurrence matrices. However, they supplement these features with knowledge about the properties of SAR images. The use of various texture features have been studied for analyzing SAR images by Du [54]. He has used Gabor filters for extracting texture features and has successfully segmented SAR images into categories of water, new forming ice, older ice, and multi-year ice.

The work of Mecocci *et al.* [112] present a wavelet-based algorithm combined with a *fuzzy c-means* classifier. Lindsay *et al.* [105] have used the 1D discrete wavelet transform (DWT) based on Daubechies wavelet filter. A wavelet-based texture feature set is derived in [66]. It consists of the energy of subimages obtained by the overcomplete wavelet decomposition of local areas in SAR images, where the downsampling between wavelet levels is omitted.

Simard *et al.* [148] study the use of a decision tree classifier and multiscale texture measures to extract thematic information on the tropical vegetation cover from the Global Rain Forest Mapping (GRFM) JERS-1 SAR mosaics. The aim of the work by [118] is to show how coastline can be derived from SAR images by using wavelet and active contour methods. In a first step an edge detection method suggested by Mallat *et al.* [109] is applied to SAR images to detect all edges above a certain threshold. A block-tracing algorithm (BA) then determines the boundary area between land and water.

Several other wavelet-based segmentation for geoscience and remote sensing applications have also been reported in the literature [162, 160].

Other approaches to segmentation of remotely sensed images include various fuzzy thresholding techniques reported in [124]. Genetic algorithm as a classifier has been investigated in the domain of satellite imagery for partitioning different land cover regions from satellite images, having complex/overlapping class boundaries in [19]. Muchoney and Williamson [115] have shown neural network classifiers to provide supervised classification results that significantly improve on traditional classification algorithms such as the Bayesian (maximum likelihood [ML]) classifier.

All of these above methods use supervised classification where *a priori* knowledge about the images are supplied. We apply the methodology developed in Chapter 6 to carry out this segmentation where no *a priori* knowledge about the image is available.

7.3.2 Segmentation of *IRS* and *SPOT* images

In this section, the results on implementation of our algorithm on several remotely sensed images are provided. The images considered are two IRS-1A images and one SPOT image.

Data sets

IRS images : The IRS-1A images (Figs. 7.15 & 7.16 and Figs. 7.19 & 7.20) were obtained from Indian Remote Sensing Satellite which is a circular sun-synchronous satellite, rotating around the earth at the rate of 14 orbits per day, at an altitude of 904 km and a repetition cycle of 22 days (NRSA 1986). This satellite is equipped with two different sensors LISS (Linear Imaging Self Scanner)-I and LISS-II. LISS-I has a spatial resolution of 72.5m × 72.5m while that for LISS-II is 36.25m × 36.25m. The IRS-1A images used for this work were taken using the scanner LISS-II in the wavelength range $0.45\mu m$ - $0.86\mu m$. The whole spectrum range is decomposed into four spectral bands namely,

- band 1 - blue band of wavelength 0.45μm - 0.52 μm,
- band 2 - green band of wavelength 0.52μm - 0.59 μm,
- band 3 - red band of wavelength 0.62μm - 0.68 μm and
- band 4 - near infra red wavelength 0.77μm - 0.86 μm.

The images in Figs.7.15(a)-7.15(c) & 7.16, cover an area around the city of Calcutta in these four bands. In all these figures the prominent black stretch across them is the river *Hoogly*. There is a prominent light patch on the bottom right corner, which is the *Salt Lake stadium* and the black patches nearby are the *fisheries*. In the upper right part of the images, there is a distinct line structure corresponding to the *airport runway*. In total there are five major classes in which the regions of Calcutta IRS images can be classified. These are *water bodies* (WB), *vegetation* (VEG), *habitation* (HAB), *city area* (CA) and *open spaces* (OS).

Figs. 7.19(a)-7.19(c) & 7.20, show a part of the city of Bombay in the four aforesaid bands respectively. The elongated city area is surrounded by the Arabian sea. There is a *concrete structure* (on the right side top corner) connecting Bombay to New Bombay. On the southern part of the city, there are several islands including the famous *Elephanta islands*. The *dockyard* is situated on the south-eastern part of Bombay, which can be seen as a set of three finger like structure. On the upper part of the images, towards left, there is a distinct crisscrossed structure, which corresponds to the *Santa Cruz airport*. In total the regions of Bombay images can be classified into six major

classes, *viz.*, *turbid water 1* (TW1), *turbid water 2* (TW2), *concrete* (CONCR), *habitation* (HAB), *vegetation* (VEG) and *open spaces* (OS). The sea water is decomposed into two classes TW1 and TW2 for better classification since they have somewhat different reflectance properties due to variation in sea water density (as seen in Fig. 7.20).

SPOT image : The SPOT image was taken by the French Satellites SPOT (Systems Probataire d' Observation de la Terre) [136], launched in 1986 and 1990. They carry two imaging devices that consist of a linear array of charge coupled device (CCD). Two imaging modes are possible, the multispectral and panchromatic modes. The image considered here has three bands in the multispectral mode. These are:

- band 1 - green band of wavelength 0.50μm - 0.59 μm,
- band 2 - red band of wavelength 0.61μm - 0.68 μm and
- band 3 - near infra red band of wavelength 0.79μm - 0.89 mum.

Figs. 7.23(a)-7.23(b) & 7.24 are SPOT images of Calcutta and were taken in these three bands with a spatial resolution of $20.0m \times 20.0m$. In all these figures the prominent black stretch across them corresponds to the river *Hoogly*. There are two distinct black, elongated patches below the river, on the left side of the images. These are water bodies, the one to the left being *Garden Reach Lake* and the one to the right being the *Khiderpore dockyard*. Just to the right of these water bodies, there is a thin line starting from the right bank of the river and going to the bottom edge of the images, which corresponds to *Talis nala*. On the right side of the images, there is a triangular patch which is the *race course*. On the top right hand side of the images, there is a thin line, stretching from the top edge and ending in the middle of the image, which corresponds to the *Beleghata canal*. A bridge called the *Rabindra setu*, cuts the river near the top of the images. This image has seven major classes, which are *turbid water* (TW), *pond water* (PW), *concrete* (CONCR), *vegetation* (VEG), *habitation* (HAB), *open space* (OS) and *roads/bridges* (B/R).

All the images considered in this investigation are of size 512 $\times$ 512. Due to poor illumination, the actual object classes present in the input images are not visible clearly. So we have presented the histogram equalized images in Figs. 7.15(a)-7.15(c) & 7.16, 7.19(a)-7.19(c) & 7.20 and 7.23(a)-7.23(b) & 7.24, which highlight the different land cover regions. But the algorithms are implemented on the actual inputs.

Experimental results

Here we demonstrate the effectiveness of our methodology over several remotely sensed images as described in Section 7.3.2. In order to validate the importance of neuro-fuzzy feature evaluation, we show how the feature dimensionality can be greatly reduced after feature evaluation. The test images have several fine (line) structures (roads and bridges). In order to detect these structures the local window size has been kept small (3×3).

The total number of features considering all the bands of IRS-1A Calcutta image, are found to be 10 using the proposed feature extraction methodology described in Section 6.2. Fig. 7.17 shows the segmented output of the IRS-1A image using all these 10 features considering the number of classes to be $c = 5$. The β index is found to be 3.65887. The neuro-fuzzy feature evaluation algorithm reduces the number of features to only one for which the segmentation result is given in Fig. 7.18. The value of β index is found to be 3.83431 signifying an improvement in segmentation quality. The *stadium* and *fisheries* as well as the *airport runway* are quite distinctly discernible in the segmented output (Fig. 7.18) as compared to Fig. 7.17. We can comment on this finding that since we have considered all the 4 bands for feature extraction, most of the features so extracted contain very less information and hence can be discarded. It can also be inferred that out of these 4 bands only one band is significant and contains most of the image information. It is also evident from the Figs. 7.15(a)-7.15(c) & 7.16 that the image corresponding to the near infra red band (Fig. 7.16) is the most significant band. The feature with which we have obtained the above segmentation result after neuro-fuzzy evaluation corresponds to band-4 and validates our comment. So we can easily disregard all the other bands and keep only the band-4 image for feature extraction. Comparing Figs. 7.17 and 7.18 we find that we get better segmentation output in the *Salt lake stadium* area, also the compactness of the various classes present in the IRS-1A Calcutta image increases after the feature evaluation step.

The segmentation result of the IRS-1A image of Bombay is shown in Fig. 7.21 considering number of classes to be c=6, and the β value is found to be 8.01462. The number of extracted features from all the 4 bands is found to be 12 as obtained by Section 6.2. The feature dimensionality is reduced to one after neuro-fuzzy feature evaluation. As in the previous case, β index (8.21309) shows better segmentation quality. In the

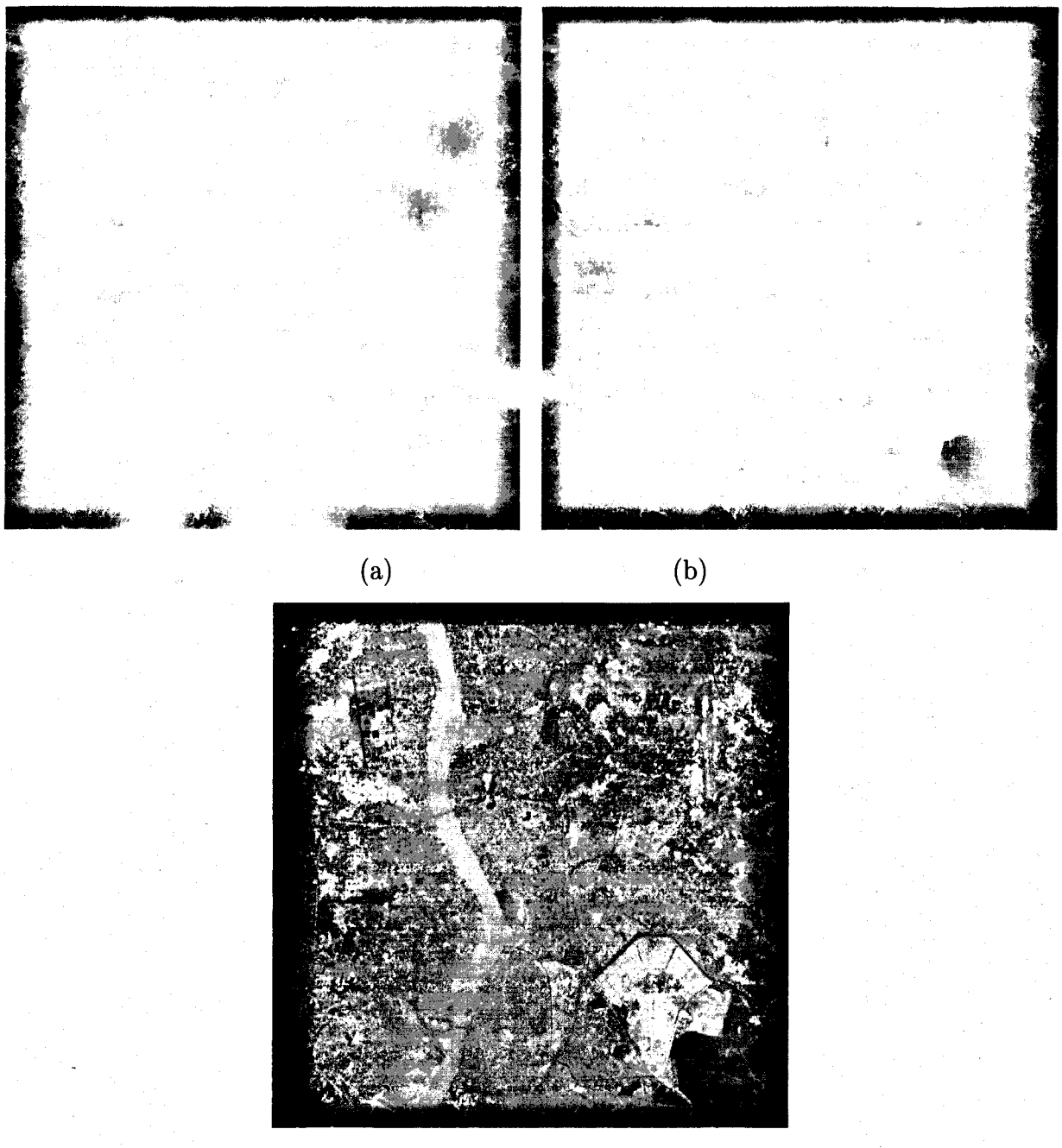


Figure 7.15: IRS-1A image of Calcutta a) band-1 b) band-2 c) band-3

Figure 7.16: IRS-1A image of Calcutta band-4

Figure 7.17: Segmented output of Calcutta IRS-1A image 7.15a-d. with $c = 5$ and $\beta = 3.65887$

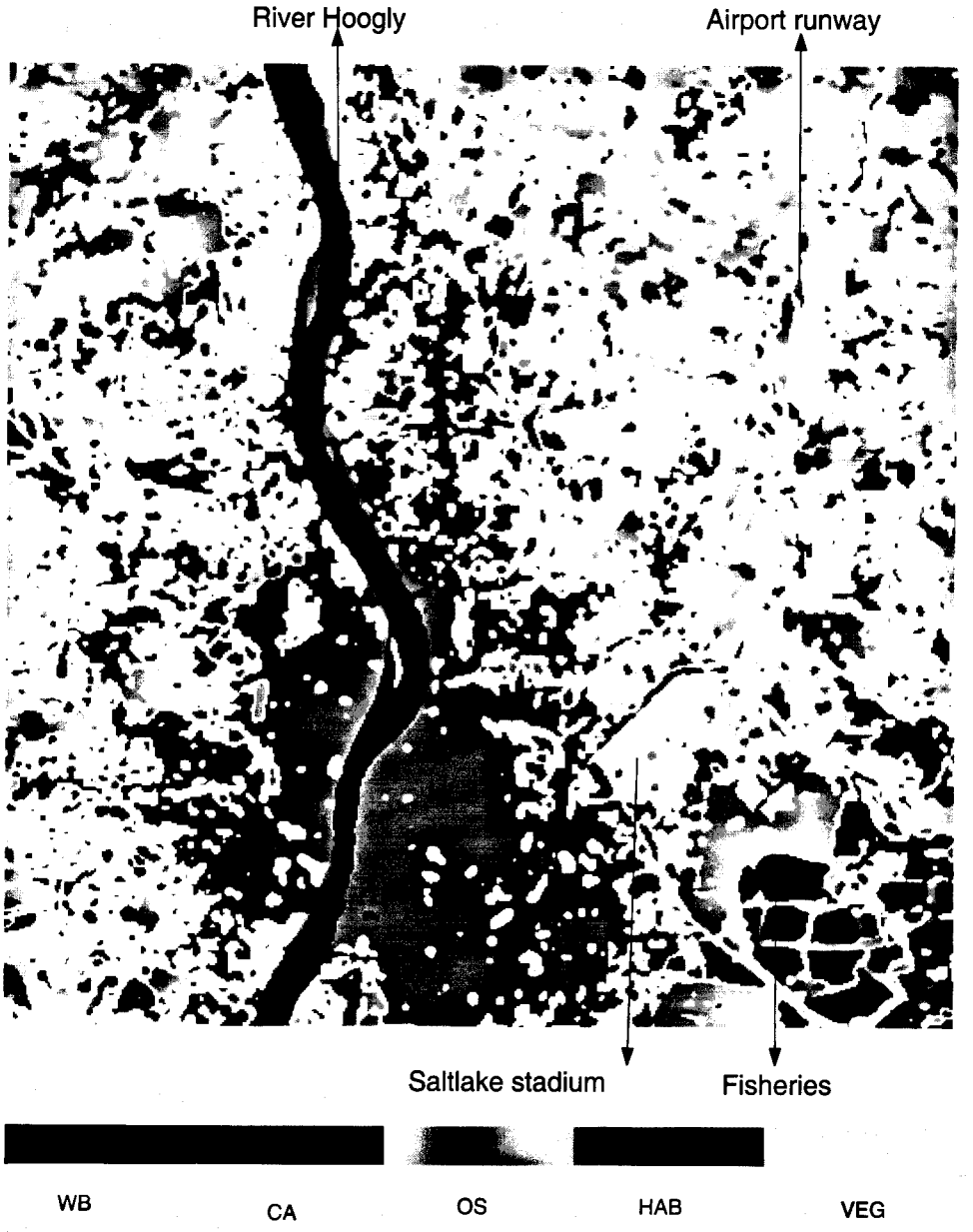


Figure 7.18: Segmented output of Calcutta IRS-1A image 7.15(a)-(d). with $c = 5$ and $\beta = 3.84578$

segmented output after the feature evaluation step (Fig. 7.22), it can be seen that the *dockyard*, the *concrete structure* connecting Bombay and New Bombay and *Santa Cruz airport* are very well detected. Even for this image of Bombay we can reiterate the above comments. Here also the feature obtained after the feature evaluation step corresponds to band-4 image and contains most of the image information. Comparing Figs. 7.21 and 7.22 we find that various objects and regions (*santa cruz airport, dockyard, roads and bridges* etc.) are identified more prominently. Moreover, the feature evaluation step increases the compactness of the various classes similar to the case of IRS-1A Calcutta image.

In the case of SPOT image of Calcutta (Figs. 7.23(a)-7.23(b) & 7.24), we have initially considered all the 10 features extracted by the wavelet based method. The segmentation result is given in Fig. 7.25 with c=7 and a β value of 3.27832. After the neuro-fuzzy feature evaluation the number of features reduces to three. The segmentation result is given in Fig. 7.26, and β value (3.45361) indicates better segmentation quality. It is to be noted that not only the *race course* but also a triangular outline (which is an open space) corresponding to the *track* of the *race course* are also detected. Here the *Rabindra Setu* has also been detected along with the *Talis nala* and *Beleghata canal.* Moreover, the *Khiderpore dock* and *Garden Reach Lake* are quite discernible. In this case all the three bands are important and furnish some information. The features obtained after the feature evaluation step correspond to each of the three bands. Comparing Figs. 7.25 and 7.26, it can be seen that the *race course* can be identified quite vividly (Fig. 7.26). In this case also the compactness of the various classes increases after the feature evaluation step.

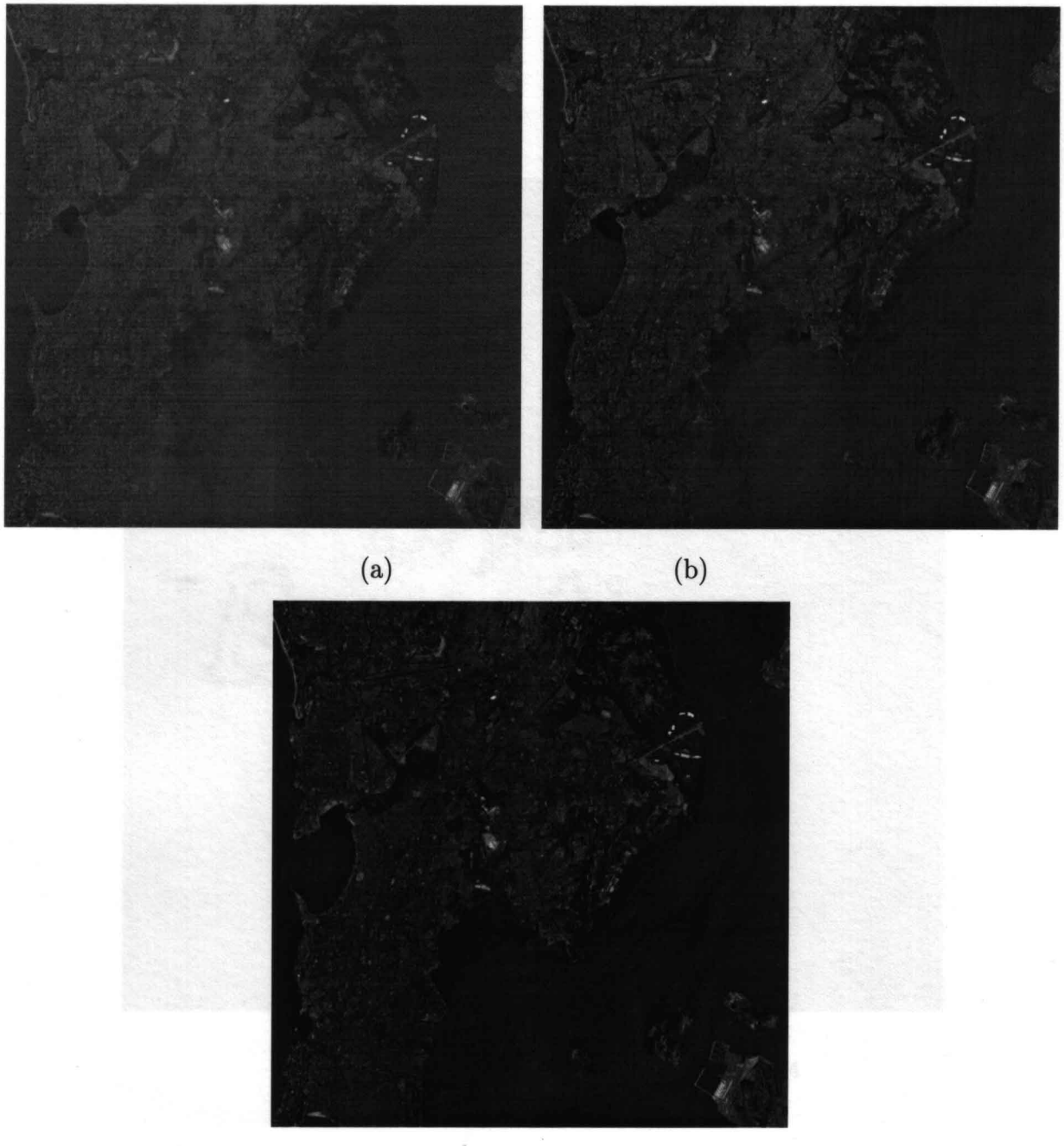


Figure 7.19: IRS -1A image of Bombay a) band-1 b) band-2 c) band-3

Figure 7.20: IRS -1A image of Bombay band-4

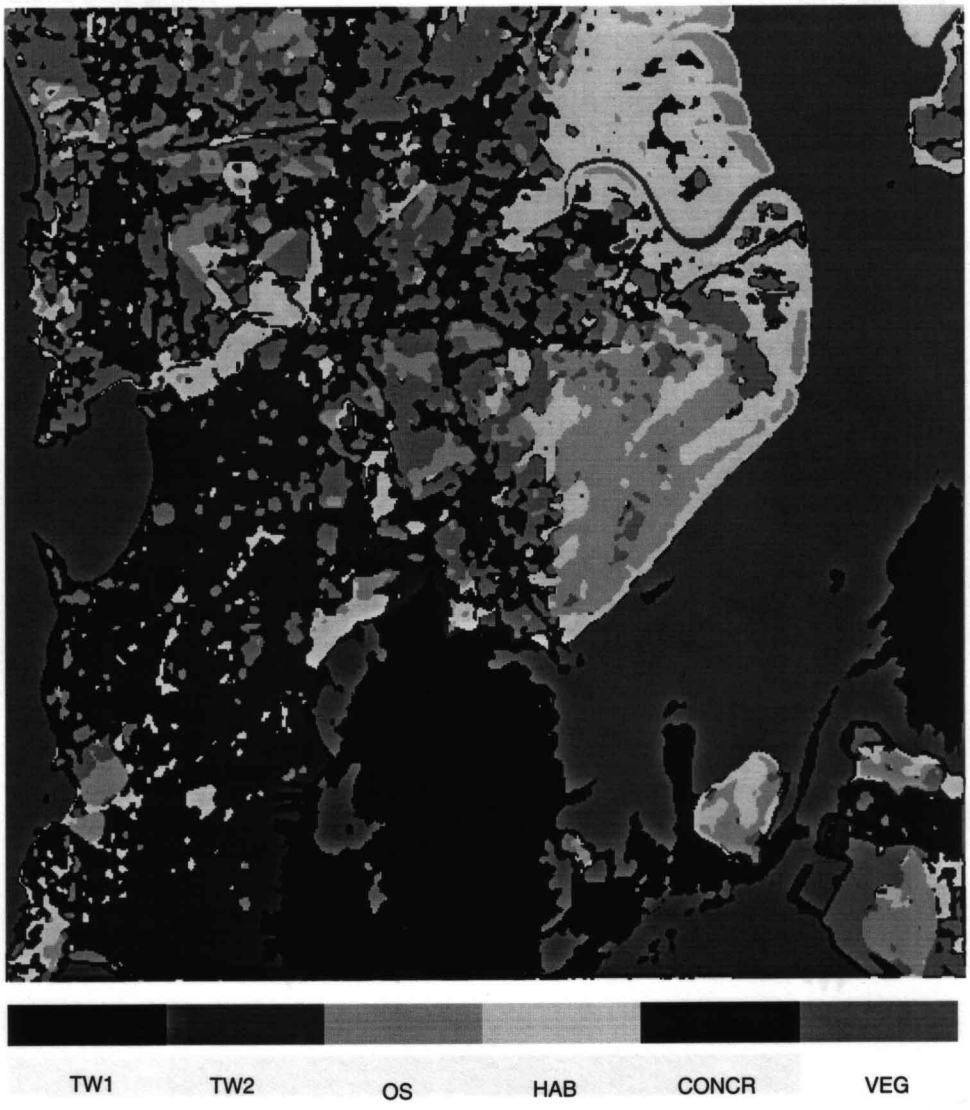


Figure 7.21: Segmented output with $c = 6$ and $\beta = 8.01462$

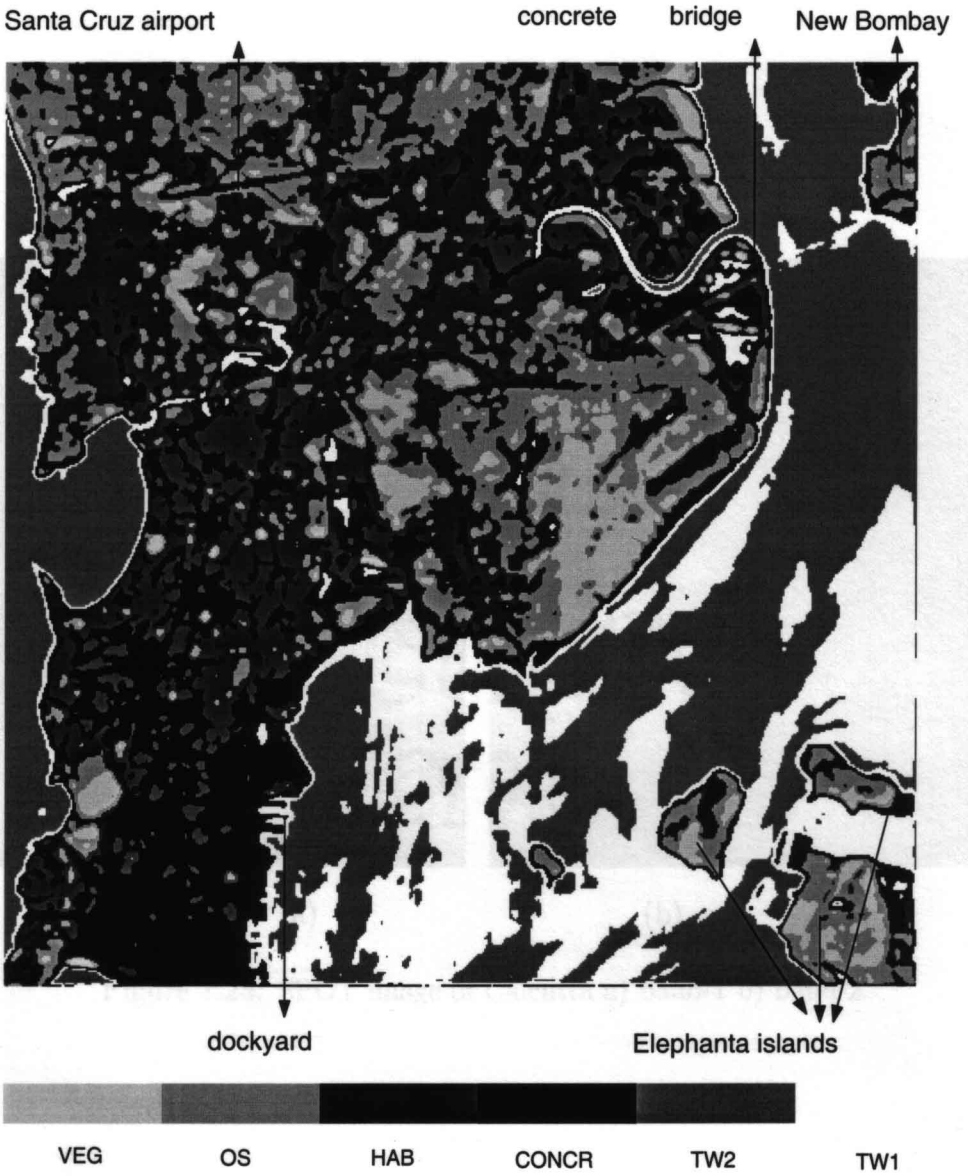
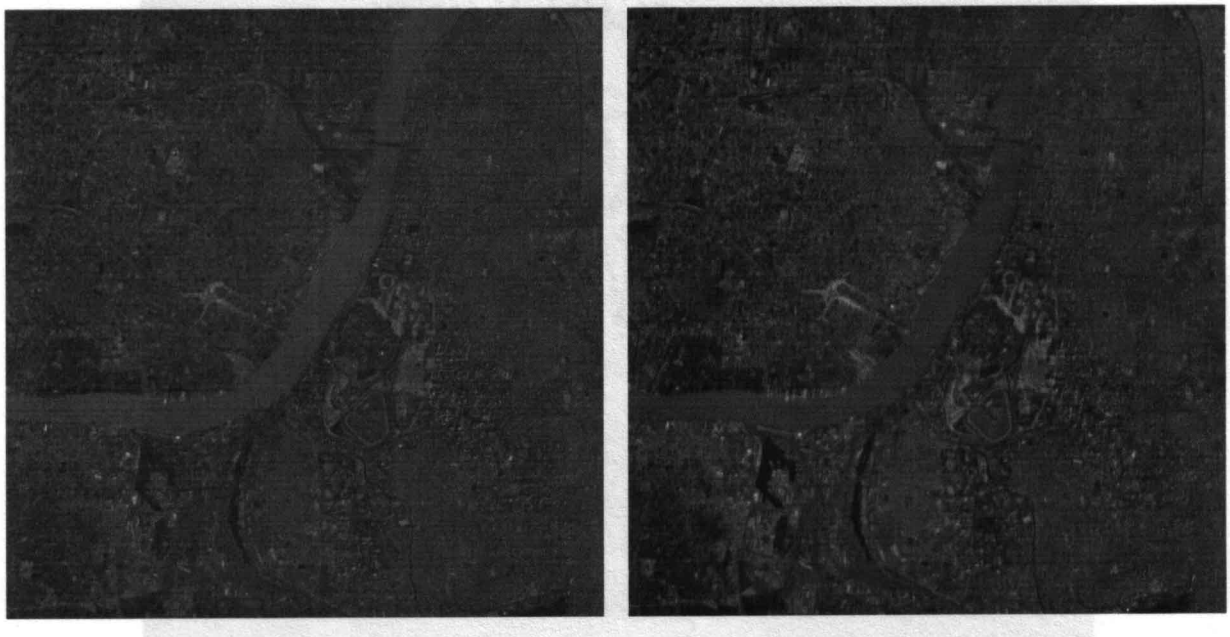


Figure 7.22: Segmented output with $c = 6$ and $\beta = 8.21309$



(a) (b)

Figure 7.23: SPOT image of Calcutta a) band-1 b) band-2

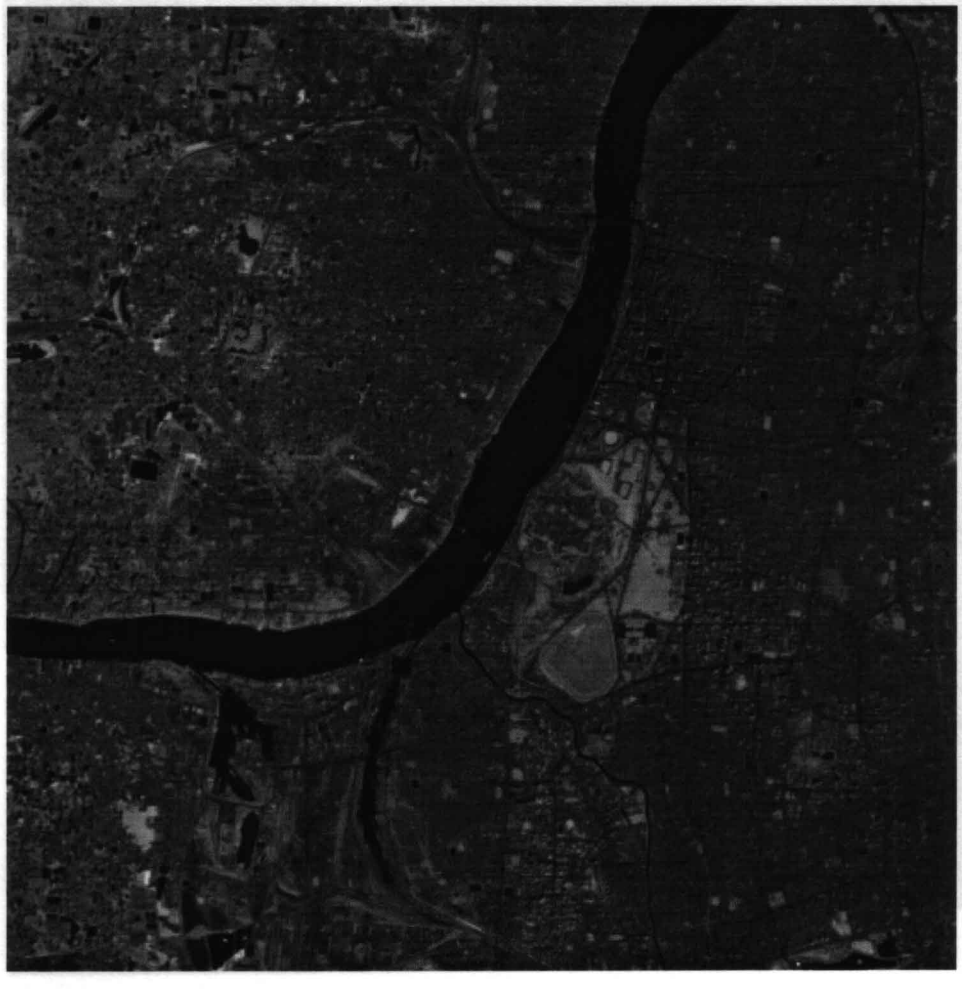


Figure 7.24: SPOT image of Calcutta band-3

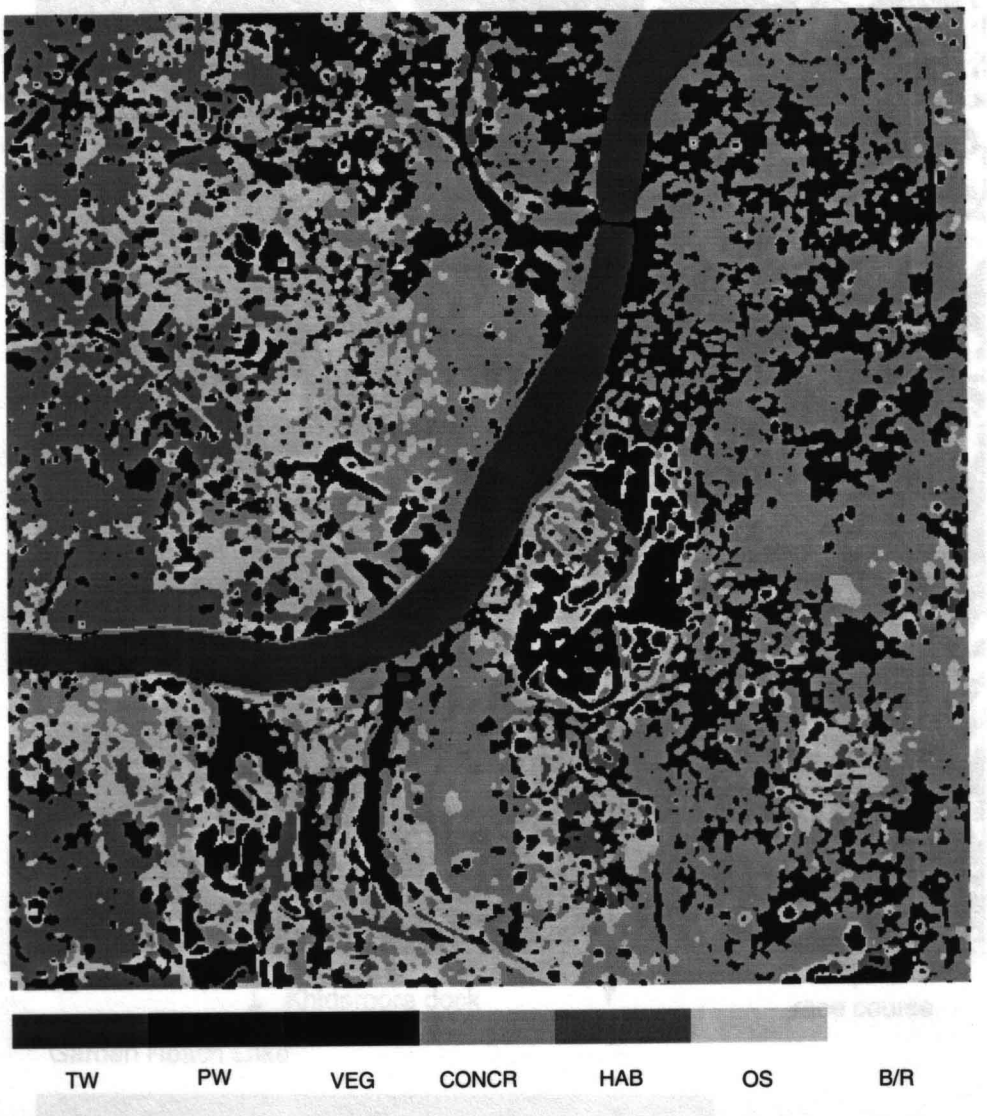


Figure 7.25: Segmented output with $c = 7$ and $\beta = 3.27832$

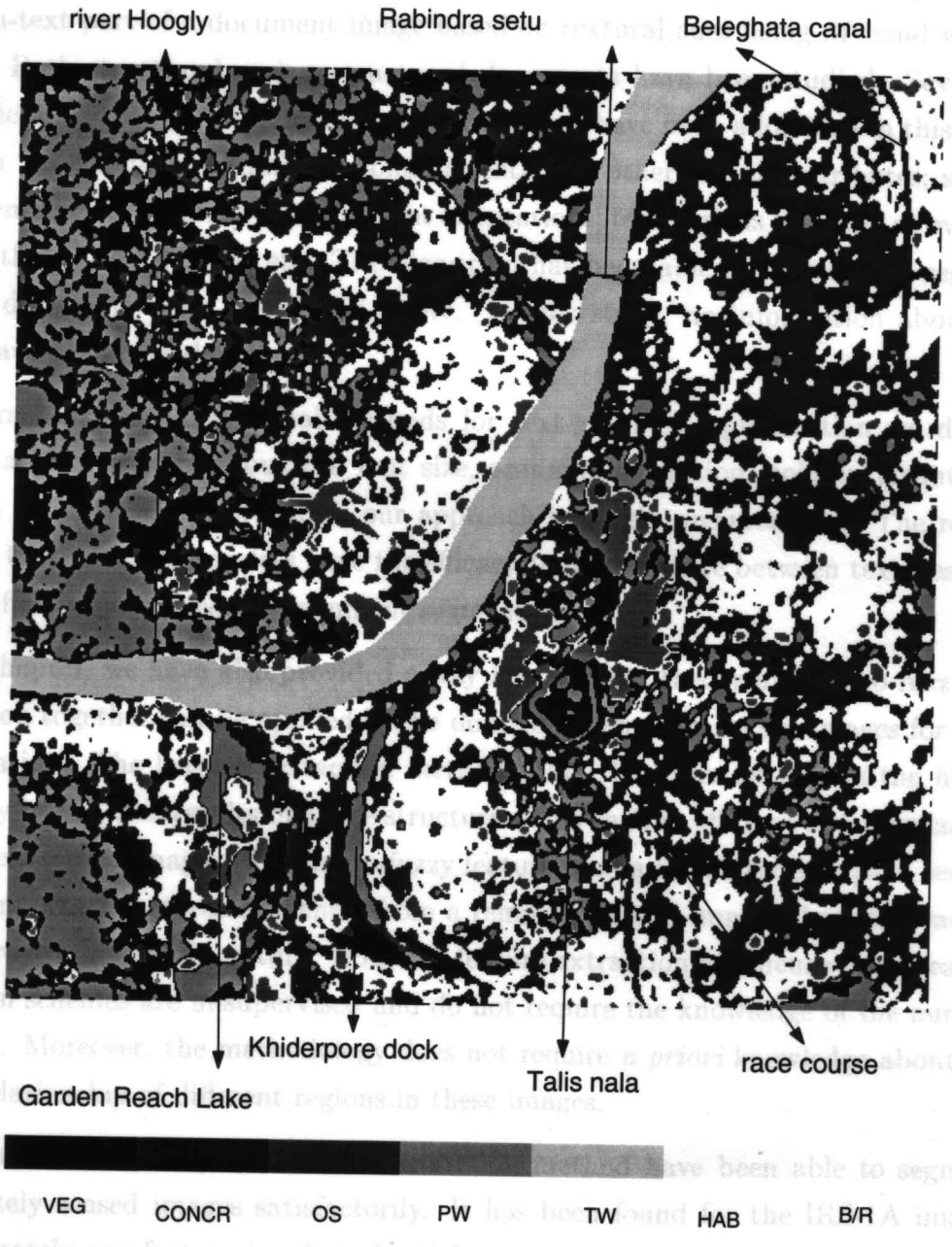


Figure 7.26: Segmented output with $c = 7$ and $\beta = 3.45631$

7.4 Conclusion

In this chapter we have presented a technique for segmenting out the text part from the non-text part of a document image based on textural cues using M-band wavelet filters. Both structured and unstructured documents have been studied. Several of the difficulties that occur in document segmentation have been addressed in this investigation. These include: different fonts, orientation, other textual attributes; skewed document and text regions have different orientations; text regions touching or overlapping with non-text regions etc. The algorithm has been able to successfully segment several document types mentioned above. In this study, no information about the input have been considered.

In contrast to most traditional methods for text-graphics segmentation we do not assume any knowledge about the font size, scanning resolution, column layout, orientation etc. of the input, that is our approach is purely unsupervised. The results indicate that M-band wavelets have the efficacy to discriminate between textures, and can be effectively applied for document segmentation.

In this chapter, we have also provided a way how wavelet theory and neuro-fuzzy hybridization together can be applied in the domain of remotely sensed images for their segmentation. The feature extraction method splits the lower as well as the higher frequency bands, and results in a tree structure. This enables efficient characterization of remotely sensed images. The neuro-fuzzy feature evaluation method helps in searching for important features efficiently from a remotely sensed image where the various classes are overlapping in nature. Both the feature extraction and neuro-fuzzy feature evaluation schemes are unsupervised and do not require the knowledge of the number of classes. Moreover, the methodology does not require *a priori* knowledge about the spatial relationship of different regions in these images.

The features obtained by the feature extraction method have been able to segment the remotely sensed images satisfactorily. It has been found for the IRS-1A images that ultimately one feature is selected which corresponds to the band 4 of these images. However, for the SPOT image three features have been selected. Almost all the desired classes have been obtained in the segmented output satisfactorily. Moreover, the compactness of the various class regions increases after the neuro-fuzzy feature evaluation step.

Chapter 8

Conclusions and Scope for Further Research

8.1 Conclusions and Discussions

The present thesis has demonstrated the advantages of the multiresolution processing ability of wavelets for improved texture analysis, by developing some new algorithms for texture feature extraction. The basic principle of extraction of these features is based on linear filtering and estimation of local energy of the filter response. The features so extracted have been used for both classification and segmentation tasks.

In the first part of our investigation, we have utilized the widely used 2-band wavelet frame decomposition for characterizing textures. For robust texture analysis, features characterizing a texture should be invariant to rotation, translation, gray-scale variations of it & tolerant to noise corruption. The features so obtained perform appreciably well as far as robustness of texture classification and segmentation is concerned.

Chapters 3-4, involve the above studies. Simple and computationally efficient features have been extracted based on the fact that circularly symmetric wavelet gives rotationally invariant features, as these wavelets are independent of orientation. Non-separable wavelet transformations (leading to frames) are shown to be more suitable over the separable mode for achieving rotation and translation invariance. Simple modification of the methodology for extraction of rotation and translation invariant features incorporates an additional property of gray-scale invariance. This is authenticated experimentally. Apart from this, experimental studies have shown high noise tolerance of our methodology. This demonstrate the appropriateness of wavelet frame analysis (DWF) over the standard DWT.

Classification performance of the extracted feature set has been compared with features extracted using DWPT. Comparison clearly establishes the suitability of overcomplete wavelet frames for texture characterization with respect to all the above mentioned invariance. Classification errors have mainly occurred in textures that are almost difficult to discriminate visually and which exhibit high anisotropy. Segmentation performance using these features are also good except some edge inaccuracies. One reason is lack of knowledge about the proper size of local window (a local windowing is involved for extraction of features). A judicious choice of the local window size is essential in order to capture the local periodicity of the micro-textures and to localize the boundaries between texture classes accurately. An important aspect of the feature extraction algorithm is that the number of features used is small, compared to that

used by other related techniques reported in the literature. In essence we have been able to achieve a considerable reduction in feature dimensionality.

We can therefore conclude that, texture signatures based on multiresolution wavelet frames analysis holds great potential for accomplishing subtle discrimination and robust classification against rotation, translation, gray-scale variations and noise and is computationally simple and efficient. One drawback of the extracted features is that the segmentation accuracy is not very satisfying. Another limitation is that, it can not efficiently segment images consisting of a large number of classes.

The second part of our work dealt with texture characterization for segmentation, with an aim to enhance the segmentation quality. For this purpose we have used alternative (non-octave) decompositions using M ($M > 2$) band wavelet. A larger value of $M(M > 2)$ produce a more flexible tiling of the scale-space plane (compared to 2-band wavelet), by giving a mixture of a logarithmic and linear frequency resolution. Also the M-band wavelet decomposition yields a large number of subbands, which characterize a texture better in terms of its constituent energies. Therefore, improved segmentation accuracy has been achieved.

In Chapter 5, two-class texture segmentation problem using M-band wavelet (M=4) has been investigated. The M-band wavelet decomposition of a texture image gives an efficient representation in terms of frequencies in different directions and resolutions. The image is decomposed into $M \times M$ subbands, features are derived (within a local neighborhood of every pixel) from various combinations of these subbands. An adaptive selection of the local window size, based on the spectral frequency content of the images has been developed. The extracted features have been ranked according to a feature quality index and a few of them selected based on some performance criterion to reduce feature redundancy. Superiority of performance of the extracted features over those obtained by several existing methods has been shown. This has been further verified with segmentation results on various texture images including some benchmark images containing overlapping texture classes.

It is shown that the use of M-band wavelet decomposition of the texture image gives an efficient representation of the image which facilitates an improved segmentation of the different texture regions and allows it to be rotation invariant. The filtering and the feature extraction operations account for major computational cost, however our

methodology is very simple and computationally less expensive in comparison to some related existing techniques. The number of features required are quite small compared to some related ones reported in [132].

In Chapter 6, a multi-class texture segmentation technique using M-band wavelet packet frames has been developed. It is found that a further recursive decomposition of the high frequency regions characterize a texture more efficiently. The present investigation establishes the potentiality of an *M-band wavelet packet* transform for efficient texture characterization in terms of its constituent signal energies. Instead of going for a full decomposition tree, significant bases have been selected adaptively. A neuro-fuzzy approach under unsupervised learning, for selecting a near optimal set of features, from those obtained after the selection of the bases is developed.

The features obtained by the feature extraction method have been able to segment different benchmark test images satisfactorily. The neuro-fuzzy feature evaluation method has reduced the number of extracted features greatly and thereby enhances the quality of segmentation with only a few number of features. The superiority of our methodology in segmenting the texture images to that of others obtained by various existing methods has been shown both qualitatively and quantitatively. This implicates that in our method, dimensionality of the feature space can be considerably reduced. while still maintaining a good segmentation quality. It has been however found for certain typical cases that, similar segmentation results are obtained even without the neuro-fuzzy evaluation step.

An important point to be noted is that, except for the knowledge about the total number of classes present in a composite image no other *a priori* information (spatial information about the pixel labeling) about the test images are required in all our algorithms.

Chapter 7, puts together results of investigation obtained so far on two real life applications. In the first part of the chapter, an efficient and computationally fast method for segmenting text and graphics part of document images based on textural cues has been developed. Both structured and unstructured documents have been studied. Several of the difficulties that occur in document segmentation have been addressed in this investigation. These include: different fonts, orientation, other textual attributes; skewed document with text regions having different orientations; text regions touching or over-

lapping with non-text regions etc. The algorithm has been able to successfully segment several document types mentioned above. In this study, no *a priori* information about the input image has been considered.

The later part of the chapter involves studies on the effectiveness of the features extracted in Chapter 6 for segmenting real life data, such as remotely sensed images. The efficacy of the methodology is demonstrated on two 4-band IRS-1A images and a 3-band SPOT image. The neuro-fuzzy feature evaluation method has been able to select important features efficiently. Almost all the desired classes are obtained in the segmented output satisfactorily without any *a priori* knowledge about the spatial relationship of different regions. The compactness of the various class regions have been found to increase after the neuro-fuzzy feature evaluation step.

8.2 Scope for Further Research

In the present thesis, classification technique has been restricted to the unsupervised mode. Even without any *a priori* knowledge about the input, satisfactory classification performances have been obtained. To improve the performance further, supervised learning can be used. Data having complex structures like, overlapping classes, *e.g.*, mixed/unstructured document and remotely sensed images, can presumably be classified more correctly under supervised mode.

We have worked with $M(=4)$-band wavelet in the present thesis. An interesting extension to this work can be the design of wavelets for odd values of M (*i.e.*, $M = 3,5$ etc.). A choice of odd value of M may make the scaling filter irregular, *i.e.*, it may not retain its low pass nature, rather it may have a combination of low pass and high pass responses in the frequency domain. This may be beneficial for characterizing textures more efficiently in some cases. Since textures have a wide range of structural variability, we conjecture that, irregularity in the scaling and wavelet functions may characterize random textures in a better way. This requires a detailed theoretical investigation.

Textures have most of their energies concentrated predominantly in the mid-band while, non-textured areas have energies in the low-band portions of the frequency spectrum. On this basis one can investigate a number of non-conventional, non-dyadic wavelet transformations, like the *split trick, maximum overlap wavelet transform* etc.

In our investigation we have not covered all types of texture like, random, aperiodic etc. One can investigate further with all these types of texture and find whether a modification of the methodologies developed is required or not.

Second generation image compression techniques for complex images can be developed based on the results obtained for texture segmentation, for efficient and flexible compression.

The investigation can further be extended to interactive segmentation techniques, which have now become very popular for efficient image retrieval.

Algorithm for document segmentation can be extended to the color domain so that it can efficiently work in a more generic environment and facilitate region based manipulation of the text during type setting.

Appendix A

Operation of the neural network

When pth and qth patterns are presented to the input layer, the activation produced by ith ($1 \leq i \leq 2n$) input node is $v_i^{(0)} = u_i^{(0)}$ where, $u_i^{(0)} = x_{pi}, for\ 1 \leq i \leq n$ and $u_{i+n}^{(0)} = x_{qi}, for\ 1 \leq i \leq n$, the total activations received by ith and $(i+n)$th ($1 \leq i \leq n$) input node, respectively. The total activation received by jth hidden node (connecting ith and $(i+n)$th, $1 \leq i \leq n$, input nodes) is given by

$$u_j^{(1)} = 1 \times v_i^{(0)} + (-1) \times v_{i+n}^{(0)}, \quad for\ 1 \leq i \leq n, \tag{A.1}$$

and the activation produced by it is $v_j^{(1)} = (u_j^{(1)})^2$. The total activation received by the output node which computes μ^T-values, is $u_T^{(2)} = \sum_j W_j v_j^{(1)}$, and that received by the other output node which computes μ^O-values, is $u_O^{(2)} = \sum_j v_j^{(1)}$. Therefore, $u_T^{(2)}$ and $u_O^{(2)}$ represent d_{pq}^2 as given by (6.8, where $0 \leq w_i < 1$ and $w_i = 1$, $\forall i$, respectively. The activations, $v_T^{(2)}$ and $v_O^{(2)}$, of the output nodes represent μ_{pq}^T and μ_{pq}^O for pth and qth pattern pair, respectively. Thus,

$$\begin{aligned} v_T^{(2)} &= 1 - \frac{(u_T^{(2)})^{\frac{1}{2}}}{D}, & if\ (u_T^{(2)})^{\frac{1}{2}} \leq D, \\ &= 0, & otherwise, \end{aligned} \tag{A.2}$$

and

$$\begin{aligned} v_O^{(2)} &= 1 - \frac{(u_O^{(2)})^{\frac{1}{2}}}{D}, & if\ (u_O^{(2)})^{\frac{1}{2}} \leq D, \\ &= 0, & otherwise. \end{aligned} \tag{A.3}$$

The evaluation index (which is computed off the network), in terms of these activations, is then written (from (6.6) as

$$E(\mathbf{W}) = \frac{2}{s(s-1)} \sum_{p} \sum_{q \neq p} \frac{1}{2} [v_T^{(2)}(1 - v_O^{(2)}) + v_O^{(2)}(1 - v_T^{(2)})]. \tag{A.4}$$

The task of minimization of $E(\mathbf{W})$ (A.4) with respect to $\mathbf{W}$ is performed using gradient-descent technique, where the change in W_j ($\triangle W_j$) is computed as

$$\triangle W_j = -\eta \frac{\partial E}{\partial W_j}, \forall j, \tag{A.5}$$

where η is the learning rate.

For computation of $\frac{\partial E}{\partial W_j}$ corresponding to a pair of patterns, the following expressions are used.

$$\frac{\partial E(\mathbf{W})}{\partial W_j} = \frac{1}{2} \left[1 - 2v_O^{(2)}\right] \frac{\partial v_T^{(2)}}{\partial W_j}, \tag{A.6}$$

$$\begin{aligned} \frac{\partial v_T^{(2)}}{\partial W_j} &= -\frac{\frac{1}{2}(u_T^{(2)})^{-\frac{1}{2}} \frac{\partial u_T^{(2)}}{\partial W_j}}{D}, \quad if\ (u_T^{(2)})^{\frac{1}{2}} \leq D \\ &= 0, \qquad\qquad\qquad otherwise, \end{aligned} \tag{A.7}$$

and

$$\frac{\partial u_T^{(2)}}{\partial W_j} = v_j^{(1)}. \tag{A.8}$$

Bibliography

[1] M. Acharyya and M. K. Kundu, "Robust texture classification using wavelet frames," *Image Processing and Communication: An International Journal*, vol. 5, no. 2, pp. 19–37, 1999.

[2] M. Acharyya and M. K. Kundu, "Classification of rotated textures using over-complete wavelet frames," *Journal of IETE for Research, India*, vol. 46, no. 5, pp. 363–370, 2000.

[3] M. Acharyya and M. K. Kundu, "Efficient rotation-invariant feature extraction for texture segmentation via multiscale wavelet frames," in *Proc. Indian Conf. on Computer Vision, Graphics and Image Processing, ICVGIP 2000*, (Bangalore, India), pp. 1–8, December 2000.

[4] M. Acharyya and M. K. Kundu, "Two texture segmentation using M-band wavelets," in *Proc. 15th. Intl. Conf. on Pattern Recognition, ICPR 2000*, (Barcelona, Spain), pp. 405–408, September 2000.

[5] M. Acharyya and M. K. Kundu, "An adaptive approach to unsupervised texture segmentation using M-band wavelet," *Signal Processing*, vol. 81, no. 7, pp. 1337–1356, 2001.

[6] M. Acharyya and M. K. Kundu, "Adaptive basis selection for multitexture segmentation by M-band wavelet packet frame," in *Proc. Intl. Conf. on Image Processing, ICIP 2001*, (Thessaloniki, Greece), pp. 622–625, October 2001.

[7] M. Acharyya and M. K. Kundu, "M-band wavelet analysis for document image segmentation," in *Proc. National Conf. on Intelligent Computing and VLSI, NCICV 2001*, (Kalyani, India), pp. 58–63, February 2001.

[8] M. Acharyya and M. K. Kundu, "Multiscale segmentation of document images using M-band wavelets," in *Proc. 9th Intl. Conf. on Computer Analysis of Images and Patterns, CAIP 2001*, vol. LNCS 2124, (Warsaw, Poland), pp. 510–517. September 2001.

[9] M. Acharyya and M. K. Kundu, "Wavelet-based texture segmentation of remotely sensed images," in *Proc. Intl. Conf. on Image Analysis and Processing, ICIAP 2001*, (Palermo, Italy), pp. 69–74, September 2001.

[10] M. Acharyya and M. K. Kundu, "Extraction of noise tolerant gray-scale transform and rotation-invariant features for texture segmentation using wavelet frames," *IEEE Trans. Systems, Man and Cybernetics*, (Accepted).

[11] M. Acharyya and M. K. Kundu, "Document image segmentation using wavelet scale-space features," *IEEE Trans. Circuits and Systems for Video Technology*, (Accepted).

[12] M. Acharyya, R. K. De, and M. K. Kundu, "Extraction of features using M-band wavelet packet frame and their neuro-fuzzy feature evaluation for multitexture segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, (Revised version submitted).

[13] M. Acharyya, R. K. De, and M. K. Kundu, "Segmentation of remotely sensed images using wavelet features and their evaluation in soft computing framework," *IEEE Trans. Geoscience and Remote Sensing*, (Communicated).

[14] N. Ahuja, and A. Rosenfeld, "Mosaic models for texture," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 3, no. 1, pp. 1–11, 1981.

[15] A. Al-Janobi, "Performance evaluation of cross diagonal texture matrix method of texture analysis," *Pattern Recognition*, vol. 34, pp. 171–180, 2001.

[16] F. Albregtsen, H. Schulerud, and L. Yang, "Texture classification of mouse liver cell nuclei using invariant moments of consistent regions," in *Proc. Conf. on Analysis and Image Processing, LNCS*, pp. 496–502, September 1995.

[17] O. Alkin and H. Caglar, "Design of efficient M-band coders with linear phase and perfect reconstruction properties," *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1579–1590, 1995.

[18] A. Antaonacopulos, "Page segmentation using the description of the background," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 350–369, 1998.

[19] S. Bandyopadhyay and S. K. Pal, "Pixel classification using variable string genetic algorithms with chromosome differentiation," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 303–308, 2001.

[20] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Networks*, vol. 5, pp. 537–550, 1994.

[21] G. Battle, "A block spin construction of ondelettes," *Comm. Math. Physics*, vol. 110, pp. 601–615, 1987.

[22] J. R. Bergen and M. S. Landy, "Computational modeling of visual texture segregation," in *Computational Models of Visual Processing* (M. S. Landy and J. A. Movshan, eds.), pp. 472–481, Cambridge, MA: The MIT Press, 1991.

[23] J. C. Bezdek and P. Castelaz, "Prototype classification and feature selection with fuzzy sets," *IEEE Trans. Systems, Man and Cybernetics*, vol. 7, pp. 87–92, 1977.

[24] J. S. D. Bonet and P. Viola, "Texture recognition using a non-parametric multi-scale statistical model," in *Proc. IEEE Computer Society Conference on CVPR*, (Santa Barbara, CA), pp. 641–647, June 1998.

[25] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Processing*, vol. 39, pp. 2025–2043, September 1991.

[26] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 55–73, 1990.

[27] P. Brodazt, *Textures: A Photographic Album for Artists and Designers.* New York: Dover Publications, 1966.

[28] C. S. Burrus, A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transform: A primer.* Upper Saddle River, NJ (USA): Prentice-Hall Inc., 1998.

[29] P. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Communications*, vol. 31, pp. 532–540, 1983.

[30] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *Journal of Physiology*, vol. 197, pp. 551–566, 1968.

[31] P. H. Carter, "Texture discrimination using wavelets," in *SPIE Applications of Digital Image Processing XIV*, vol. 1567, pp. 432–438, 1991.

[32] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree structured wavelet transform," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 42–44, 1993.

[33] B. B. Chaudhuri, P. Kundu, and N. Sarkar, "Detection of gradation of oriented texture," *Pattern Recognition Letters*, vol. 14, pp. 147–153, 1993.

[34] P. Chauvet, J. Lopez-Krahe, E. Taflin, and H. Maitre, "System for an intelligent office document analysis, recognition and description," *Signal Processing*, vol. 32, no. 1-2, pp. 161–190, 1993.

[35] R. Chellapa, "Two dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition* (L. N. Kanal and A. Rosenfeld, eds.), pp. 79–112, Amsterdam, The Netherlands: North Holland, 1985.

[36] C. C. Chen and D. C. Chen, "Multiresolution Gabor filters in texture analysis," *Pattern Recognition Letters*, vol. 17, no. 10, pp. 1069–1076, 1996.

[37] C. C. Chen and D. C. Chen, "Filtering methods for texture discrimination," *Pattern Recognition Letters*, vol. 20, pp. 783–790, 1999.

[38] J. L. Chen and A. Kundu, "Rotation and gray-scale transform invariant texture identification using wavelet decomposition and hidden Markov model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 208–214, 1994.

[39] P. C. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, pp. 64–69, Jan 1983.

[40] Y. Q. Chen, M. S. Nixon, and D. W. Thomas, "Statistical geometric features for texture classification," *Pattern Recognition*, vol. 28, no. 4, pp. 537–552, 1995.

[41] Y. Chitre and A. P. Dhawan, "M-band wavelet discrimination of natural textures," *Pattern Recognition*, vol. 32, pp. 773–789, 1999.

[42] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models," *IEEE Trans. Image Processing*, vol. 10, pp. 1309–1321, 2001.

[43] W. S. Chou, "Classifying image pixels into shaped, smooth, and textured points," *Pattern Recognition*, vol. 32, pp. 1697–1706, 1999.

[44] M. Clark and A. C. Bovik, "Texture segmentation using Gabor modulation/demodulation," *Pattern Recognition Letters*, vol. 6, pp. 261–267, 1987.

[45] A. Cohen. I. Daubechies, and J. Feauveau, "Bi-orthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.

[46] F. S. Cohen, Z. Fan, and M. A. Patel, "Classification of rotation and scaled textured images using Gaussian Markov random field models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 192–202, 1991.

[47] R. R. Coifman and M. V. Wickerhauser, "Entropy based algorithms for best basis selection," *IEEE Trans. Information Theory*, vol. 38, pp. 713–718, Mar. 1992.

[48] R. G. Congalton, R. G. Oderwald, and R. A. Mead, "Assessing landsat classification accuracy using discrete multivariate analysis statistical techniques," *Photogrammetric Engineering and Remote Sensing*, vol. 49, pp. 1671–1678, 1983.

[49] I. Daubechies, "Orthogonal bases for compactly supported wavelets," *Comm. Pure Appl. Math*, vol. 41, pp. 909–996, 1988.

[50] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: Soc. Ind. Applied Math, 1992.

[51] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, pp. 847–856, 1980.

[52] R. K. De, N. R. Pal, and S. K. Pal, "Feature analysis : Neural network and fuzzy set theoretic approaches," *Pattern Recognition*, vol. 30, pp. 1579–1590, 1997.

[53] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Englewood Cliffs: Prentice–Hall, Inc., 1982.

[54] L. J. Du, "Texture segmentation of SAR images using localized spatial filtering," in *Proc. Intl. Geoscience and Remote Sensing Symposium*, pp. 1983–1986, 1990.

[55] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons, 1973.

[56] D. F. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. Image Processing*, vol. 23, pp. 291–309, 1995.

[57] C. R. Dyer, T. Hong, and A. Rosenfeld, "Texture classification using gray level cooccurrence based based on edge maxim," *IEEE Trans. Systems, Man and Cybernetics*, vol. 10, pp. 158–163, 1980.

[58] K. Etemad, D. Doermann, and R. Chellappa, "Multiscale segmentation of unstructured document pages using soft decision integration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 92–96, 1997.

[59] K. Etemad and R. Chellapa, "Separability-based multiscale basis selection and feature extraction for signal and image classification," *IEEE Trans. on Image Processing*, vol. 7, pp. 1453–1465, Oct. 1998.

[60] F. Farrokhnia, *Multichannel Filtering Techniques for Texture Segmentation and Surface Quality Inspection.* PhD thesis, Michigan State university, 1990.

[61] N. Fatemi-Ghomi, P. L. Palmer, and M. Petrou, "Performance evaluation of texture segmentation algorithms based on wavelets," in *Proc. Workshop on Performance Characteristics of Vision Algorithms, ECCV-96*, April 1996.

[62] M. Flickner, H. Sawhney, W. Niblack, J. Sahley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[63] S. R. Fountain and T. N. Tan, "Efficient rotation-invariant texture features for content-based image retrieval," *Pattern Recognition*, vol. 31, no. 11, pp. 1725–1732, 1999.

[64] J. M. Francos, A. Z. Meiri, and B. Porat, "A unified texture model based on a 2-d wold like decomposition," *IEEE Trans. Signal Processing*, vol. 41, no. 8, pp. 2665–2677, 1993.

[65] K. S. Fu, *Syntactic Pattern Recognition and Applications.* Englewood Cliffs: Prentice–Hall, Inc., 1982.

[66] S. Fukuda and H. Hirosawa, "A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2282–2286, 1999.

[67] K. Fukunaga and L. D. Hostetlo, "Optimisation of k-nearest neighbour density estimates," *IEEE Trans. Information Theory*, vol. 19, pp. 320–326, 1973.

[68] D. Gabor, "Theory of communication," *Journal of Inst. Elect. Engr.*, vol. 93, pp. 429–459, 1946.

[69] A. Gagalowicz, "A new method for texture fields synthesis: Some applications of the study of human vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 3, no. 5, pp. 520–533, 1981.

[70] P. Garcia, M. Petrou, and S. Kamata, "The use of Boolean model for texture analysis of gray images," *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 227–235, 1999.

[71] L. V. Gool, P. Dewaele, and A. Oosterlinck, "Texture analysis anno 1983 : A survey," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 336–357, 1985.

[72] H. Greenspan, R. Goodman, and P. Perona, "Rotation-invariant texture recognition using a steerable pyramid," in *Proc. Intl. Conf. on Pattern Recognition, ICPR '94*, pp. 162–167, 1994.

[73] T. Greiner, J. P. Casel, and M. Pandit, "Texture analysis with a texture matched M-channel wavelet approach," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP '93*, vol. 5, pp. V–129–V–132, 1993.

[74] A. Grossman and J. Morlet, "Decomposition of Hardy functions into square integrable wavlets of constant shape," *SIAM J. Math. Anal.*, vol. 15, pp. 723–736, 1984.

[75] T. O. Gulsrud and S. Kjφde, "Optimal filter for detection of stellate lesions and circumscribed masses in mammograms," in *Proc. SPIE's Visual Communications and Image Processing*, pp. 430–440, March 1996.

[76] F. Guoliang and X. Xiang-Gen, "A joint multicontext and multiscale approach to Bayesian image segmentation," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 2680–2688, 2001.

[77] G. M. Haley and B. S. Manjunath, "Rotation-invariant texture classification using a complete space-frequency model," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 255–269, 1999.

[78] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 768–804, 1979.

[79] R. M. Haralick, "Document image understanding: Geometric and logical layout," in *Proc. of IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, pp. 385–390, June 1994.

[80] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture feature for image classification," *IEEE Trans. Systems, Man and Cybernatics*, vol. 8, no. 6, pp. 610–621, 1973.

[81] G. Harit, S. Chaudhury, P. Gupta, N. Vohra, and S. D. Joshi, "A model guided document image analysis scheme," in *Proc. Sixth Intl. Conf. on Document Analysis and Recognition, ICDAR '01*, pp. 1137 – 1141, 2001.

[82] D. C. He and L. Wang, "Texture features based on texture spectrum," *Pattern Recognition*, vol. 24, no. 5, pp. 391–399, 1991.

[83] T. Hofmann, J. Puzicha, and J. M. Buhmann, "Unsupervised texture segmentation in a deterministic annealing framework," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 803–818, 1998.

[84] X. D. Huang, Y. Akiri, and M. A. Jack, *Hidden Markov Models for Speech Recognition.* Edinburgh, U.K.: Edinburgh Univ. Press, 1990.

[85] J. H. Husøy, *Subband Coding of Still Images and Video.* PhD thesis, Norwegian Institute of Technology, 1991.

[86] A. K. Jain and S. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Applications*, vol. 5, no. 3, pp. 169–184, 1992.

[87] N. Jayant and P. Noll, *Digital Coding of Waveforms : Principles and Applications to Speech and Video.* Englewood Cliffs, New Jersy: Prentice Hall Publications, 1984.

[88] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing, ICASSP '80*, pp. 291–294, IEEE 1980.

[89] B. Julesz, "Textons, elements of texture perception and their interactions," *Nature*, vol. 290, pp. 91–97, 1981.

[90] L. M. Kaplan, "Extended fractal analysis for texture classification and segmentation," *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1572–1585, 1999.

[91] R. L. Kashyap and A. Khotanzad, "A model-based method for rotation-invariant texture classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 4, pp. 472–481, 1986.

[92] A. Khotanzad and R. Kashyap, "Feature selection for texture recognition based on image synthesis," *IEEE Trans. Systems, Man and Cybernetics*, vol. 17, pp. 1087–1095, 1987.

[93] K. Konstantinides and D. Tretter, "A method for variable quantization in JPEG for improved text quality in compound documents," in *Proc. IEEE Intl. Conf. on Image Processing, ICIP '98*, pp. 565–568, Oct. 1998.

[94] J. Kovacevic and M. Vetterelli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for r^n," *IEEE Trans. on Information Theory*, vol. 38, no. 2, pp. 533–555, 1993.

[95] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 7, pp. 737–747, 1993.

[96] P. Kundu and B. B. Chaudhuri, "Fuzzy geometric feature-based texture classification," *Pattern Recognition Letters*, vol. 14, pp. 825–832, 1993.

[97] O. J. Kwon and R. Chellappa, "Segmentation based image compression," *Optical Engineering*, vol. 32, pp. 1581–1587, 1993.

[98] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1186–1190, 1993.

[99] A. Laine and J. Fan, "Frame representation for texture segmentation," *IEEE Trans. Image Processing*, vol. 5, no. 5, pp. 771–779, 1996.

[100] N. Lamei, K. D. Hutchinson, M. M. Crawford, and N. Khazenie, "Cloud-type discrimination via multispectral texture analysis," *Optical Engineering*, vol. 33, no. 4, pp. 1303–1313, 1994.

[101] K. L. Laws, "Rapid texture identification," in *Proc. SPIE*, vol. 238, pp. 376–380, 1980.

[102] F. Lebourgeois, Z. Bublinski, and H. Emptoz, "A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents," in *Proc. IEEE Intl. Conf. on Image Processing, ICIP '92*, pp. 272–276, 1992.

[103] T. S. Lee, "Image representation using 2-d Gabor filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[104] J. Li and R. M. Gray, "Context-based multiscale classification of document images using wavelet coefficient distributions," *IEEE Trans. Image Processing*, vol. 9, pp. 1604–1616, 2000.

[105] R. W. Lindsay, D. B. Percival, and D. A. Rothrock, "The discrete wavelet transform and the scale analysis of the surface properties of sea ice," *IEEE Trans. Geoscience and Remote Sensing*, vol. 34, no. 3, pp. 771–787, 1996.

[106] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America*, vol. A, no. 7, pp. 923–932, 1990.

[107] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[108] S. Mallat, "Zero-crossings of a wavelet transform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 1019–1033, 1993.

[109] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, 1992.

[110] R. Manduchi and J. Portilla, "Independent component analysis of textures," in *Proc. 7th IEEE Intl. Conf. on Computer Vision*, (Kerkyra, Greece), pp. 1054–1060, September 1999.

[111] T. Matsuyama, S. Miura, and M. Nagao, "Structural analysis of natural textures by Fourier transformation," *Computer Vision, Graphics, Image Processing*, vol. 21, pp. 347–362, 1983.

[112] A. Mecocci, P. Gamba, A. Marazzi, and M. Barni, "Texture segmentation in remote sensing images by means of packet wavelets and fuzzy clustering," in *Proc. European Symposium on Satellite and Remote Sensing II*, vol. SPIE 2584, pp. 142–157, September 1995.

[113] MINERVA, http://www.project-minerva.ex.ac.uk.

[114] MIT Vision and Modeling Group, http://www.media.mit.edu/vismod/.

[115] D. Muchoney and J. Williamson, "A Gaussian adaptive resonance theory neural network classification algorithm applied to supervised land cover mapping us-

ing multitemporal vegetation index data," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 1969–1977, 2001.

[116] V. Murino, C. Ottonello, and S. Pagnan, "Noisy texture classification: A higher order statistics approach," *Pattern Recognition*, vol. 34, no. 4, pp. 383–393, 1998.

[117] I. Ng, T. Tan, and J. Kittler, "On local linear transform and Gabor filter representation of texture," in *Proc. Intl. Conf. on Pattern Recognition, ICPR '92*, pp. 627–631, 1992.

[118] A. Niedermeier, E. Romaneesen, and S. Lehner, "Detection of coastline SAR images using wavelet methods," *IEEE Trans. Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2270–2281, 2000.

[119] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation-invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.

[120] S. K. Pal, J. Basak, and R. K. De, "Fuzzy feature evaluation index and connectionist realization," *Information Sciences*, vol. 105, pp. 173–188, 1998.

[121] S. K. Pal and B. Chakraborty, "Fuzzy set theoretic measures for automatic feature evaluation," *IEEE Trans. Systems, Man and Cybernetics*, vol. 16, pp. 754–760, 1986.

[122] S. K. Pal, R. K. De, and J. Basak, "Unsupervised feature evaluation : A neuro-fuzzy approach," *IEEE Trans. Neural Networks*, vol. 11, pp. 366–376, 2000.

[123] S. K. Pal, A. Ghosh, and M. K. Kundu, eds., *Soft Computing for Image Processing*. Heidelberg: Physica Verlag, 2000.

[124] S. K. Pal, A. Ghosh, and B. U. Shankar, "Segmentation with remotely sensed images with fuzzy thresholding, and quantitative evaluation," *Intl. Journal of Remote Sensing*, vol. 21, no. 11, pp. 2269–2300, 2000.

[125] S. K. Pal and S. Mitra, *Neuro-fuzzy Pattern Recognition: Methods in Soft Computing*. New York: John Wiley, 1999.

[126] O. Pichler, A. Teuner, and B. J. Hosticka, "A comparison of texture feature extraction using adaptive Gabor filtering, pyramidal and tree structured wavelet transforms," *Pattern Recognition*, vol. 29, no. 5, pp. 733–742, 1996.

[127] R. Porter and N. Canagarajah, "A robust automatic clustering scheme for image segmentation using wavelets," *IEEE Trans. Image Processing*, vol. 5, no. 4, pp. 662–665, 1996.

[128] M. Pregenzer, G. Pfurtscheller, and D. Flotzinger, "Automated feature selection with a distinctive sensitive learning vector quantizer," *Neurocomputing*, vol. 11, pp. 19–29, 1996.

[129] W. H. Press, S. A. Teukolsky, W. T. Vellerling, and B. P. Flannery, *Numerical recipes in C.* New Delhi: Cambridge University Press, 1995.

[130] T. Randen and J. H. Husφy, "Segmentation of text/image documents using texture approaches," in *Proc. NOBIM-konferansen-94*, pp. 60–67, June 1994.

[131] T. Randen, *Filter and Filter Bank Design for Image Texture Recognition.* PhD thesis, Norwegian Univ. of Science and Technology, Trondheim, Norway, 1997.

[132] T. Randen and J. H. Husφy, "Filtering for texture classification : A comparative study," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp. 291–310, April 1999.

[133] A. R. Rao and B. G. Schunck, "Computing oriented texture fields," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 2, pp. 157–185, 1991.

[134] T. R. Reed and J. M. H. du Buf, "A review of recent texture segmentation and feature extraction techniques," *CVGIP: Image Understanding*, vol. 57, no. 3, pp. 359–372, 1993.

[135] T. R. Reed and H. Wechsler, "Segmentation of textured images and gestalt organization using spatial/ spatial frequency representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1–12, Jan. 1990.

[136] J. A. Richards, *Remote Sensing Digital Image Analysis: An Introduction.* New York: Springer-Verlag, 1993.

[137] E. Rignot and R. Kowk, "Extraction of textural features in SAR images: Statistical model and sensitivity," in *Proc. IEEE Geoscience and Remote Sensing Symposium, IGARSS '90*, May 1990.

[138] O. Rioul, "A discrete time multiresolution theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 2591–2606, August 1993.

[139] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, pp. 14–38, October 1991.

[140] A. Rosenfeld and B. S. Lipkin, *Picture Processing and Psychopictorics.* Academic Press, 1970.

[141] A. Rosenfeld, eds., *Multiresolution Image Processing and Analysis.* New York: Springer-Verlag, 1984.

[142] G. H. Rosenfeld and K. Fitzpatrick-Lins, "Coefficient of agreement as a measure of thematic classification accuracy," *Photogrammetric Engineering and Remote Sensing*, vol. 52, pp. 223–227, 1986.

[143] M. A. Ruzon, V. C. Gulick, R. L. Morris, and T. L. Roush, "Autonomous scene analysis of digital images for MARS sample return and beyond," in *Proc. 30th Lunar and Planetary Science Conf.*, 1999.

[144] N. Saito and R. R. Coifman, "Local discriminant bases and their applications," *J. Mathematical Imaging and Vision*, vol. 5, no. 4, pp. 337–358, 1995.

[145] W. A. C. Schmidt and J. P. Davis, "Pattern recognition properties of various feature spaces for higher order neural networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 795–801, 1993.

[146] J. R. Schriever and R. G. Congalton, "Evaluating seasonal variability as an aid to cover-type mapping from landsat thematic mapper in the northeast," *Photogrammetric Engineering and Remote Sensing*, vol. 3, pp. 321–327, 1995.

[147] R. Setino and H. Liu, "Neural-network feature selector," *IEEE Trans. Neural Networks*, vol. 8, pp. 654–662, 1997.

[148] M. Simard, S. S. Saatchi, and G. D. Grandi, "The use of decision tree and multiscale texture for classification of JERS-1 SAR data over tropical forest," *IEEE Trans. Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2310–2321, 2000.

[149] S. N. Srihari, "Document image understanding," in *Proc. IEEE Computer Society Fall Joint Computer Conf*, pp. 87–96, November. 1986.

[150] P. Steffen, P. N. Heller, R. A. Gopinath, and C. S. Burrus, "Theory of regular M-band wavelet bases," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3497–3510, 1993.

[151] J. M. Steppe and K. W. Bauer, Jr., "Improved feature screening in feedforward neural networks," *Neurocomputing*, vol. 13, pp. 47–58, 1996.

[152] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley - Cambridge Press, 1996.

[153] S. Sural and P. K. Das, "A two step algorithm and its parallelisation for the generation of minimum containing rectangles for document image segmentation," in *Proc. Intl. Conf. on Document Analysis and Recognition, ICDAR '99*, pp. 173–176, 1999.

[154] C. L. Tan, B. Yuan, W. Huang, and Z. Zang, "Text/graphics separation using pyramid operations," in *Proc. Intl. Conf. on Document Analysis and Recognition, ICDAR '99*, pp. 169–172, 1999.

[155] Y. Y. Tang, Z. K. Chen, J. H. Y. Mak, and D. H. Xi, "Wavelet method for breast cancer detection in digital mammograms," in *Proc. 2nd Intl. Conf. on Multimodal Interface, ICMI '99*, pp. VI–121 – 126, January 1999.

[156] Y. Y. Tang, Y. Hou, J. Song, and X. Yang, "Mixture-state document segmentation using wavelet-domain hidden Markov tree models," in *Proc. 2nd. International Conference on Wavelet Analysis and its Application, ICWAA 2001*, (Hong Kong), pp. 230–236, December 2001.

[157] Y. Y. Tang, L. Yang, J. Liu, and H. Ma, *Wavelet Theory and its Application to Pattern Recognition*. Singapore: The World Scientific, 2000.

[158] Y. Tao, T. R. Ioerger, and Y. Y. Tang, "Extraction of rotation-invariant signature based on fractal geometry," in *Proc. Intl. Conf. on Image Processing, ICIP 2001*, pp. 1090–1093, October 2001.

[159] A. Teuner, O. Pichler, and B. J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters," *IEEE Trans. Image Processing*, vol. 4, pp. 863–870, June 1995.

[160] P. Thitimajshima, "Multiresolution fuzzy clustering for SAR image segmentation," in *Proc. Intl. Geoscience and Remote Sensing Symposium, IGARSS '99*, vol. 5, pp. 2507–2509, 1999.

[161] J. T. Tou and R. C. Gonzales, ***Pattern Recognition Principles.*** Adision-Wesley, Reading, 1974.

[162] D. C. Tseng, H. M. Tsai, and C. C. Lai, "Unsupervised texture segmentation for multispectral remote sensing images," in *Proc. Intl. Conf. on Pattern Recognition, ICPR'98*, pp. 1630–1632, 1998.

[163] M. R. Turner, "Texture discrimination by Gabor function," *Biological Cybernetics*, vol. 55, pp. 71–82, 1986.

[164] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1549–1560, 1995.

[165] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 717–728, 1989.

[166] G. Van de Wouwer, P. Vautrot, P. Scheunders, S. Livens, D. Van Dyck, and N. Bonnet, "Rotation-invariant texture segmentation using continuous wavelets," in *Proc. IEEE UK Symposium on Applications of Time-Frequency and Time-Scale Methods*, pp. 129–132, 1997.

[167] M. Vetterelli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, pp. 2207–2232, 1992.

[168] F. M. Wahl, K. Y. Wong, and R. G. Kasey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 20, pp. 375–390, 1982.

[169] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision, Graphics, Image Processing*, vol. 47, pp. 327–352, 1989.

[170] T. P. Weldon and W. E. Higgins, "Integrated approach to texture segmentation using multiple Gabor filters," in *Proc. Intl. Conf. on Image Processing, ICIP '96*, pp. 955–958, September 1996.

[171] W. R. Wu and S. C. Wei, "Rotation and gray-scale transform invariant texture classification using spiral resampling, subband decomposition and hidden Markov model," *IEEE Trans. Image Processing*, vol. 5, no. 10, pp. 1423–1434, 1996.

[172] L. Ying, A. Hertzmann, H. Biermann, and D. Zorin "Texture and shape synthesis on surfaces," in *Proc. 12th Eurographics Workshop on Rendering*, (London, UK), June 2001.

List of Publications of the Author

1. M. Acharyya and M. K. Kundu, "Robust texture classification using wavelet frames," *Image Processing and Communication: An International Journal*, vol. 5, no. 2, pp. 19–37, 1999.

2. M. Acharyya and M. K. Kundu, "Classification of rotated textures using over-complete wavelet frames," *Journal of IETE for Research, India*, vol. 46, no. 5, pp. 363–370, 2000.

3. M. Acharyya and M. K. Kundu, "Efficient rotation-invariant feature extraction for texture segmentation via multiscale wavelet frames," in *Proc. Indian Conf. on Computer Vision, Graphics and Image Processing, ICVGIP 2000*, (Bangalore, India), pp. 1–8, December 2000.

4. M. Acharyya and M. K. Kundu, "Two texture segmentation using M-band wavelets," in *Proc. 15th. Intl. Conf. on Pattern Recognition, ICPR 2000*, (Barcelona, Spain), pp. 405–408, September 2000.

5. M. Acharyya and M. K. Kundu, "An adaptive approach to unsupervised texture segmentation using M-band wavelet," *Signal Processing*, vol. 81, no. 7, pp. 1337–1356, 2001.

6. M. Acharyya and M. K. Kundu, "Adaptive basis selection for multitexture segmentation by M-band wavelet packet frame," in *Proc. Intl. Conf. on Image Processing, ICIP 2001*, (Thessaloniki, Greece), pp. 622–625, October 2001.

7. M. Acharyya and M. K. Kundu, "M-band wavelet analysis for document image segmentation," in *Proc. National Conf. on Intelligent Computing and VLSI, NCICV 2001*, (Kalyani, India), pp. 58–63, February 2001.

8. M. Acharyya and M. K. Kundu, "Multiscale segmentation of document images using M-band wavelets," in *Proc. 9th Intl. Conf. on Computer Analysis of Images and Patterns, CAIP 2001*, vol. LNCS 2124, (Warsaw, Poland), pp. 510–517, September 2001.

9. M. Acharyya and M. K. Kundu, "Wavelet-based texture segmentation of remotely sensed images," in *Proc. Intl. Conf. on Image Analysis and Processing, ICIAP 2001*, (Palermo, Italy), pp. 69–74, September 2001.

10. M. Acharyya and M. K. Kundu, "Extraction of noise tolerant gray-scale transform and rotation-invariant features for texture segmentation using wavelet frames," *IEEE Trans. Systems, Man and Cybernetics*, (Accepted).

11. M. Acharyya and M. K. Kundu, "Document image segmentation using wavelet scale-space features," *IEEE Trans. Circuits and Systems for Video Technology*, (Accepted).

12. M. Acharyya, R. K. De, and M. K. Kundu, "Extraction of features using M-band wavelet packet frame and their neuro-fuzzy feature evaluation for multi-texture segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, (Revised version submitted).

13. M. Acharyya, R. K. De, and M. K. Kundu, "Segmentation of remotely sensed images using wavelet features and their evaluation in soft computing framework," *IEEE Trans. Geoscience and Remote Sensing*, (Communicated).