Wavelet Transform for Texture Analysis With Application to Document Analysis

by

Andrew W. Busch, BEng(Hons), BIT

PhD Thesis

Submitted in Fulfilment

of the Requirements

for the Degree of

Doctor of Philosophy

at the

Queensland University of Technology

School of Electrical & Electronic Systems Engineering

August 2004

Substitute the "official" thesis signature page here.

Keywords

texture classification, texture analysis, wavelet transform, document analysis, multi-resolution, segmentation, script recognition, Gaussian mixture model, quantisation

Abstract

Texture analysis is an important problem in machine vision, with applications in many fields including medical imaging, remote sensing (SAR), automated flaw detection in various products, and document analysis to name but a few. Over the last four decades many techniques for the analysis of textured images have been proposed in the literature for the purposes of classification, segmentation, synthesis and compression. Such approaches include analysis the properties of individual texture elements, using statistical features obtained from the grey-level values of the image itself, random field models, and multichannel filtering. The wavelet transform, a unified framework for the multiresolution decomposition of signals, falls into this final category, and allows a texture to be examined in a number of resolutions whilst maintaining spatial resolution.

This thesis explores the use of the wavelet transform to the specific task of texture classification, and proposes a number of improvements to existing techniques, both in the area of feature extraction and classifier design. By applying a nonlinear transform to the wavelet coefficients, a better characterisation can be obtained for many natural textures, leading to increased classification performance when using first and second order statistics of these coefficients as features. In the area of classifier design, a combination of an optimal discriminate function and a non-parametric Gaussian mixture model classifier is shown to experimentally outperform other classifier configurations.

By modelling the relationships between neighbouring bands of the wavelet trans-

form, more information regarding a texture can be obtained. Using such a representation, an efficient algorithm for the searching and retrieval of textured images from a database is proposed, as well as a novel set of features for texture classification. These features are experimentally shown to outperform features proposed in the literature, as well as provide increased robustness to small changes in scale.

Determining the script and language of a printed document is an important task in the field of document processing. In the final part of this thesis, the use of texture analysis techniques to accomplish these tasks is investigated. Using maximum a posterior (MAP) adaptation, prior information regarding the nature of script images can be used to increase the accuracy of these methods. Novel techniques for estimating the skew of such documents, normalising text block prior to extraction of texture features and accurately classifying multiple fonts are also presented.

Contents

List of Tables

List of Figures

Acronyms & Abbreviations

ANN Artificial neural network

CWT Continuous wavelet transform

DCT Discrete cosine transform

DET Detection error trade-off

DFT Discrete Fourier transform

DWT Discrete wavelet transform

EER Equal error rate

EM Expectation maximisation

EMD Earth mover's distance

FFT Fast Fourier transform

FWT Fast wavelet transform

GLCM Grey level co-occurrence matrix

GMM Gaussian mixture model

GRF Gibbs random field

HMM Hidden Markov model

LCPDF Local conditional probability density function

LMS Least mean square

LPC Linear predictive coefficient

MAP Maximum *a posterior*

MLP Multi-layer perceptron

MRF Markov random field

MSE Mean-squared error

OCR Optical character recognition

SAR Synthetic aperture radar

STFT Short time Fourier transform

WT Wavelet transform

Certification of Thesis

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher educational institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed: _____

Date: _____

Acknowledgments

Writing this thesis has undoubtedly been the most difficult, time consuming, and rewarding experience of my life to date, and would not have been possible without the help of many people. First and foremost, my most sincere thanks must go to my supervisor, Associate Professor Wageeh Boles, for his guidance, wisdom, and most of all his infinite patience and understanding, without which this thesis may not have been completed at all. Although I am sure there are many better PhD students than myself, I can think of no better supervisor. Thank you for convincing me to undertake this course, for your friendship and support throughout, and for believing in me even when I did not.

Many thanks must also go to Professor Sridha Sridharan, my associate supervisor and director of our research group. His support in providing funding and equipment, and finding suitable external projects for me to work on was invaluable, and made my progress through this project much smoother than it otherwise would have been. The support of the Australian government, in the form of a postgraduate scholarship, as well as funding by the Australian Defence Science and Technology Organisation (DSTO) is also gratefully acknowledged.

Throughout my time at the Queensland University of Technology I have had the privilege of working alongside some fantastic people, namely my fellow students in the Speech, Audio, Image and Video Research laboratory. Not only has my research been immensely aided by their presence, I have also formed friendships which mean more to me than any degree ever could. In particular, my thanks go

to Jason Pelecanos for his invaluable help in many areas both work-related and otherwise, to Simon Lucey for opening my eyes to the wonderful world of pattern recognition theory, to Michael Mason for his help in all things technical, and to John Dines for his constant stream of humourous emails and outstanding music collection. Scott Lowther has helped with a significant portion of the document processing work presented in this thesis, and my sincere thanks must go to him for this. To everybody else I have shared the lab with over the last four years, thank you for the card games, the witty and sometimes intelligent conversation, and for making it a great place to work.

Finally, my love and thanks go to my partner Michelle, my parents, and the rest of our families and friends, who have been a constant source of support for me while completing this work, through both the good times and the bad. Although many of you may not understand, nor ever want to understand, much of what is written beyond this page, you should know that it could not have been done without you.

Andrew W. Busch

Queensland University of Technology

August 2004

Chapter 1

Introduction

1.1 Motivation and Overview

Texture analysis is an important area in the field of image processing, with applications in computer vision, graphics, medical imaging, remote sensing, document image analysis and quality control to name but a few. Texture analysis techniques have been used for the classification, segmentation, and synthesis of these types of images. Over the last four decades, a large number of techniques for achieving this goals have been proposed, with approaches ranging from studying the properties of the primitive texture elements, or textons, using statistics of the individual pixel values, modelling the images with random field models, filtering of the images with a variety of kernels, to the most recent techniques which analyse the textures over multiple scales. Much of the work in the field of texture analysis has been inspired in part by studies of the human visual system, which have revealed the existence of cells in the visual cortex which respond only to stimuli of certain spatial frequencies and orientations.

1.1.1 Evaluation of Texture Analysis Techniques

Measuring the relative performance of different texture analysis techniques is a problem that has received significant attention in the literature. Because of the almost infinite range of possible textures, as well as the difficulty in defining the nature of texture itself, it is almost impossible to provide a global measurement of the performance of a particular algorithm. Rather, a relative measure can be obtained only within the scope of a particular set of images or a specific application. To facilitate such comparisons, a number of texture databases have been compiled which are widely used in the research community, including images from the Brodatz album, and the VisTex and MeasTex databases. Each of these collections contains a variety of natural and artificially created textured images covering a wide range of applications and environments. The performance of different feature sets, however, can still differ in terms of both absolute and relative performance when applied to each of these sets of images. In this thesis the task of evaluating the performance of texture analysis techniques is investigated in more detail, with the performance of a number of approaches studied under a variety of conditions.

1.1.2 The Wavelet Transform for Texture Analysis

The wavelet transform has recently emerged as a formal, unified framework for the multiresolution decomposition of signals and images, finding applications in an extremely large range of fields including mathematics and many areas of signal and image processing. Texture analysis is one such field, with a large number of techniques utilising the wavelet transform presented in the literature. These algorithms have been shown to perform very well, with excellent segmentation and classification results shown over a wide range of images.

1.2 Aims and Objectives

The general aims of this thesis are:

(i) To provide a thorough review of the theory of the wavelet transform and the history of texture analysis.

(ii) To investigate techniques for adapting and improving these techniques such that better classification accuracy can be obtained, with specific focus on utilising the coefficients of the wavelet transform for this purpose.

(iii) To investigate the relationships between the bands of the wavelet decomposition of a textured image, and use such relationships to create an improved texture model.

(iv) To theoretically and experimentally explore the advantages and disadvantages of various classifier designs in the context of texture classification, and to improve on such designs where possible.

(v) To apply these methods, as well as existing techniques of texture analysis, to the problem of automated script recognition of document images.

More specifically, the research objectives are:

(i) To undertake a thorough investigation of commonly used texture analysis techniques, by creating a structured testing environment which evaluates the performance of each technique for varying image types, in varying amounts of noise, and when altered slightly in scale or orientation.

(ii) To develop a better quantisation system for wavelet coefficients in order to improve texture classification performance.

(iii) To investigate the effect of feature reduction and classifier design on overall classification accuracy.

(iv) To study the relationships between scales of the wavelet transform coefficients, and use such information to develop a new method of texture characterisation.

(v) To apply a number of texture analysis techniques, including those developed in previous work, to the specific problem of automated script detection, and evaluate the performance of such methods.

1.3 Scope

The problem of texture analysis has been studied for many decades, over which time a number of approaches to the problem have been developed. This thesis will provide a general description of many of these techniques, however the most focus will be given to methods which employ the use of multi-resolution analysis. Such techniques have recently been shown to provide excellent results.

1.4 Outline of Thesis

The remainder of this thesis is organised as follows:

Chapter 2 presents a thorough theoretical and practical overview of the wavelet transform, its development, and applications. Topics covered include the development of the wavelet transform from the existing Fourier transform and short-time Fourier transform, the continuous and discrete time wavelet transforms, wavelet frames, the development of the fast wavelet transform, and a number of variations to the transform such as the wavelet packet decomposition and the two dimensional wavelet transform.

Chapter 3 outlines the major contributions to date in the field of texture analysis, summarising the work presented by numerous researchers over the last

three decades. A number of definitions of texture are presented, showing the difficulty of describing what constitutes texture in an image without a specific application as reference. The various tasks in the field of texture analysis, such as classification, segmentation and synthesis are explained, with examples of each supplied from the literature. A brief description of a number of common techniques used to perform such tasks is then presented, with examples showing the advantages and disadvantages of each as well as their use in applications to date.

Chapter 4 presents a study on the effect that quantisation of wavelet coefficients has on texture classification, and proposes a new quantisation technique based on the logarithm function which is shown can significantly improve classification performance. Using this technique, a total of five new sets of texture features are proposed and evaluated, with results showing a reduction in classification error rates of up to 50% over three independent texture databases.

Chapter 5 presents a summary of feature reduction techniques and general classifier theory and design, with particular focus on their use in the field of texture classification. The advantages and disadvantages of a number of different forms of classifiers are investigated, and their use in the field of texture classification reviewed. Using this information, a classifier design for texture analysis is proposed using a combination of linear discriminate analysis (LDA) and a Gaussian mixture model (GMM) classifier. This combination is experimentally found to outperform other classifiers when applied to a standard texture classification task.

Chapter 6 presents a novel algorithm for texture characterisation and classification which utilises scale relationships between bands of a wavelet decomposition of textured images. Using this representation and the recently proposed earth mover's distance, an efficient system for the searching and retrieval of textured images from a database is developed. Features extracted from the scale co-occurrence representation are also used for classification tasks,

showing reduced error rates when compared to other techniques proposed in the literature, and more robustness to small changes in scale.

Chapter 7 presents the results of an investigation into the use of texture classification techniques when applied to the problem of automated script identification in document images. A brief review of the field of document analysis is presented, showing the need for this task as a precursor to optical character recognition (OCR). A number of common texture features, as well as those developed in previous chapters, are tested for this purpose, and the results presented. In order to reduce the amount of training data required for this task, MAP adaptation is used to take advantage of prior knowledge regarding the appearance of printed text. Finally, the problem of multi-font recognition is addressed by the use of a clustered LDA algorithm prior to classification.

Chapter 8 summarises the contributions of this thesis and presents a number of avenues for possible future research.

1.5 Original Contributions

This thesis presents a number of original contributions to the fields of texture classification and document analysis. These are summarised as:

(i) A new quantisation scheme for wavelet coefficients used in texture classification is proposed, which has been experimentally shown to improve the performance of existing texture features. Using this scheme, a total of five new texture feature sets are developed which show significant improvement in classification accuracy when compared to similar existing features.

(ii) A feature reduction and classification design is developed for use in texture classification tasks, using a combination of principal component analysis,

linear discriminate analysis and a Gaussian mixture model classifier. Experimental evidence shows that such a combination can be used to provide a stable and robust model of a textured image, with improved classification accuracy when compared with classifiers currently used in the literature.

(iii) A novel technique for representing texture using relationships between adjacent scales of a wavelet decomposition is developed, extending existing work which extracts features from each band individually. This new representation is called the *scale co-occurrence representation*.

(iv) Using the proposed scale co-occurrence representation of texture, the earth mover's distance metric is used to develop a similarity measure between two textured images. Using this measure, a novel method for retrieving textures from a database is proposed which is experimentally shown to give excellent results. By representing the textures in signature form rather than with full matrices, and by performing a tree structured search based on the levels of the wavelet decomposition, significant computational savings can be realised.

(v) A novel set of texture features is extracted from the scale co-occurrence representation for use in segmentation and classification tasks. Experimental results show that such features outperform comparable features obtained independently from each band. These features are also shown to be robust to the presence of small changes in scale, outperforming all other tested texture features in environments where such changes exist. This property make the scale co-occurrence features an attractive choice for many applications.

(vi) Experimental evidence is shown which suggests that the scale co-occurrence and spatial co-occurrence feature sets are complementary, with significant differences in the quality of the models created for different texture types. Taking advantage of this observation, a number of methods of combining the two sets to further improve classification accuracy are investigated. By using the sum method of classifier combination, it is shown that the overall

classification error rate can be reduced.

(vii) A novel method for determining the skew angle of a document image is presented in chapter 7, with results on a large document database showing improved performance when compared to existing techniques.

(viii) Prior to extracting texture features from script images, normalisation must be carried out to ensure robustness. A technique for performing such normalisation is developed in chapter 7 which accomplishes this task independently of the script or language of the input image.

(ix) A novel approach for determining the script of a document image using textural information is presented in chapter 7, with the texture features developed in chapters 4 and 6, as well as existing texture features, evaluated for this purpose. In order to ensure that the extracted features are meaningful and robust, an algorithm for prior normalisation of text regions is also presented.

1.6 Experimental Approaches for Accurate Evaluation

When evaluating the performance of the techniques developed in this thesis, special care has been given to the choice of training and testing data, ensuring that the reported results are both statistically significant and non-biased. Where possible, training and testing images have been extracted from separate images, such that at no time will any portion of an image from one set be present in the other. In situations where this is not possible, and only one image is available for both training and testing, half of the image is used for each purpose, again ensuring that no overlap occurs. When evaluating results from different feature sets, exactly the same training and testing images were used in all cases to ensure a meaningful comparison can be made.

1.7 Publications resulting from research

The following fully-refereed journal articles and conference papers have been produced as a result of the work in this thesis:

1.7.1 International Journal Publications

(i) A. Busch, W. W. Boles and S. Sridharan, "Scale co-occurrence features for texture classification," accepted for publication in *Journal of Electronic Imaging*, 2003.

(ii) A. Busch, W. W. Boles and S. Sridharan, "Texture for script identification", submitted to *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2004.

1.7.2 International Conference Publications

(i) A. Busch and W. W. Boles, "Multi-resolution pre-processing technique for rotation invariant texture classification," in *Proceedings of WOSPA*, 2000.

(ii) A. Busch, W. W. Boles, S. Sridharan and V. Chandran, "Texture analysis for script recognition," in *Proceedings of IVCNZ 2001*, pp. 289-293, 2001.

(iii) A. Busch and W. W. Boles, "Texture classification using multiple wavelet analysis," in *Proceedings of DICTA 2002*, pp 341-345, 2002.

(iv) A. Busch and W. W. Boles, "Texture classification using wavelet scale relationships," in *Proceedings of ICASSP 2002*, vol. 4, pp 3584-3587, 2002.

(v) A. Busch, W. W. Boles and S. Sridharan, "A multiresolution approach to document segmentation," in *Proceedings of WOSPA 2002*, pp. 43-46, 2002.

(vi) A. Busch, W. W. Boles, S. Sridharan and V. Chandran, "Detection of unknown forms from document images," in *Proceedings of Workshop on Digital Image Computing 2003*, pp. 141-144, 2003.

(vii) A. Busch, W. W. Boles and S. Sridharan, "Calculating the similarity of textures using wavelet scale relationships," in *Proceedings of ANZIIS*, pp. 507-512, 2003.

(viii) A. Busch, W. W. Boles and S. Sridharan, "Logarithmic quantisation of wavelet coefficients for improved texture classification performance," in *Proceedings of ICASSP*, pp. 569-572, 2004.

(ix) A. Busch, W. W. Boles and S. Sridharan, "Combining scale and spatial wavelet features for improved texture modelling", accepted for presentation at *International Conference on Computational Intelligence for Modelling Control and Automation*, 2004.

Chapter 2

The Wavelet Transform

2.1 Introduction

The wavelet transform represents a relatively new addition to the field of signal processing, which has developed from a rather obscure origin. The first use of a similar transform is generally accredited to mathematicians studying harmonic analysis, although computer scientists researching multiscale image processing used techniques of a similar nature . The application of these transforms to general signal processing was first proposed by a group of French researchers, including Meyer, Mallat and Daubechies [1, 2, 3].

The general concept of the wavelet transform is to analyse a signal as a superposition of wavelets across multiple scales. As such, wavelet theory is closely related to the Fourier transform, and comparisons between the two are easily made.

This chapter shall give a brief history of the development of the wavelet transform from other frequency domain techniques. Fourier theory is explained, including the Fourier Transform (FT) and the Short-time Fourier Transform (STFT), and its relationship to the wavelet transform outlined. The mathematical the-

ory behind the wavelet transform is then developed, including both the continuous wavelet transform (CWT) and discrete wavelet transform (DWT). The two-dimensional form of the transform as well as the dyadic wavelet transform are discussed, as is wavelet packet analysis. The wavelet transform has found applications in many fields, and some examples of such applications are outlined in the final section of this chapter.

2.2 Fourier Theory

The Fourier Transform is a cornerstone in the field of signal processing, allowing a signal to be transformed between the time (or spatial) and frequency domains. The transform exists for both continuous and discrete signals, and is extensible to any number of dimensions.

2.2.1 The Fourier Series and Fourier Transform

Any periodic waveform, $f(t)$, can be represented as the sum of an infinite number of sin and cos waves, together with a DC component (ref). Such a representation is known as the Fourier Series, and can be expressed as

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + \sum_{n=1}^{\infty} b_n \sin(n\omega t) \tag{2.1}$$

where t is an independent variable which often represents time, but can and does represent other quantities such as distance. ω represents the periodic frequency of the signal, also known as the fundamental frequency or first harmonic.

The parameters of the Fourier series, a_n and b_n, can be calculated from the original signal by the following equations:

$$a_0 = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} f(t)\ dt \tag{2.2}$$

$$a_n = \frac{2}{T_p} \int_{-T_p/2}^{T_p/2} f(t) \cos(n\omega t)\ dt \tag{2.3}$$

$$b_n = \frac{2}{T_p} \int_{-T_p/2}^{T_p/2} f(t) \sin(n\omega t)\ dt \tag{2.4}$$

It is worth noting that the first of these terms, a_0, is equal to the average of the signal over one complete period (T_p) of the signal, and is thus the DC component.

Taking advantage of complex exponential notation, (2.1) can be more compactly represented as

$$f(t) = \sum_{n=-\infty}^{\infty} d_n e^{-jn\omega t}\ dt \tag{2.5}$$

where

$$d_n = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} f(t) e^{-jn\omega t}\ dt \tag{2.6}$$

When using this representation, d_n is complex, with the real and imaginary components representing the cosine and sine coefficients respectively. Although the series now includes negative values of n, this is only of mathematical significance with the only real effect being that the values of d will be half those of the corresponding values of a and b.

The primary disadvantage of Fourier series analysis is that it is only useful for purely periodic functions. In order to generalise the transform to any type of signal, it is therefore necessary to increase the period T_p of the signal to infinity. As the period approaches this amount, the spacing between harmonic components approaches zero, meaning that the discrete frequency intervals of $n\omega$ are replaced by a single variable ω, and the frequency spectrum becomes continuous. In doing this, the Fourier series equations become

$$d(\omega) = \frac{d\omega}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{2.7}$$

It has become convention to normalise this equation by dividing by $d\omega/2\pi$ to obtain

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\ dt \tag{2.8}$$

$$= \langle e^{j\omega t}, f(t) \rangle \tag{2.9}$$

(2.9) is commonly known as the Fourier Transform or Fourier analysis, and represents the frequency spectrum of the signal. Generally, this spectrum will be complex in nature, and it is commonly split into its magnitude and phase components. The inverse Fourier transform is then given by

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} \, d\omega \tag{2.10}$$

which is also known as the Fourier synthesis formula.

2.2.2 Short-Time Fourier Transform

When analysing stationary, bandwidth-limited signals, the Fourier transform is an important tool. However, in practice, the Fourier transform has a number of disadvantages. Firstly, temporal information from the entire signal is required to calculate every part of the transform, meaning future behaviour of the signal must be known. For this reason, the Fourier transform is unsuited to many real-time tasks. Additionally, certain features in a signal, such as discontinuities, have a large effect on the entire spectrum, often overshadowing other more important features.

To overcome such deficiencies, the original transform has been modified to give the Short-Time Fourier Transform (STFT), which uses a fixed-size window to apply the Fourier transform to only a small section of the signal at once [4]. This can be represented by

$$\int_{-\infty}^{\infty} f(t) e^{-j\omega t} w(t-b) \, dt \tag{2.11}$$

The windowing function $w(t-b)$ is a localised function which is shifted over the temporal axis to compute the transform at several positions b. Thus, the transform has become time-dependant as well as frequency dependant.

The resolution in time and frequency of the STFT cannot be arbitrarily small, but must satisfy the condition known as the Heisenberg inequality (2.12) [5].

$$\sigma_t \sigma_\omega \geq \frac{1}{2} \tag{2.12}$$

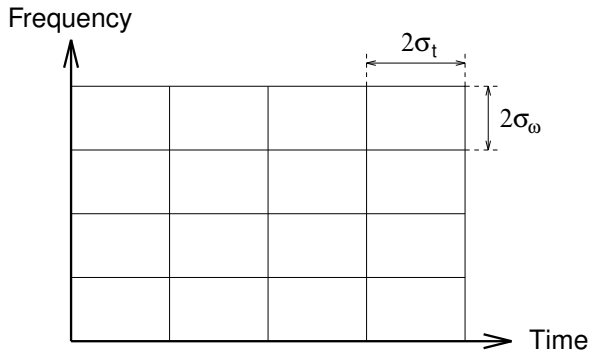


Figure 2.1: Time-frequency resolution of the Short-Time Fourier Transform

This inequality implies that one can only trade time resolution for frequency resolution, or vice versa. The optimal window for time-frequency localisation is achieved by using a Gaussian windowing function, as this will meet the bound with equality in (2.12). In this special case, the STFT is known as a Gabor transform[4].

Although the STFT is an improvement over the original Fourier Transform in that it achieves both time and frequency localisation, it still has a number of disadvantages when used as a tool for multi-resolution analysis. The primary drawback is that the same size window function is used for each frequency, meaning that the time and frequency resolutions, as shown in figure 2.1 are constant. To avoid this, it is necessary to develop a transform which varies both σ_t and σ_ω in the time-frequency plane.

2.3 Continuous Wavelet Transform

The wavelet transform (WT) has been developed to allow some temporal or spatial information in the Fourier domain [6]. The idea of the wavelet transform is to use a family of functions localised in both time and frequency. To accomplish this, the transform function, known as the mother wavelet, is modified by trans-

lations and dilations. The frequency variable used with the Fourier transform is therefore replaced by the dilation parameter.

In order to be classed as a wavelet, the analysing function $\psi(t)$, with Fourier transform $\Psi(\omega)$, must satisfy the following admissibility condition:

$$C(\Psi) = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|}\, d\omega < \infty \tag{2.13}$$

If ψ is absolute summable and Ψ is continuous, then this condition implies that $\Psi(0) = 0$, and is small in the neighbourhood of $\omega = 0$.

Essentially, this admissibility criteria ensures that the function, called the mother wavelet, is a bandpass filter. From this function, a family of functions which are the actual wavelets can be derived according to the following equation

$$\psi_{a,b} = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right) \tag{2.14}$$

The parameter b, a real number, is a translation parameter, while a, a real non-zero number, is used for dilation. For $a \ll 1$, the resulting wavelets are shrunken versions of the mother wavelet ψ, with the frequency spectrum concentrated mostly in higher frequencies. Conversely, with $a \gg 1$, the wavelets are spread, with most of the spectrum being lower in frequency. The constant $|a|^{-1/2}$ is to ensure that the energy of each wavelet $\psi_{a,b}$ is the same as the energy of the mother wavelet ψ. Since the support of $\psi_{a,b}$ varies proportionally to the dilation parameter a, the corresponding frequency radius σ_ω and temporal radius σ_t also change in proportion and inverse proportion respectively. While still satisfying (2.12), time resolution becomes arbitrarily good at high frequencies, whilst frequency resolution becomes arbitrarily at low resolutions, as can be seen in 2.2.

Given this family of wavelets, the CWT of a function $f \in L^2(\mathbb{R})$ is defined as the inner product:

$$\begin{aligned} Wf(a,b) &= \langle f, \psi_{a,b}\rangle \\ &= |a|^{-1/2}\int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t-b}{a}\right)\, dt \end{aligned} \tag{2.15}$$

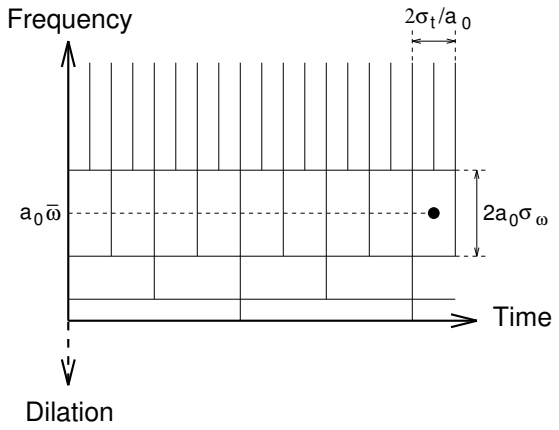


Figure 2.2: Time-frequency resolution of the Wavelet Transform

where ψ^* is the complex conjugate of ψ. Alternatively, the CWT can be expressed as the output of a filter matched to $\psi_{a,b}$ at time b

$$Wf(a,b) = f * \tilde{\psi}_{a,b} \tag{2.16}$$

where $*$ denotes linear convolution and $\tilde{\psi} = \psi^*(-t)$. As $\tilde{\psi}$ also satisfies the admissibility criterion of (2.13), it is also a mother wavelet.

Also because of this criterion, it can be shown that the wavelet transform is reversible by using the *resolution of identity* formula [7].

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Wf(a,b)\psi_{a,b}(t) \frac{da\ db}{a^2} \tag{2.17}$$

However, this reconstruction is only valid in terms of the signal's energy. Since $\int \psi(t)dt = 0$, this reconstruction will always have a zero mean. Other variations on the reconstruction (2.17) are also possible. If f is a real function and ψ is an analytical signal, it can be shown that [7]

$$f(t) = \frac{2}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Re\left\{Wf(a,b)\psi_{a,b}(t)\right\} \frac{da\ db}{a^2} \tag{2.18}$$

If both f and ψ are analytical, then $Wf(a,b) = 0$ for $a < 0$, allowing (2.17) to be simplified to

$$f(t) = \frac{1}{C_\psi} \int_{0}^{\infty} \int_{-\infty}^{\infty} Wf(a,b)\psi_{a,b}(t) \frac{da\ db}{a^2} \tag{2.19}$$

An extremely important reconstruction technique consists of using a separate function $\breve{\psi}$, such that [7]

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)||\breve{\Psi}(\omega)|}{|\omega|}\, d\omega < \infty \tag{2.20}$$

If this condition is satisfied, the reconstruction is then given by

$$f(t) = \frac{1}{C_{\psi,\breve{\psi}}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Wf(a,b)\breve{\psi}(t)\frac{da\ db}{a^2} \tag{2.21}$$

where

$$C_{\psi,\breve{\psi}} = \int_{-\infty}^{\infty} \frac{\Psi(\omega)\breve{\Psi}^*(\omega)}{|\omega|}\, d\omega \neq 0 \tag{2.22}$$

Given the condition given in (2.20) holds, ψ and $\breve{\psi}$ may have very different properties, and indeed need not both be admissible.

2.4 Wavelet Series

The general CWT maps a 1-D signal into 2-D (dilation and location) space, leading to much redundancy. In order to reduce this redundancy, it is necessary to sample the dilation and translation parameters a and b. An intuitive method for such sampling is $a = a_0^j$ and $b = ka_0^j b_0$, where $j, k \in \mathbb{Z}$ and $a_0 > 1$, $b_0 > 0$. This form of sampling leads to the sequence of wavelets

$$\psi_{j,k}(t) = a_0^{-j/2}\psi\left(\frac{1}{a_0^j} - kb_0\right) \tag{2.23}$$

and wavelet coefficients

$$Wf(j,k) = \langle f, \psi_{j,k}\rangle = \int_{-\infty}^{\infty} f(t)\psi_{j,k}^*(t)\ dt \tag{2.24}$$

To ensure adequate reconstruction from such coefficients, it is necessary to find a_0, b_0 and $\psi(t)$ such that

$$f(t) \approx \frac{1}{C_\psi}\sum_j\sum_k Wf(j,k)\psi_{j,k}(t) \tag{2.25}$$

It is clear from (2.23) and (2.24) that if a_0 is in the vicinity of 1, and b_0 is small, the wavelet series will form an overcomplete representation, and reconstruction is possible with few restrictions on $\psi(t)$. Conversely, if a_0 and b_0 are larger, an orthonormal basis will only be possible for a limited set of mother wavelets.

2.4.1 Wavelet Frames

A family of functions $\{x_k\}$ in a Hilbert space H is called a *frame* if there exists two constants $A > 0$, $B < \infty$, such that for all y in H [8]

$$A\|y\|^2 \leq \sum_k |\langle x_k, y\rangle|^2 \leq B\|y\|^2 \tag{2.26}$$

where A, B are referred to as the *frame bounds*. In the case of $A = B$, this is known as a *tight frame*, with (2.26) becoming

$$\sum_k |\langle x_k, y\rangle|^2 = A\|y\|^2 \tag{2.27}$$

and signal reconstruction possible as follows:

$$y = \frac{1}{A}\sum_k \langle x_k, y\rangle x_k \tag{2.28}$$

Although this equation is similar the expansion formula for an orthonormal basis, in general a frame is not an orthonormal basis, with vectors with the frame possibly linearly dependent. Only if $A = B = 1$, with $\|x_k\| = 1$ for all k, does x_k constitute an orthonormal basis.

Applying this theory to the wavelet series developed in 2.4, a *wavelet frame* is any set of wavelets $\psi_{j,k}$ such that for all functions f

$$A\|f\|^2 \leq \sum_j \sum_k |\langle f, \psi_{j,k}\rangle|^2 \leq B\|f\|^2 \tag{2.29}$$

Given $\psi(t)$, a_0 and b_0, it is possible to calculate the frame bounds A and B using Daubechies' formulae [1]

$$A = \frac{2\pi}{b_0}\left\{\inf_{1\leq|\omega|\leq a_0}\sum_{j=-\infty}^{\infty}|\Psi(a_0^j\omega)|^2 - \sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}\left[\beta\left(\frac{2\pi}{b_0}k\right)\beta\left(-\frac{2\pi}{b_0}k\right)\right]^{1/2}\right\} \tag{2.30}$$

$$B = \frac{2\pi}{b_0}\left\{\sup_{1\leq|\omega|\leq a_0}\sum_{j=-\infty}^{\infty}|\Psi(a_0^j\omega)|^2 - \sum_{\substack{k=-\infty\\k\neq 0}}^{\infty}\left[\beta\left(\frac{2\pi}{b_0}k\right)\beta\left(-\frac{2\pi}{b_0}k\right)\right]^{1/2}\right\} \tag{2.31}$$

where

$$\beta(s) = \sup_{\omega} \sum_{j} |\Psi(a_0^j \omega)||\Psi(a_0^j \omega + s)| \tag{2.32}$$

must decay at least as fast as $(1 + |s|)^{-(1+\epsilon)}$ with $\epsilon > 0$. The accuracy of signal reconstruction is also governed by these frame bounds, with

$$\begin{aligned} f(t) &\approx \frac{2}{A+B} \sum_j \sum_k Wf(j,k)\psi_{j,k}(t) \\ &= \frac{2}{A+B} \sum_j \sum_k Wf(j,k)\psi_{j,k}(t) + Rf(t) \end{aligned} \tag{2.33}$$

where Rf is the remainder term, bounded by

$$\int_{-\infty}^{\infty} |Rf(t)|^2 \, dt \leq \frac{B-A}{B+A} \int_{-\infty}^{\infty} |f(t)|^2 \, dt \tag{2.34}$$

If the wavelet frame is tight ($A = B$), the wavelet family functions as an orthonormal basis, although they may not be linearly independent in the general case. It can be seen from (2.34) that the remainder term becomes zero in this case. Furthermore, if the frame constant for such a tight wavelet frame becomes unity, the reconstruction formula exactly matches that of function with respect to an orthonormal basis,

$$f(t) = \sum_j \sum_k Wf(j,k)\psi_{j,k}(t) \tag{2.35}$$

2.4.2 Orthonormal Wavelet Bases

When using frames, much flexibility in the choice of sampling patterns a_0, b_0 is offered, with somewhat more restriction on the choice of mother wavelet $\psi(t)$. Often, however, this leads to significant redundancy in the wavelet representation. Although this redundancy leads to further robustness, and gives a closer approximation to the CWT, for many applications such as audio and image compression, little or no redundancy is desirable.

To achieve this, an orthonormal set of wavelets $\psi_{j,k}$ can be created by careful choice of both $\psi(t)$ and (a_0, b_0). The first and simplest such set was developed

by Haar [9], almost a century before the wavelet transform was first formally described. This basis is defined as

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.36}$$

and $a_0 = 2$, $b_0 = 1$. This leads to a wavelet family

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \tag{2.37}$$

which constitutes an orthonormal basis in $L^2(\mathbb{R})$. There are many other examples of orthonormal wavelet bases, some of which can be found in [7].

2.5 Dyadic Wavelet Transform

Due to the sampling of both the dilation and translation parameters, the wavelet frames discussed in 2.4.1 are in general non-invariant under translations. That is, if two signals are shifted versions of one another, the resulting wavelet frame representations will generally be different. Given a finite number of dilations $j_1 \leq j \leq j_0$, only for translations of $b_0 a_0^{j_0}$ (or multiples thereof) will the coefficients be necessarily identical. As this value is significantly larger than the sampling time, this may be extremely undesirable in applications where a stable representation is required, such as patter recognition tasks. In order to overcome this problem, Mallat and Zhong have proposed a transform whereby only the dilation parameter a is discretised along the dyadic sequence $(2^j)_{j\in\mathbb{Z}}$ known as the *dyadic wavelet transform*[10].

For a mother wavelet $\psi(t)$, the family of wavelets is then given by

$$\psi_j(t) = \frac{1}{2^j}\psi\left(\frac{t}{2^j}\right) \tag{2.38}$$

and the WT at a given level j and position t by

$$W_j f(t) = f * \psi_j(t) \tag{2.39}$$

or

$$W_j F(\omega) = F(\omega)\Psi(2^j\omega) \tag{2.40}$$

where W_jF,F and Ψ are the FT of W_jf, f and ψ respectively. As can be seen by (2.38), low values of j correspond to higher resolution levels (frequencies), and high values of j to the lower resolution levels.

If there exist two positive constants A and B such that

$$A \leq \sum_{j=-\infty}^{\infty} |\Psi(2^j\omega)|^2 \leq B \tag{2.41}$$

then the entire frequency spectrum is covered by dilations of $\Psi(\omega)$. It then follows that F, and therefore f, can be recovered from the dyadic wavelet transform. The reconstruction wavelet $\chi(t)$ can be any function whose Fourier transform $X(\omega)$ satisfies

$$\sum_{j=-\infty}^{\infty} \Psi(2^j\omega)X(2^j\omega) = 1 \tag{2.42}$$

Reconstruction is then performed by the summation

$$f(t) = \sum_{j=-\infty}^{\infty} W_j f * \chi_j(t) \tag{2.43}$$

Using Parseval's theorem it can be shown that the dyadic wavelet transform is stable and complete, with more stability obtained as $\frac{A}{B} \to 1$.

2.6 Discrete Wavelet Transform

The theory described so far in this chapter has dealt with continuous time signals, and their resulting transforms onto a wavelet basis set $\psi(t)$. In practice, however, it is more common that the signal of interest is sampled at discrete time intervals. Because of this, it is often more convenient to use the discrete wavelet transform (DWT), whereby both the signal and parameters (a, b) are discretised. An efficient implementation of this transform has been developed, which relies on a Quadrature Mirror Filter (QMF) pair and downsampling. This implementation is known as the fast wavelet transform (FWT).

2.6.1 Fast Wavelet Transform

The basis of the FWT is the multiresolution analysis presented in [2]. The multiresolution analysis of $L^2(\mathbb{R})$ is defined on a sequence of closed subspaces $V_j, j \in \mathbb{Z}$, such that the following properties are satisfied [3]:

$V_j \subset V_{j-1}$

$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j-1}$

$f(t) \in V_0 \Leftrightarrow f(t+1) \in V_0$

$\bigcup_{j=-\infty}^{\infty} V_j$ is dense in $L^2(\mathbb{R})$ and $\bigcap_{j=-\infty}^{\infty} V_j = \{0\}$

Within each subspace V_j, it is possible to find an orthogonal complement W_j in V_{j-1}, such that $W_j \oplus V_j = V_{j-1}$. In [2], Mallat showed that there exists two functions, the scaling, $\phi(t)$, and wavelet, $\psi(t)$, such that the sets $\{\phi(x-l)\}_{l\in\mathbb{Z}}$ and $\{\psi(x-l)\}_{l\in\mathbb{Z}}$ constitute an orthonormal basis of V_0 and W_0 respectively. It can also be shown that these two functions satisfy the difference equations

$$\phi(t) = 2\sum_k h(k)\phi(2t-k) \tag{2.44}$$

$$\psi(t) = 2\sum_k g(k)\phi(2t-k) \tag{2.45}$$

where

$$h(k) = \langle \phi_1(t), \phi(t-k) \rangle \tag{2.46}$$

$$g(k) = \langle \psi_1(t), \phi(t-k) \rangle \tag{2.47}$$

The projection of a signal $f(t)$ onto V_j results in an approximation signal $A_j f$ at resolution level j. This can be expressed as

$$A_j f = \langle f(t), \phi_j(t - 2^j n) \rangle = f * \tilde{\phi}_j(2^j n) \tag{2.48}$$

The difference between these approximation signals at each level is known as the detail signal, and denoted by $W_j f$, such that

$$W_j f = A_{j-1} f - A_j f \tag{2.49}$$

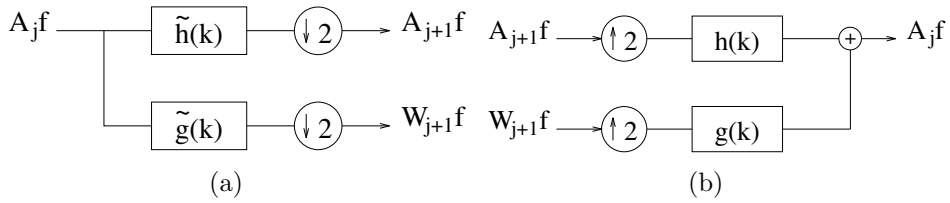


Figure 2.3: Block diagram showing the (a) decomposition and (b) reconstruction of a signal using the FWT algorithm

Since W_j and V_j form an orthogonal basis, this detail signal can also be obtained via an orthogonal projection of f onto W_j.

$$W_j f = \langle f(t), \psi(t - 2^j n) \rangle = f * \tilde{\psi}_j(2^j n) \tag{2.50}$$

From (2.48) and (2.50) it can be seen that the approximation and detail signals at each resolution level j can be calculted by resampling the projections of f onto the high- and low-pass filters ψ and ϕ. Furthermore, combining these equations with (2.44) and (2.45) yeilds

$$A_j f = \sum_k \tilde{h}(2n - k) \langle f(u), \phi_{j-1}(u - 2^{j-1} k) \rangle = A_{j-1} f * \tilde{h}(2n) \tag{2.51}$$

$$A_j f = \sum_k \tilde{g}(2n - k) \langle f(u), \phi_{j-1}(u - 2^{j-1} k) \rangle = A_{j-1} f * \tilde{g}(2n) \tag{2.52}$$

This is an important result, as it shows that the approximation and detail signals at any level of the wavelet decomposition can be obtained by convolution of the previous approximation signal (or original signal at level 0) with the filters $g(k)$ and $h(k)$, followed by downsampling by a factor of 2. Such an implementation allows lower resolution levels (those with higher values of j) to be more efficiently calculated since fewer samples are required to be filtered.

Reconstruction of a signal decomposed by the FWT is performed in a manner similar to that of the dyadic wavelet transform given by (2.43). A graphical representation of the FWT decomposition and reconstruction algorithms is shown in Figure 2.3.

Due to the decimation step of the FWT algorithm, each set of coefficients at each level has only half as many elements as those at the preceding level. This

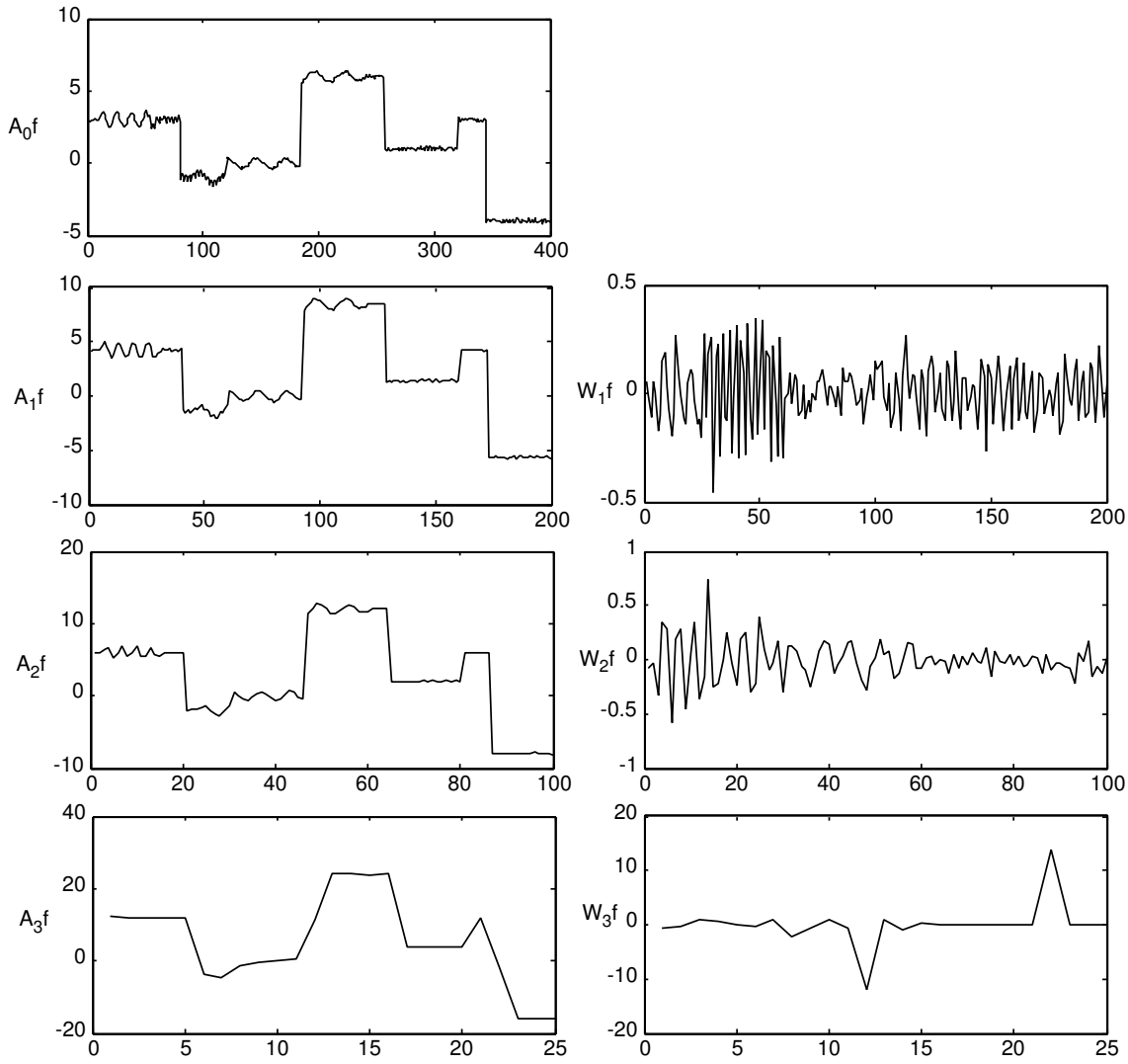


Figure 2.4: Example of FWT on a one dimensional signal showing a decomposition to four levels using the Haar wavelet. The detail and approximation signals are shown at each level.

property allows proportionally faster calculation of lower resolution coefficients. An example of the FWT on a one dimensional signal is shown in figure 2.4.

2.6.2 Undecimated Fast Wavelet Transform

One of the main disadvantages of the FWT is the non-translation invariance caused by the decimation at each subband. Because of this, a small change in the

spatial domain of the target signal can result in significant changes in the wavelet coefficients. Such invariance to small translations are highly undesirable in many applications such as pattern recognition and classification tasks. By removing the subsampling operation from the FWT algorithm, translation invariance can be obtained. In order to retain the frequency characteristics of the FWT, it is then necessary to construct new versions of $g(k)$ and $h(k)$ at each level. This is done by effectively upsampling the filters by a factor of two at each subsequent resolution level, creating a new set of $\{g(k)\}$ and $\{h(k)\}$ such that [11, 12]

$$h_0(k) = h(k) \tag{2.53}$$

and

$$h_j(k) = \begin{cases} h_{j-1}(k/2) & k = 2m, m \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \tag{2.54}$$

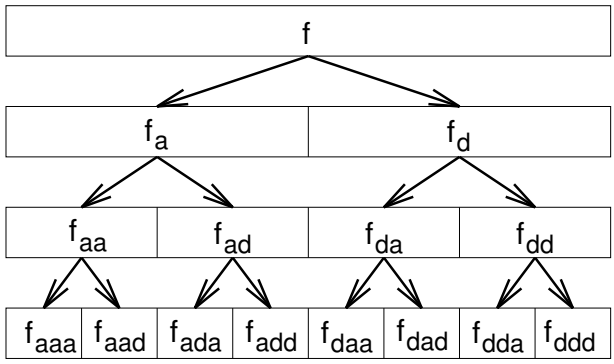
The set of functions $\{g(k)\}$ is constructed similarly.

Such a representation effectively calculates the FWT for all possible translations of the signal f, and can be seen as a discrete-time equivalent to dyadic wavelet transform described in section 2.5 which as previously discussed was translation invariant.

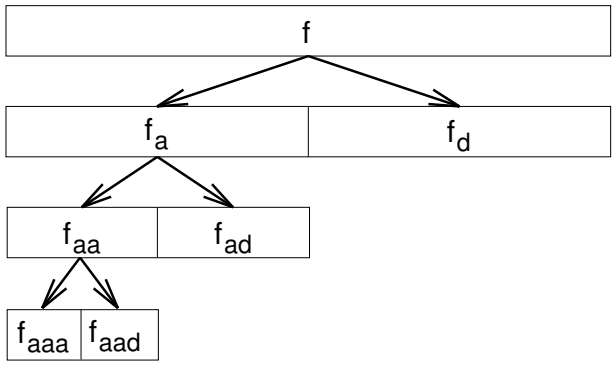
2.7 Wavelet Packet Transform

Using the DWT provides simultaneous localisation of both time and frequency, giving a vast improvement in signal understanding over the traditional DFT. As a consequence of this time-frequency localisation, the frequency divisions of the DWT occur in octave bands rather than equal steps. As such, time resolution at high frequencies is very good, while frequency resolution is much higher at low frequencies. In higher bands, therefore, the ability to accurately localise frequencies is limited.

The wavelet packet representation has been proposed to overcome this limitation,



(a)



(b)

Figure 2.5: Example of wavelet packet analysis showing (a) full wavelet packet analysis with decomposition of both approximation and detail signal at each level, and (b) FWT decomposition of approximation signal only

and enable arbitrarily high frequency resolution at any point of the signals spectrum. Rather than only the approximation signal being further decomposed at each successive level of the WT, both the detail and approximation signals are decomposed using the filters $g(k)$ and $h(k)$. This type of analysis can be represented as a binary tree structure within which one has the choice to continue or stop the analysis at each branch, giving rise to a number of possible basis choices.

Figure 2.5(a) shows a graphical representation of the wavelet packet decomposition of a signal f, compared to the normal FWT shown in figure 2.5(b).

2.8 Two Dimensional Wavelet Transform

The wavelet transform theory can be generalised to any dimensionality desired, however in this section only the two dimensional, discrete WT, with important applications in image processing, is considered. The two dimensional multiresolution analysis which describes the transform is derived directly from the one dimensional equivalent. It defines the space $L^2(\mathbb{R}^2)$ as a hierarchy of embedded subspaces V_j, such that none of the subspaces intersect, and for each function $f(\mathbf{t}) \in V_j,\ \mathbf{t} \in \mathbb{R}^2$, the following condition holds

$$f(\mathbf{t}) \in V_j \Leftrightarrow f(\mathbf{Dt}) \in V_{j-1} \tag{2.55}$$

where $\mathbf{D}$ is a 2×2 matrix with integer elements, and eigenvalues with absolute values greater than 1. This matrix performs the same function as a downsampling in the one dimensional FWT implementation, with the downsampling factor equal to its determinate, $|\mathbf{D}|$. The individual elements of the matrix indicate which samples of $f(t)$ are kept and which are discarded. If $\mathbf{n}$ and $\mathbf{u}$ are the points in the input and output images respectively, this can be represented as

$$\mathbf{n} = \mathbf{D} \cdot \mathbf{u} \tag{2.56}$$

or, in expanded form

$$\begin{bmatrix} n_1 \\ n_2 \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.57}$$

From this equation, it can be seen that the eigenvalues of $\mathbf{D}$ represent the dilation factors in each direction. If these values are all greater than 1, decimation in both directions is assured.

As in the one dimensional case, there exists in the multiresolution analysis a function $\phi(\mathbf{t})$ such that the family of functions $\phi(\mathbf{t} - \mathbf{k}),\ \mathbf{k} \in \mathbb{Z}^2$ constitutes an orthonormal basis for V_0. From this, it can be shown that

$$\phi_{j,\mathbf{k}}(\mathbf{t}) = |\mathbf{D}|^{-j/2}\phi(\mathbf{D}^{-j}\mathbf{t} - \mathbf{k}),\ \mathbf{k} \in \mathbb{Z}, \mathbf{t} \in \mathbb{R} \tag{2.58}$$

constitutes an orthonormal basis for V_j. In a similar manner to the one dimensional case of (2.44), $\phi(\mathbf{t})$ can also be expressed by the difference equation

$$\phi(\mathbf{t}) = |\mathbf{D}| \sum_{\mathbf{k}\in\mathbb{Z}^2} h(\mathbf{k})\phi(\mathbf{Dt} - \mathbf{k}) \tag{2.59}$$

In order to create a full orthogonal basis for the signal, a set of $|\mathbf{D}| - 1$ wavelet functions are required. Such functions can also be expressed in a similar fashion to the one dimensional version of (2.45), giving

$$\psi^l(\mathbf{t}) = |\mathbf{D}| \sum_{\mathbf{k}\in\mathbb{Z}^2} g^l(\mathbf{k})\phi(\mathbf{Dt} - \mathbf{k}), \quad l = 1, 2, \ldots, |\mathbf{D}| - 1 \tag{2.60}$$

The resulting set of wavelet functions thus constitutes an orthonormal basis of $L^2\mathbb{R}^2$.

The two dimensional WT is calculated by filtering the sampled signal $f(\mathbf{k})$ with the filters $h(\mathbf{k})$ and $g^l(\mathbf{k})$, $\quad l = 1, 2, \ldots, |\mathbf{D}| - 1$. Downsampling after each such filtering operation is performed by keeping only the elements indicated by $\mathbf{D}$. This entire procedure is then performed successively on the approximation signal to the required number of levels j.

The two dimensional wavelet transform has a number of forms, depending on the properties of the downsampling matrix $\mathbf{D}$. The two general cases are the *separable* and *non-separable* two dimensional wavelet transforms.

2.8.1 Separable Wavelets

If $\mathbf{D} = 2\mathbf{I}$, downsampling is performed by a factor of two in both the horizontal and vertical directions at each level of the transform. This special case is known as the separable wavelet transform, since the transform may be computed by filtering in each dimensional separately.

Using this type of transform, it is possible to define the scaling function $\phi(\mathbf{t})$ and the three wavelet functions $\psi^l(\mathbf{t}), \quad l = 1 - 3$ as combinations of the one

dimensional scaling and wavelet functions $\phi(t)$ and $\psi(t)$. Assuming image axes of (x, y), these functions can be expressed as

$$\begin{aligned} \phi(x,y) &= \phi(x)phi(y) \\ \psi^1(x,y) &= \phi(x)\psi(y) \\ \psi^2(x,y) &= \psi(x)\phi(y) \\ \psi^3(x,y) &= \psi(x)\psi(y) \end{aligned} \tag{2.61}$$

The scaled and shifted versions of the scaling function $\phi(x, y)$ thus form the approximation subspaces V_j in $L^2\mathbb{R}^2$, with the three wavelet functions together forming the detail subspace W_j. The approximation signal $A_j f$ at each level j can thus be obtained by an orthogonal projection of the input signal $f(x, y)$ onto the subspace V_j, with the three detail signals obtained from a similar projection onto each part of the detail subspace W_j. Thus,

$$A_j f = \left\langle f(x,y), \phi(x - 2^j m, y - 2^j n) \right\rangle \tag{2.62}$$

$$W_j^1 f = \left\langle f(x,y), \psi^1(x - 2^j m, y - 2^j n) \right\rangle \tag{2.63}$$

$$W_j^2 f = \left\langle f(x,y), \psi^2(x - 2^j m, y - 2^j n) \right\rangle \tag{2.64}$$

$$W_j^3 f = \left\langle f(x,y), \psi^3(x - 2^j m, y - 2^j n) \right\rangle \tag{2.65}$$

where $(m, n) \in \mathbb{Z}^2$.

Because each of the scaling and wavelet functions are separable, each of the 2D convolutions can be carried out as two separate 1D convolutions along each axis. Using the lowpass and highpass filters $h(t)$ and $g(t)$ obtained from (2.46) and (2.47) respectively, this filtering can be carried out using the efficient implementation illustrated in figure 2.6.

Because of nature of the wavelet filters, the separable wavelet transform can be viewed as a breakdown of a signal into a set of spatially oriented frequency channels, with W_j^1 and W_j^2 representing the vertical and horizontal high frequencies respectively at resolution level j. W_j^3 presents areas with high frequencies in both directions. Such an analysis is useful in detecting horizontal and vertical lines in images, as well as crossings and corners.

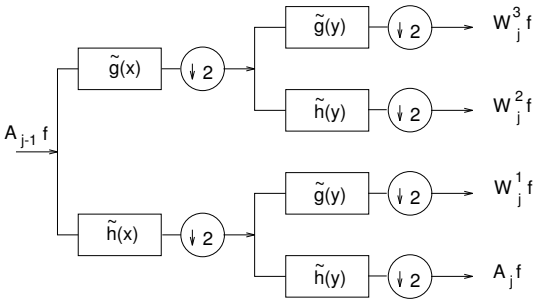


Figure 2.6: Graphical representation of single level of separable two dimensional wavelet transform.

Due to the subsampling involved in the transform, each of the approximation and detail signals contains only one quarter the pixels of those at the previous level. For this reason, a convenient representation involves placing the four sub-images together, with any further decomposition replacing the approximation image. An example of this representation on an image is shown in figure 2.7.

2.8.2 Non-Separable Wavelets

The separable wavelet transform represents a special case of the two dimensional wavelet transform, where the matrix $\mathbf{D}$ is equal to twice the identity matrix I. In many applications such a transform is useful, as it highlights important information along the principal axes of the image, as well as being computationally inexpensive to compute. In other applications, however, information from other aspects of the image is desired, and as such the use of the separable WT is inappropriate, and as such a non-separable approach is required. There are many choices of both scaling and wavelet function and sampling matrices for such cases, each with its own particular advantages and disadvantages.

On of the more common of the non-separable versions of the WT is the quincunx two dimensional wavelet transform, based on the quincunx pyramid. This transform has a downsampling rate of 2 meaning that only a single wavelet function

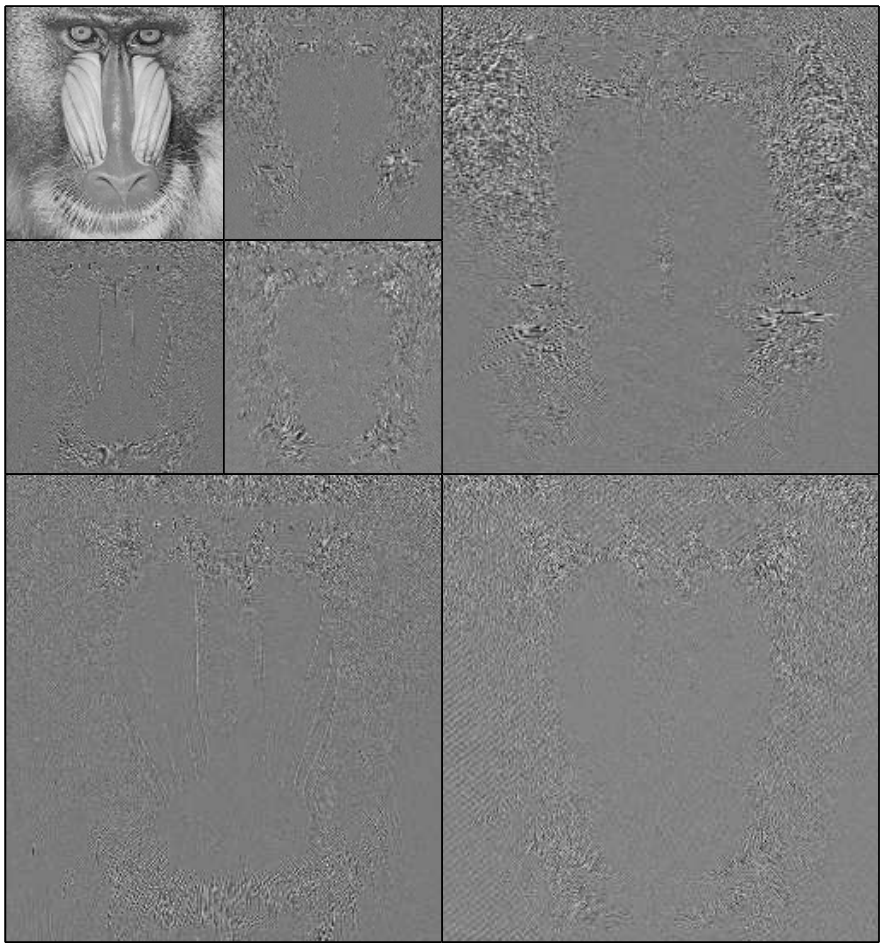


Figure 2.7: Example of the separable two dimensional wavelet transform with 2 levels of decomposition.

is used at each decomposition level. To obtain this downsampling rate, there are two choices of the matrix $\mathbf{D}$, namely

$$\mathbf{D_1} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{D_2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.66}$$

Downsampling for the quincunx WT is performed by retaining only those points (n_1, n_2) on the lattice, where $n_1 + n_2$ is even. A graphical representation of the quincunx downsampling procedure is shown in figure 2.8.

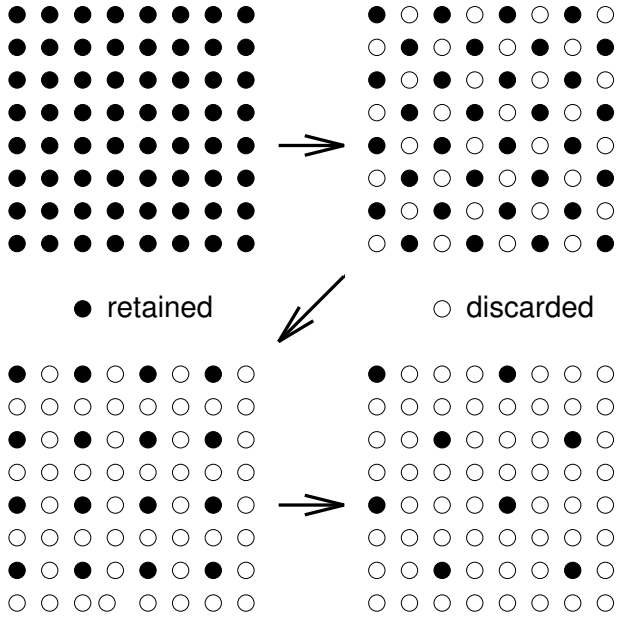


Figure 2.8: The Quincunx downsampling process.

2.9 Applications of Wavelets

Despite their relatively short history, wavelets have already been used extensively in many fields, with a multitude of practical applications employing wavelets already established. Such fields include mathematics, especially in numerical analysis problems, signal processing, image analysis and compression, and many others. The following sections show a small example of the many diverse fields in which wavelets have been successfully applied.

2.9.1 Signal and Image Analysis

Wavelets have been extensively used in signal and image analysis for purposes such as smoothing, noise reduction, edge and singularity detection, signal interpretation and object recognition. The excellent time-frequency localisation of the WT is a significant advantage in many of these areas over previous techniques. Additionally, the efficient algorithms for calculating the WT are often

significantly less computationally costly than other methods.

The recognition of objects, signals and images is one area of signal processing in which the wavelet transform has been found to be useful. By decomposing an object representation or signal into multiple scales and/or orientations, an accurate comparison between two such entities can be constructed. Jaggi *et. al.* use the WT combined with morphological techniques to describe objects in both low and high resolution, for the purposes of identifying military vehicles from infrared or radar images [13]. The basic geometry of the object can be determined from the low frequency information alone, while the high frequency details can be used for more specific recognition. The dyadic WT has also been used to describe objects in a representation which is invariant to affine transformations, a common problem in many object recognition applications [14, 15]. By using the WT of the affine arc length or enclosed area of an object contour, a dissimilarity function which compares the extrema of the resulting coefficients is used to compare two objects at a number of scales [15]. Wu and Bhanu use the magnitude, phase and frequency information of a Gabor wavelet representation of an object to match three dimensional objects, making use of a deformable Gabor grid [16].

Noise reduction is another application in the field of signal analysis in which the wavelet transform has been extensively used. By representing the signal in multiple resolutions, the wavelet transform allows noise which is present in only a few bands to be effectively isolated, regardless of its temporal location. A number of researchers have shown that applying a carefully calculated threshold to coefficients of the wavelet decomposition can effectively remove additive white Gaussian noise from a wide range of signals and applications [17, 18]. Ansleigh has proposed another technique for removing some types of non-Gaussian noise, such as spikes and unwanted harmonics, using B-spline wavelets [19]. The multi-resolution properties of the wavelet transform allow these types of noise to be isolated from the signal and more easily detected and removed. These and other wavelet-based noise removal techniques have been successfully applied to a wide

range of applications, from improving the quality of hearing aids [20] to removing phase noise from satellite SAR imagery [21].

2.9.2 Signal and Image Compression

Due to the compact representation of the wavelet coefficients, the wavelet transform is an ideal choice for use in signal compression algorithms. The theory behind such compression techniques is based on generic transform coding, where each block of the image is first transformed into another domain before compression is performed. A thorough review on this topic is provided in [22]. Wavelet compression is very similar in many respects to subband coding, in which the signal is recursively filtered and decimated in order to break it into uncorrelated, non-overlapping frequency bands [23]. In fact, early wavelet compression techniques differed from subband coding only in the choice of filters. While wavelet filters are generally designed to satisfy certain smoothness constraints, subband filters are constructed with complete separation of frequency bands in mind. Additionally, subband filters should also be near-lossless, in order to preserve as much of the original signal as possible. Once transformed, the various bands of the decomposed signal are quantised separately and encoded using traditional entropy coding techniques.

The first of the 'modern' wavelet image compression techniques was proposed by Shapiro, who exploited the multiresolution nature of wavelet decomposition in developing EZW coding [24]. This algorithm significantly improved the performance at low bit rates relative to the existing DCT compression algorithms used by the JPEG standard, as well as having many other nice properties. Since then, a number of similar techniques have been proposed by various authors [25, 26, 27, 28]. An evaluation of numerous wavelet-based image compression techniques is presented in [29]. The WT is the basis of JPEG2000, the standard which will likely supersede the current JPEG standard as the format of choice for lossy low-bitrate

image encoding.

Wavelet compression has also been used for data other than images. Hamid *et. al.* have developed such a compression algorithm for efficiently storing power disturbance data, such as transients due to ground faults and load switchings[30]. In the field of medical engineering, wavelets have also been effectively used to compress both mammogram and echocardiographic data in higher dimensional space [31, 32].

2.9.3 Numerical and Statistical Analysis

In the field of mathematics, wavelets have been extensively used to solve a wide range of numerical problems, as well as providing a basis for statistical analysis of distributions and other variables. By breaking the problem into multiple scales, more efficient methods of determining a solution are often possible using the WT.

One example of the use of the WT in solving complex numerical problems is presented in [33], where it is used to simplify and speed up the simulation of VLSI systems. Because the WT has sufficient resolution in the time domain, non-linear components can be adequately modelled, and it does not suffer from the instability which often results when working in the frequency domain. Using the WT also has advantages over comparable time-domain techniques as it allows for uniform approximation, and is generally more computationally efficient and more accurate. The WT has also been effectively used in a similar problem, that of finding the solution of transmission line equations [34]. Because of the time and frequency resolution provided by the WT, the differential equations which are used to model nonlinear components can be transformed into simple algebraic form. Results from this method show that the accuracy is equivalent to traditional forms of analysis, while providing a substantial increase in computational speed [34].

2.10 Chapter Summary

This chapter has presented an overview of the wavelet transform from both a theoretical and practical perspective. Fourier theory, the basis for frequency analysis of signals, was explained, along with the short-time Fourier transform, which incorporates temporal resolution into this transform. The Heisenberg inequality, which governs the trade-off between time and frequency resolution was discussed in this context.

The theory of the continuous wavelet transform was presented, along with a brief description of the wavelet series, wavelet frames and the dyadic wavelet transform. The multiresolution analysis leading to the fast wavelet transform was discussed, leading to a description of the FWT algorithm, an efficient algorithm for computing the wavelet transform on discrete signals. Variations on this process, including the undecimated fast wavelet transform and the wavelet packet transform were also described. The theory of wavelets was then expanded into two dimensions for application to image processing, with both the separable and non-separable forms of the two dimensional WT described.

A summary of the various applications of wavelets, was presented, showing the diversity of fields to which this theory has been successfully employed. Examples of these applications were given, including signal and image analysis, compression, de-noising, numerical analysis and many others.

Chapter 3

Texture Analysis Background

3.1 Introduction

Visual texture has been a topic of great interest in the field of image processing for some time. While traditional object recognition and segmentation techniques often rely on areas of constant intensity for object separation and description, this is rarely the case when dealing with most real-world images. Texture analysis attempts to solve this problem by supplying a description of areas which are not of uniform intensity, but rather those areas which are easily recognised by human observers as belonging to the same object or class. Typically, such areas contain a uniformly varying, periodic or structured appearance.

This chapter gives a brief summary of the history of texture analysis over the last three decades, from the pioneering work done by Julesz and Marr, to the latest multiresolution techniques. The concept of visual texture is first defined in this context, and the difficulties of such a description discussed.

Section 3.3 outlines the various sub-fields which fall under the category of texture analysis, such as texture classification, texture segmentation and texture com-

pression. The remaining sections of the chapter then outline a number of popular techniques which have been used over the years for these tasks, in roughly chronological order. Due to the large body of material on the subject present in the literature and research activity currently in progress, only a brief summary of each technique and their applications is presented here.

Section 3.5 gives examples of practical applications of texture analysis research, ranging from medical image analysis as an aid to diagnosis to interpretation of synthetic aperture rader (SAR) images, to the processing of document images.

3.2 Defining Texture

Throughout the last three decades, researchers have attempted to define exactly what is meant by the term *texture* in the field of machine vision. David Marr, who proposed the *primal sketch theory*, did much pioneering work in this field. The main thesis of his theory is that a *symbolic representation* of visual information is constructed early in the interpretation process, without the use of higher level knowledge, and that such early visual processing, including texture discrimination, operates at a symbolic level [35].

An overall definition of texture is almost impossible to formulate, as it is often dependent upon the particular application. A number of possible definitions which have been proposed in various contexts has been compiled by Coggins [36]. A few of these are

- "We may regard texture as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule." [37]
- "A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying,

or approximately periodic." [38]

- "The image texture we consider is nonfigurative and cellular... An image texture is described by the number and types of its (tonal) primitives and the spatial organisation or layout of its (tonal) primitives... A fundamental characteristic of texture: it cannot be analysed without a frame of reference of tonal primitive being stated or implied. For any smooth gray-tone surface, there exists a scale such that when the surface is examined, it has no texture. Then as resolution increases, it takes on a fine texture and then a course texture." [39]

- "Texture is defined for our purposes as an attribute of a field having no components that appear enumerable. The phase relations between teh components are thus not apparent. Nor should the field contain an obvious gradient. The intent of this definition is to direct attention of the observer to the global properties of the display - ie, its overall 'courseness,', 'bumpiness,' or 'fineness.' Physically, nonenumerable (aperiodic) patterns are generated by stochastic as opposed to deterministic processes. Perceptually, however, the set of all patterns without obvious enumerable components will include many deterministic (and even periodic) textures." [40]

- "Texture is an apparently paradoxical notion. On the one hand, it is commonly used in the early processing of visual information, especially for practical classification purposes. On the other hand, no one has succeeded in producing a commonly accepted definition of texture. The resolution of this paradox, we feel, will depend on a richer, more developed model for early visual information processing, a central aspect of which will be representational systems at many different levels of abstraction. These levels will most probably include actual intensities at the bottom and will progress through edge and orientation descriptors to surface, and perhaps volumetric descripters. Given these multi-level structures, it seems clear that they should be included in the definition of, and in the computation of, texture descriptors." [41]

- "The notion of texture appears to depend upon three ingredients: (i) some local 'order' is repeated over a region which is large in comparison to the order's size, (ii) the order consists in the nonrandom arrangement of elementary parts, and (iii) the parts are rougly uniform entities having approximately the same dimensions everywhere within the textured region." [42]

As can be seen from these examples, the definition of texture varies considerably between researchers and applications. Some of these definitions are perceptually motivated, while others are almost completely driven by a particular application.

Julesz, one of the pioneers of texture analysis, instead proposed a definition for similarity of textures relying on pre-attentive human perception over a brief time period [43]. Two textures are thus considered alike if, when a sample of one is embedded within the other, they are pre-attentively indistinguishable, with no obvious border visible. This definition, however, is reliant upon the discriminatory model of the human visual system, which is not necessarily ideal or desirable in many applications. There are many situations, for example, in which textures easily perceived as different by a human observer should be classified as alike, and conversely where two importantly different textures are pre-attentively indistinguishable to the casual human oberserver.

In his initial work, Julesz conjectured that textures with identical second-order statistics were not preattentively discriminable by humans, and thus constituted equal textures [44, 45, 46]. In this context, second-order statistics of an image represent the distribution of grey-level values of pixels pairs in the image separated by distance $\mathbf{d}$, where $\mathbf{d}$ represents the two-dimensional distance between the pixels. Thus, if two textures have identical second-order statistics, the number of pixel pairs with grey-level values (i, j), separated by $\mathbf{d}$, will be equal for all $\{i, j, \mathbf{d}\}$. In practice, $\mathbf{d}$ is usually replaced with $\{d, \theta\}$, with θ representing the angle between the two pixels, usually restricted to values of $\{0, \pi/4, \pi/2, 3\pi/4\}$. An example of two textures with identical second-order statistics is shown in figure. Note that although at first inspection the image appears to be a single

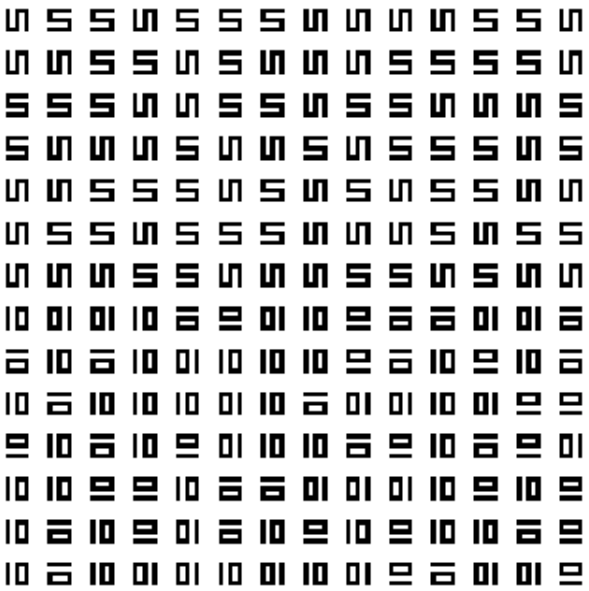


Figure 3.1: Example supporting Julesz' conjecture that textures with identical second order statistics are pre-attentively indistinguishable.

texture, closer examination will reveal that the two halves of the image contain a significantly different structural element.

Julesz' original conjecture has since been shown to be false, with numerous counter-examples published by both Julesz himself and other researchers. One such example is shown in figure 3.2, with three images having identical second-order statistics being clearly preattentively different.

The second theory proposed by Julesz is the so-called *texton* theory, which is based upon first-order measures of local image features, the so-called textons [47]. Features such as length, width, total area and orientation of these elements are then used for further analysis of the texture. One important omission from this research, however, was any discussion on the process of extracting these textons from an image.

More recent work in texture analysis has tended towards a filter-based approach. Such filters are generally designed to extract specific spatial-frequency informa-

(a) (b) (c)

Figure 3.2: Counter-examples to Julesz original conjecture. All three of these images have identical second-order statistics, but are easily preattentively distinguished by human observers

tion from the image, and use this information to classify or segment the texture. Typically, such schemes utilise many filters, which respond to different frequency bands and/or orientations. This multi-channel approach to texture analysis is intuitively appealing because it allows us to exploit differences in dominant sizes and orientations of different textures. Lending support to this method of analysis is the hypothesis that the human visual system operates in a similar way, with the retinal image being decomposed into a number of filtered images, each of which contains intensity variations over a narrow range of frequency and orientation [48]. Subsequent experiments using the visual cortex cells of monkeys showed that each responded only to a narrow range of frequency and orientation [49]. As such, texture is now usually defined in terms of spatial frequency analysis of an image, and various properties of such.

3.3 Tasks in Texture Analysis

There are a number of active areas of research which fall with the broader label of texture analysis. Although the methodologies behind the extraction of texture information for each of these problems is similar, the resulting use of such information is somewhat different. Four of the major fields falling into this

category that will be covered in this section are texture classification, texture segmentation, texture compression and texture synthesis.

3.3.1 Texture Classification

Texture classification is the process of identifying unknown samples of textured images by assigning it to a known class. This is generally a two-stage process involving first training then using the classifier, although some rule-based methods have been developed which do not require training, but rather rely on hard-coded rules regarding the texture models to be classified. Regardless of which approach is taken, some *a priori* knowledge of the classes to be recognised is required.

The first step of the classification process is training, where known texture images are used to train a classifier. Training typically involves four major steps. These are:

1. Image pre-processing
2. Sampling
3. Feature extraction
4. Classifier training

The pre-processing step typically is used for image enhancement and noise removal, although some techniques also perform scaling and rotation in this step as well, in order to compensate for variations in the training data [50].

The most important stage of the process is the feature extraction stage, at which time the sample image is transformed into a much lower dimensionality feature vector. Many different techniques have been used for this stage, and these will be discussed in detail in section 3.4.

The vectors from all of the training images are then input to the classification system for training. Again, many different classifiers have been used, and although some perform slightly better than others, generally the choice of classifier has the least effect on the overall performance of the system. Therefore, speed of training, ease of implementation, and suitability to a given task are more important factors in the choice of a classifier than is raw performance. Typical classification systems include Bayesian classifiers, neural networks, support vector machines and hidden Markov models. Experiments using Gaussian mixture models (GMM) have also provided excellent results in texture classification.

3.3.2 Texture Segmentation

Texture segmentation, unlike classification, is not concerned with determining which specific textures are within an image. Rather, the goal of this type of process is to segment a given image into regions that contain similar texture, or lack thereof. To accomplish this, appropriate measures of texture are needed in order to decide whether a given region has uniform texture. Sklansky [38] has suggested the following definition of texture, which is appropriate in the segmentation context: " region in an image has a constant texture if a set of local statistics or other local properties of the picture are constant, slowly varying, or approximately periodic."

Thus, when attempting to segment an image into regions of approximately equal texture, a local measure describing one or more features of the texture is required. Similar features to those used in texture classification are typically used, with regard given to maintaining the localisation of the information. Pure statistical methods calculated over a small region were amongst the first features used in texture segmentation. Texton theory saw a new method established, whereby individual textons, or blobs, are extracted from an image, and the properties of these textons then calculated over small regions. More recently, multi-channel

filtering has been employed to obtain localised spatial-frequency information of a texture. Typically, some kind of windowing function or non-linearity is used to calculate an energy measure at each point of the texture image.

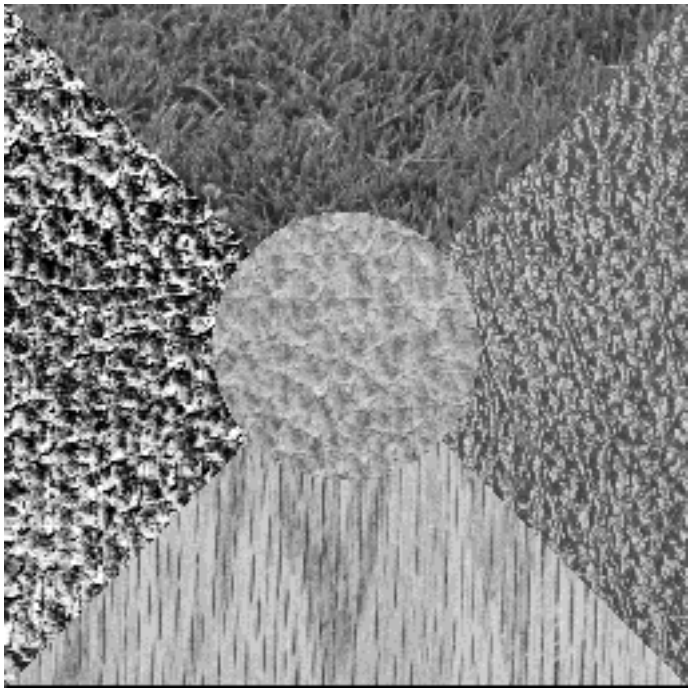
Once such features have been calculated at all areas of an image, generic boundary or edge detection algorithms may be employed to determine the location of texture edges. Alternatively, clustering algorithms may be employed in order to group all regions of similar texture together. A significant drawback to using this technique is that it is often necessary to know the number of classes, in our case distinct textures, in advance. Figure 3.3 shows an example of texture segmentation.

3.3.3 Texture Synthesis

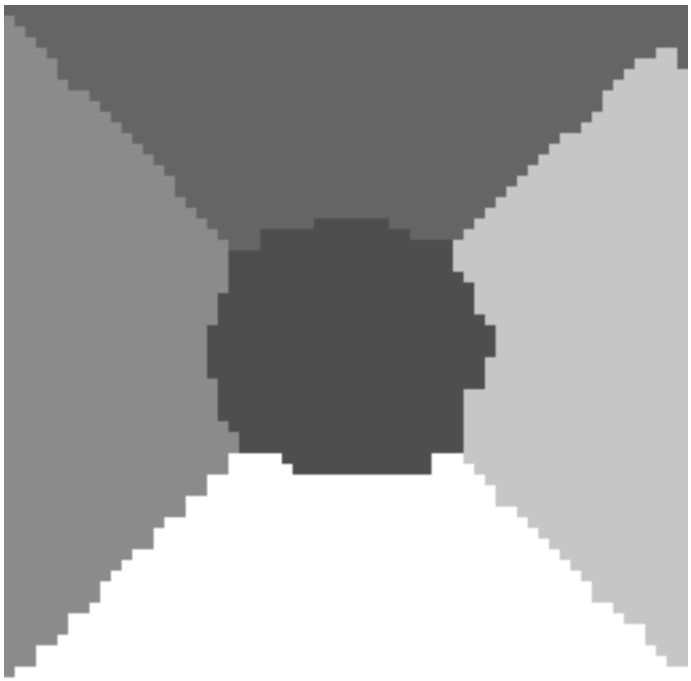
Texture synthesis is the task of creating an artificial sample of texture of any size, usually from a supplied source image, which appears natural to a casual observer. There are a number of applications which utilise such algorithms, for example computer graphics and low bitrate image coding. Texture synthesis is also often used to 'grow' small samples of a given texture into a larger size, without showing an unwanted tiling effect.

A number of methods of texture synthesis have been presented in the literature. Becchetti and Campisi use a combination of binomial linear prediction and a morphological model to characterise textures in a low dimensional feature space [51]. New samples of these textures can then be synthesised using a reconstruction filter based on such parameters.

A texture synthesis algorithm which has shown excellent results on a wide range of images has been proposed by Portilla and Simoncelli [52]. In this technique, statistical properties from a number of sources are used as constraints to synthesise a new texture from a starting point of Gaussian white noise. A steerable pyramid wavelet decomposition is used to extract many of these features, which

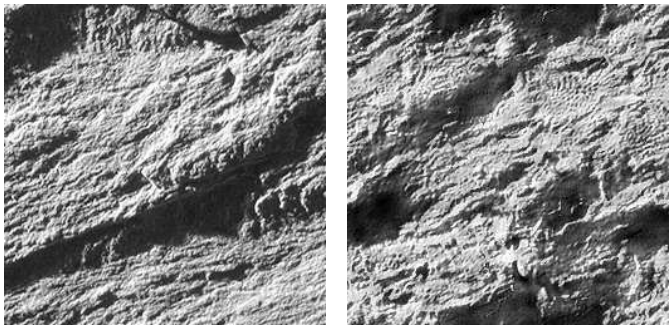


(a)



(b)

Figure 3.3: Example of texture segmentation showing (a) original image, and (b) segmented image



(a) (b)

Figure 3.4: Example of texture synthesis. (a) Original image. (b) Texture synthesised using Portilla and Simoncelli's method.

include

- statistics of original pixel values,
- autocorrelation of the approximation images at each resolution level,
- autocorrelation and correlation of coefficient magnitudes at neighbouring spatial locations, orientations and scales,
- phase correlation at neighbouring resolution levels.

An example of texture synthesis using this method is shown in figure 3.4. It can be seen from this example that the synthesised image, while not identical to the original, effectively captures the distinguishing characteristics of the texture while maintaining a necessary degree of randomness. While the final results of this technique are very impressive, it has the drawback of being extremely computationally expensive, and is therefore unsuitable for real-time or computer graphics applications.

3.3.4 Shape From Texture

Determining the shape of an object in three dimensional shape is an important task in image processing, and there exist many features in images that allow the viewer to make such a determination, for example variations in intensity on the surface of objects, the relative positions and orientations of edges and corners, and shadowing effects. Texture is another property which can be used to determine the relative orientation of a surface [53].

3.4 Texture Analysis Methodologies

To date, a number of significantly different approaches to modelling texture have been proposed, ranging from using the attributes of individual texture elements, to simple statistical measurements of grey level values, random field models, linear filtering, and more recently, features obtained from multi-resolution decompositions. In this section, an overview of a few such broad categories is presented, citing examples of use, relative performance, and other advantages and disadvantages of each.

3.4.1 Autocorrelation Features

Many texture exhibit a significant degree of repetition in the placement of their primitive structural elements. Indeed, it is not uncommon for textures to possess a degree of periodicity. The autocorrelation function of an image can be used to give a measure of such regularity, as well as an estimate of the coarseness of the texture elements. The autocorrelation of an image $I(x, y)$ can be given by

$$\rho(x, y) = \frac{\sum_{u=0}^{N} \sum_{v=0}^{N} I(u, v) I(u + x, v + y)}{\sum_{u=0}^{N} \sum_{v=0}^{N} I^2(u, v)} \tag{3.1}$$

The result of this function will depend on the nature of the texture. For textures with large structural element, the value of the autocorrelation function will decay slowly with increasing x, y. Textures with finer element will exhibit a much sharper rate of decay. For those textures with a very repetitive or periodic nature, the autocorrelation function will itself become semi-periodic, with distinct peaks and valleys present.

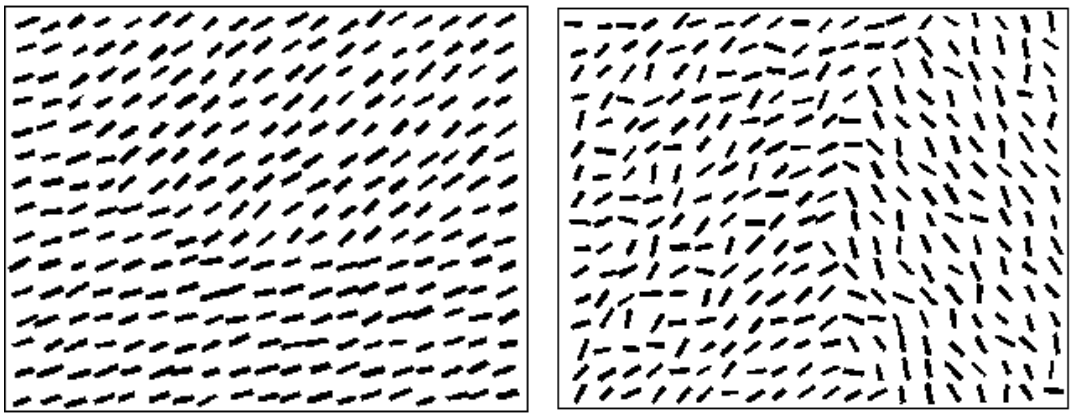
3.4.2 Structural Methods

Structural methods of texture analysis use the properties of a texture's primitive elements, or textons, to either classify or segment a region. Typical examples of such properties are geometric attributes such as height, width, area and orientation. By using statistics of such properties over local neighbourhoods, a useful texture feature is extracted. An alternative approach attempts to describe formulae or rules for the placement and attributes of these textons, and uses these rules to characterise the texture. One of the most significant areas of research in the field of structural texture analysis is the extraction of the individual texture elements, and a large number of approaches to this task have been proposed.

Voorhees has proposed a method to extract texture elements, or 'blobs', based on Laplacian of Gaussian (LoG) filtering. Such filtering is performed at a number of scales, and the results combined to identify the individual textons [54]. The resulting texture primitives are then segmented and analysed to extract such geometric features as height, width, area and orientation, as well as local measures such as blob density. By calculating a local difference function over areas of the image, texture boundaries are detected. By studying the response of human observers to different changes in texton properties, a set of rules for determining which variations are significant in determining a texture boundary is developed. An example of this is shown in figure 3.5. Taking this approach, all information regarding the significance of local variations of the different features is *a priori*,

Texton Property	Min. Perceivable Difference
Mean Orientation	15-20%
Std. Deviation of Orientation	5-20%
Total Size	30%
Height/width ratio	30%
Texton count density	35%
Texton area density	25%

Table 3.1: Minimum deviation of texton properties which are pre-attentively considered to be different by human observers.



(a) (b)

Figure 3.5: Example of textons showing limits of detectable variations in (a) mean orientation, and (b) standard deviation of orientation.

and is pre-determined by psychovisual experimentation on human subjects. Table 3.1 gives some examples of what was found to be a perceptible difference for a number of texton properties. Because of the static nature of this information, this technique is not easily adaptable to more specialised problems where differences not easily observed by humans are to be considered significant.

Tomita and Tsuji have suggested an alternative approach for extracting texture elements by performing a medial axis transform on the connected components of a segmented image [55]. Properties such as shape and intensity of these components are then used to characterise the texture.

3.4.3 Statistical Features

Statistical methods of texture analysis attempt to model an image by measuring the spatial distributions of its pixels' individual grey level values. While the simple statistics of mean and variance are generally not sufficient for the task of texture discrimination, higher order statistical measures were once thought to characterise a texture completely. Although this conjecture has since been proven false, such features provide useful information about a texture and are still in wide use today. Typical statistical measures of a texture include the grey level co-occurrence matrix, grey level difference matrix, and run length features.

Co-occurrence Matrices

Grey level co-occurrence matrices (GLCM's) capture second-order statistical information of an image. The co-occurrence matrix $P_{\mathbf{d}}(i,j)$ of an $N \times M$ image I is defined mathematically as the probability of two pixels, separated by distance vector $\mathbf{d}$, having grey level values of i and j respectively. This can be expressed as

$$P_{\mathbf{d}}(i,j) = \frac{|\{((r,s),(t,v)) : I(r,s) = i, I(t,v) = j\}|}{NM} \tag{3.2}$$

where $|\ |$ represents the cardinality of a set. Due to the variable parameter $\mathbf{d}$, the set of co-occurrence matrices is arbitrarily large. In practice, however, values of $\mathbf{d}$ are typically kept small, and are often expressed in the form (d,θ), with d representing the lineal distance in pixels, and θ the angle between them, generally restricted to the values $\{0°, 45°, 90°, 135°\}$. For computational reasons, d is generally limited to a small range of values. Another common modification to the GLCM calculation is to ensure diagonal symmetry of the matrix. This is achieved by the transformation

$$P_{(d,\theta)} = P_{(d,\theta)} + P_{(-d,\theta)} \tag{3.3}$$

Although some directional information is lost by this transform, it results in a more efficient implementation, and a reduced number of total features, since

matrices with $d < 0$ can be discarded.

Since the grey levels of each pixel are used as the indices to the GLCM, the image data must be of a discrete nature, with a finite number of levels. Thus, an important practical consideration to be made when calculating the GLCM of an image is the quantisation strategy to be used. While images are often stored in a fixed number of grey levels (eg. 256), this may not be appropriate for calculating the GLCM. Using too fine a quantisation step can result in a sparse co-occurrence matrix, leading to some meaningless and non-robust features. Conversely, having too few quantisation levels will almost certainly result in a loss of information and discrimination power. This topic is more thoroughly dealt with in Chapter 4.

While the GLCM of an image conveys useful information, it is generally too large to be usefully used for any kind of segmentation or classification task. A number of features have been extracted from the GLCM which attempt to describe various properties of the image such as energy, entropy, and cluster information. A list of the more commonly used of such features is described in Table 3.2 [56]. Each such feature corresponds to a visually recognisable property of the image. For example, homogeneous textures will have relatively high values of local homogeneity, and low values of contrast and entropy. Course textures will exhibit have a high entropy value at small values of d, and lower values as d increases. Tomita *et al* have also shown a relationship between the GLCM features and the autocorrelation features presented in 3.4.1 [55].

Although GLCM features have proven to be useful in many texture classification tasks, they have a number of disadvantages [57]. The features extracted depend heavily on the choice of (d, θ), and there is currently no established method of choosing these parameters for optimal texture characterisation. Calculating the GLCM for a large number of distances and angles is very computationally expensive, and leads to an excessive number of total features.

Co-occurrence features	Expression
Energy	$\sum_i \sum_j P^2(i,j)$
Entropy	$-\sum_i \sum_j P(i,j) \log P(i,j)$
Inertia	$\sum_i \sum_j (i-j)^2 P(i,j)$
Contrast	$\sum_i \sum_j P(i,j)\lvert i-j\rvert^k, \quad k \in \mathbb{Z}$
Local Homogeneity	$\sum_i \sum_j \frac{1}{1+(i-j)^2} P(i,j)$
Max. Probability	$\max_{i,j} P(i,j)$
Inverse Difference Moment	$\sum_i \sum_j \frac{P(i,j)}{\lvert i-j\rvert^k}, \quad i \neq j, \quad k \in \mathbb{Z}$
Cluster Shade	$\sum_i \sum_j (i - M_x + j - M_y)^3 P(i,j)$
Cluster Prominence	$\sum_i \sum_j (i - M_x + j - M_y)^4 P(i,j)$
Inf. Measure of Correlation	$\frac{-\sum_i \sum_j P(i,j) \log P(i,j) - H_{xy}}{\max(H_x, H_y)}$

where $M_x = \sum_i \sum_j iP(i,j)$ and $M_y = \sum_i \sum_j jP(i,j)$

$$H_{xy} = -\sum_i \sum_j P(i,j) \log \left(\sum_j P(i,j) \cdot \sum_i P(i,j) \right)$$

$$H_x = -\sum_i \left\{ \sum_j P(i,j) \cdot \log \sum_j P(i,j) \right\}$$

$$H_y = -\sum_j \left\{ \sum_i P(i,j) \cdot \log \sum_i P(i,j) \right\}$$

Table 3.2: Typical co-occurrence features extracted from GLCM's.

Run Length Matrices

A grey-level run is defined as a set of consecutive, collinear pixels having the same grey value. Therefore, the length of such a run is the number of unbroken pixels of that grey value in a given direction. A run length matrix $R_\theta(i,d)$ is therefore defined as the number of occurrences of an unbroken run of exactly d pixels, all having the grey level value i, in direction θ. Similarly to co-occurrence matrices, θ is typically restricted to values of $\{0°, 45°, 90°, 135°\}$.

The run length matrix therefore has a number of rows equal to the number of

Run-length features	Expression
Short Run Emphasis	$\frac{\sum_i \sum_d \frac{R(i,d)}{d^2}}{S}$
Long Run Emphasis	$\frac{\sum_i \sum_d j^2 R(i,d)}{S}$
Grey Level Nonuniformity	$\frac{\sum_i (\sum_d R(i,d))^2}{S}$
Run Length Nonuniformity	$\frac{\sum_d (\sum_i R(i,d))^2}{S}$
Run Percentage	$\frac{S}{\sum_i \sum_d dR(i,d)}$

where $S = \sum_i \sum_d R(i,d)$

Table 3.3: Common features extracted from a run length matrix.

grey levels in the image, and a number of columns equal to the image's largest dimension. As with the GLCM, the image must be quantised to a finite number of levels before the run length matrix can be calculated, and the number of levels chosen strongly effects the final result. Too many levels will result in very short run lengths, giving no meaningful information. Conversely, too few quantisation levels will result in erroneous large runs leading to classification errors.

Features similar to those of the GLCM can be extracted from the run length matrix, with examples of these shown in table 3.3.

3.4.4 Random Field Texture Models

A random field model of texture attempts to form a description of the image that can be used not only for segmentation and classification tasks, but also to synthesize it. Thus, the parameters of the model must capture all essential qualities of the texture.

Markov random fields have been used extensively to model many types of images.

The basis of such a model is the assumption that for a given image, the intensity value of each pixel is dependant only on the intensities of the neighbouring pixels. Formally, each pixel is modelled as a site s on a lattice $S = \{s_1, s_2, \ldots, s_N\}$, with the grey level value of the pixel represented by x_s. The *state space* for the model is the set of all possible grey levels of each pixel, defined as $\Lambda = \{0, 1, 2, \ldots, L-1\}$, where L is the number of grey levels in the image. The *configuration space* for the set of variables $x = \{x_s, s \in S\}$ is given by Ω, and represents the set of all possible images. If the joint probability function on Ω is defined as P, then $P(x) > 0,\ \forall x \in \Omega$. Besag [58] has shown that such a probability density function is uniquely determined by its local conditional probability density functions (LCPDF) $P_s(x_s|x_r,\ r \neq s)$.

An MRF tightens this definition by allowing each pixel only to be described in terms of its local neighbourhood $\{r \in \mathcal{N}_s \subset S\}$, where $\mathcal{N}_s$ refers to the local neighbourhood of s. The set of all neighbourhoods of an image, known as the *neighbourhood system* is defined as $\mathcal{N} = \{\mathcal{N}_s, s \in S\}$. Hence

$$P_s(x_s|x_r,\ r \neq s) = P(x_s|x_r,\ r \in \mathcal{N}_s) \quad s \in S, x \in \Omega \tag{3.4}$$

For homogenous MRF's, it is also required that the neighbourhoods are symmetrical, ie. $s \in \mathcal{N}_t \Leftrightarrow t \in \mathcal{N}_s$. One typical neighbourhood $\mathcal{N}^o = \{\mathcal{N}_s^o, s = (i,j) \in \mathcal{S}\}$ is given in [59] as

$$\mathcal{N}_s^o = \{r = (k,l) \in \mathcal{S} : \mathbf{0} < (k-i)^2 + (l-j)^2 \leq o\} \tag{3.5}$$

where o is the *order* of the neighbourhood, and is related to the size of the region. Neighbourhoods of order $o = 1, 2, 8$ are shown in figure 3.6.

Estimates of the LCPDF for a given sample image can be calculated in a number of ways. Paget and Longstaff [60] use multi-dimensional histograms built using the Kronecker function to build a random field model for texture synthesis, where the dimensionality of this histogram is dependent on the order of the neighbourhood used. A non-parametric approach using Parzen windows is then used to estimate the LCPDF from such a histogram. Davidson *et. al.* use a form

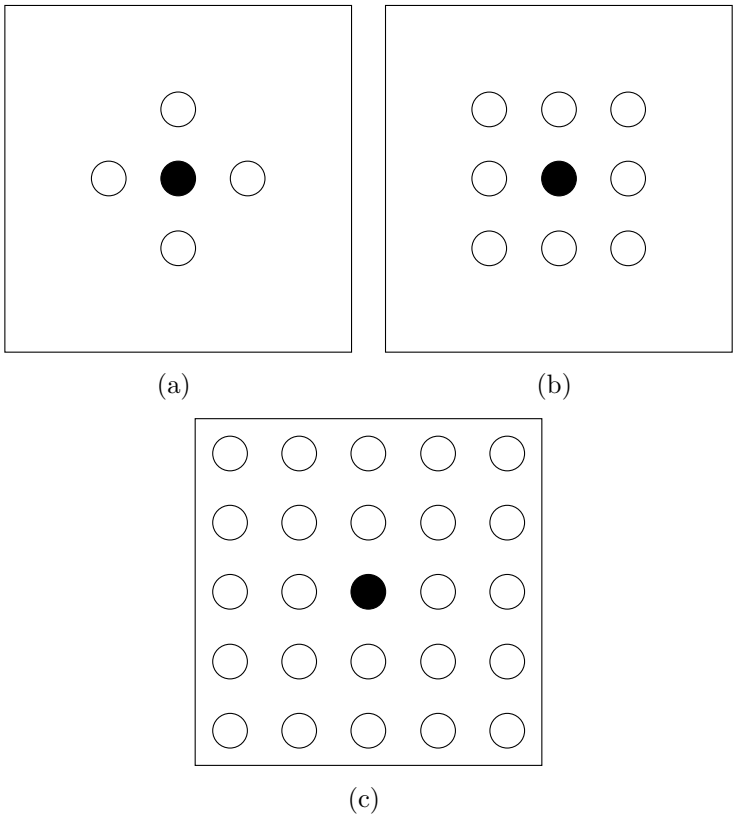


Figure 3.6: Neighbourhoods used for MRF of order (a) 1, (b) 2, and (c) 8

of a Markov mesh model to define the probability function of a textured image, allowing an explicit closed form of the corresponding joint probabilities [61]. Additional examples of MRFs used for texture synthesis and classification are given in [62, 63]. Speis and Healey present a summary of a number of MRF methods of texture analysis in [64].

A similar method of modelling texture is based on the Gibbs random field (GRF), in which the entire lattice is characterised using a probability mass function

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} e^{-U(\mathbf{x})} \quad \forall \mathbf{x} \in \mathbf{\Omega} \tag{3.6}$$

where $U(\mathbf{x})$ is an energy function usually specified in terms of neighbourhoods, and Z is a normalising constant known as the partition function. $U(\mathbf{x})$ is generally expressed in terms of *cliques*, which are sub-regions of the local neighbourhood consisting of pairs, triples and quadruples of pixels. Numerous methods of defin-

ing such an energy function in a parametric form have been proposed in the literature, with examples given in [65, 66].

There exists a duality between the MRF and GRF representations of an image using the same neighbourhood system, such that for every MRF there exists a unique GRF and vice versa [58]. This means that using such fields, a texture can be modelled either globally by specifying the total energy of the lattice, or locally by specifying the interactions between neighbouring pixels in the form of conditional probabilities.

3.4.5 Texture Filters

The literature presents a number of different filtering techniques for texture analysis, which, while similar in concept, differ significantly in approach and implementation details. One of the pioneering techniques for texture analysis using filtering was proposed by Laws, who used a bank of band-pass filters to characterise texture [67]. Since then, numerous different filter-based techniques have been suggested, many using Gabor or Gabor-like filter-banks, or the more recent wavelet transform. This section will give a brief overview of a number of such approaches, with the exception of the wavelet transform-based methods which will be discussed separately in section 3.4.6

Laws Filter Masks

One of the first filtering approaches to texture analysis was proposed by Laws [67], and was based on the observation that gradient operators such as the Sobel and Laplacian filters often enhanced the microstructure of many textures. He has suggested the use of a bank of separable filters, using five band-pass filters along each axes of the image. The coefficients of these filters and their descriptions are shown in 3.4. Using this scheme, a total of 25 subband images are created, from

Filter	Coefficients
h1 (Level)	$[1, 4, 6, 4, 1]$
h2 (Edge)	$[-1, -2, 0, 2, 1]$
h3 (Spot)	$[-1, 0, 2, 0, -1]$
h4 (Wave)	$[-1, 2, 0, -2, 1]$
h5 (Ripple)	$[1, -4, 6, -4, 1]$

Table 3.4: Coefficients of the one dimensional Laws texture filters

which a number of statistical features such as energy, variance, and higher order moments can be extracted.

Gabor Filters

In recent times, Gabor filters have emerged as one of the most commonly used techniques in the field of texture analysis. These filters are a particularly attractive choice as they are optimal in a time-frequency sense. That is, their resolution in time and frequency satisfies the equality condition of the Heisenberg inequality of (2.12) presented in section 2.2.2. The use of Gabor filters was also inspired in part by studies of the human visual system. Psychovisual studies of human subjects has indicated that the visual system responds at some level to both frequency and orientation input. Examination of the visual cortex of Macaque monkeys has revealed cells which seemingly respond only to particular spatial frequencies and orientations [48]. Such observations, combined with the previously hypothesised pre-attentive nature of texture discrimination, indicate that a bank of spatial frequency and orientation tuned filters, such as Gabor filters, would be suitable for texture analysis [68].

In essence, a two dimensional Gabor filter consists of a sinusoidal plane wave modulated by a two dimensional Gaussian envelope. The impulse response of one possible Gabor filter is given by

$$g(x, y) = \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\}\cos(2\pi u_0 x) \tag{3.7}$$

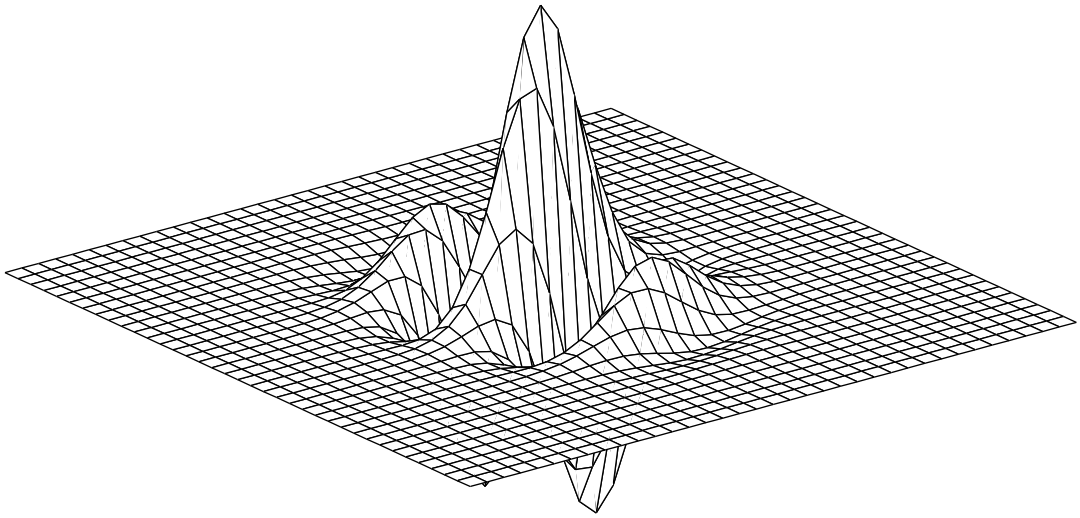


Figure 3.7: Spatial domain representation of a real Gabor function

where u_o is the frequency of the plane wave, and σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes respectively. This example of a Gabor filter is known as a *symmetric* Gabor filter, as it is symmetrical about the y axis. An *asymmetric* Gabor function is also defined as

$$g(x,y) = \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\}\cos(2\pi u_0 x - \pi/2) \tag{3.8}$$

It is possible to modify the orientation of the sinusoidal component to any angle θ by means of a rigid rotation of the coordinate system, defined by

$$x = x\cos\theta + y\sin\theta \tag{3.9}$$

$$y = -x\sin\theta + y\cos\theta \tag{3.10}$$

Figure 3.7 shows an example of a symmetric Gabor function in the spatial domain.

Gabor filters are essentially bandpass filters which can be tuned to any spatial frequency and orientation. This can be seen by the Fourier transform of (3.7), which can be shown to be

$$\begin{aligned} G(u,v) = {} & A\left[\exp\left\{-\frac{1}{2}\left(\frac{(u-u_o)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)\right\}\right] + \\ & A\left[\exp\left\{-\frac{1}{2}\left(\frac{(u+u_o)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)\right\}\right] \end{aligned} \tag{3.11}$$

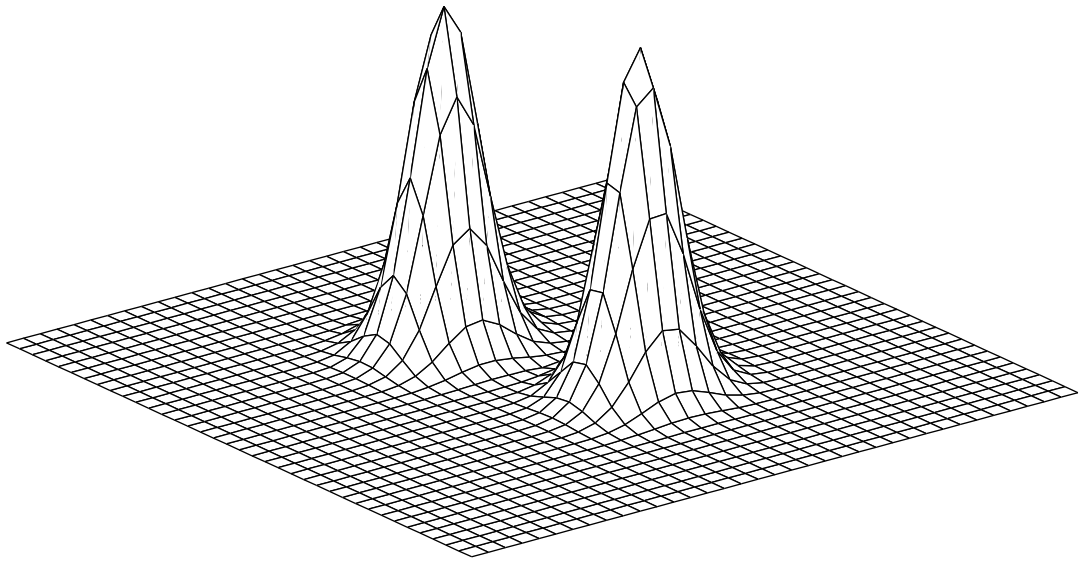


Figure 3.8: Frequency spectrum of Gabor function in figure 3.7

where $A = 2\pi\sigma_x\sigma_y$, $\sigma_u = \frac{1}{2\pi\sigma_x}$ and $\sigma_v = \frac{1}{2\pi\sigma_y}$. A graphical representation of this is shown in figure 3.8. The frequency and orientation bandwidths, B_r and B_θ, of the Gabor filter are also of great importance, and can be calculated by [69]

$$B_r = \log_2 \left(\frac{u_0 + (2\ln 2)^{1/2}\sigma_u}{u_0 - (2\ln 2)^{1/2}\sigma_u} \right) \tag{3.12}$$

$$B_\theta = 2\tan^{-1} \left(\frac{(2\ln 2)^{1/2}\sigma_v}{u_0} \right) \tag{3.13}$$

where B_r is in octaves, B_θ is in radians, and each represents the 3dB bandwidth of the Gabor spectrum. Using (3.12), (3.12) and (3.13) it is possible to design Gabor filters with arbitrary frequency, orientation and bandwidth. By combining a selection of such filters, the entire frequency spectrum can be adequately covered, allowing excellent signal representation.

A large number of texture feature sets have been proposed in the literature using the Gabor filters thus defined. Jain and Farrokhnia use banks of symmetric Gabor filters to segment images of natural textures [70]. The outputs of these filters undergo transformation using a non-linear function followed by local energy computation in order to calculate a texture feature vector at each pixel location. In order to utilise spatial information, the physical coordinates of pixel are also

included in this vector. Also proposed is a method for reducing the number of filters used in order to increase computational efficiency. Similar segmentation and classification algorithms based on Gabor energies are also presented in [69, 71, 72].

More recently, a new set of texture features based on the *grating cell operator* has been proposed [73, 74]. This non-linearity applied to the output of Gabor filterbanks responds only when at least three bars is present in the local field. Studies of the visual cortex of monkeys has found the existence of cells which react in a similar manner when presented with such a stimulus [49].

Another approach to characterising textures using Gabor functions is to match the filters in some way to the textures being analysed. By analysing the spectral feature contrasts obtained from successive Gabor filtering at multiple scales, the parameters of a tuned filter can be estimated [75]. It has been shown that more accurate and efficient segmentation can be achieved by using the outputs of such filters. Dunn and Higgins have also shown that improved segmentation results are possible if one uses *a priori* information about the textures likely to be present when choosing the parameters of the Gabor filters [76].

Because of their selectivity in both scale *and* orientation, Gabor filters can be used to successfully identify textures at particular orientations. By applying a rotation-invariant transform to these features, it is then possible to create a set of texture descriptors which is invariant to rotation. The Fourier transform has been used in this regard to formulate a set of rotation invariant texture features for discriminating between different types of printed text regardless of orientation [77].

3.4.6 Wavelet Texture Features

Over the last two decades, the wavelet transform has emerged to provide a more formal, solid and unified framework for multiscale signal analysis, with implementations that are generally more efficient than existing equivalent methods. Compared to the Gabor filtering approach presented above, the wavelet transforms cover the frequency domain more exactly, can reduce correlations between the bands of the decomposition, and allow adaptive pruning of the transform leading to increased computational efficiency [78]. Texture analysis is one of many applications which has seen significant use of the WT, with many algorithms for extracting texture features for segmentation, classification and synthesis proposed in the literature.

The earliest such features involved calculating the energy, or a similar measure, present in each of the subbands resulting from the wavelet decomposition of an image [79, 80, 81]. Such features have been shown to perform reasonably well in classification and segmentation tasks, with accuracy similar to or exceeding that of the Gabor energy features previously described. Using multiple analysing wavelets has also been shown to improve the overall classification accuracy, as each can detect different characteristic features of textures [82].

Other first order statistics are computed by constructing a histogram $h(u)$ of the wavelet coefficients at each level. To obtain such a histogram, uniform quantisation of the coefficients is used. Observations of such histograms for many natural textured images has led to the conjecture that they may be adequately modelled by the parameters of the generalised Gaussian function defined as [2]

$$h(u) = Ke^{-\left(\frac{|u|}{\alpha}\right)^{\beta}} \tag{3.14}$$

The parameters α models the width of the histogram peak, while β in inversely proportional to the rate of decay of the function. The normalisation constant K ensures that the function remains a true probability density function by having

a total area of 1. These parameters can be calculated from the histogram $h()$ via

$$\beta = F^{-1}\left(\frac{m_1^2}{m_2}\right) \tag{3.15}$$

$$\alpha = m_1\frac{\Gamma(1/\beta)}{\Gamma(2/\beta)} \tag{3.16}$$

$$K = \frac{\beta}{2\alpha\Gamma(1/\beta)} \tag{3.17}$$

where

$$m_1 = \int |u|h()du \tag{3.18}$$

$$m_2 = \int u^2h(u)du \tag{3.19}$$

$$\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}dt \tag{3.20}$$

$$F(x) = \frac{\Gamma^2(2/x)}{\Gamma(3/x)\Gamma(1/x)} \tag{3.21}$$

This form of generalised Gaussian function has been experimentally shown to give a good approximation of the histograms of wavelet coefficients for natural images [83].

The wavelet packet transform has also been extensively used in texture analysis problems, as it allows greater resolution in the frequency domain. Laine and Fan use an undecimated wavelet packet representation to provide an overcomplete representation of textures for segmentation [78]. By applying the Hilbert transform to the output signals of such a transform, a translation-invariant energy estimation at each position of the lattice is obtained and then used for unsupervised texture segmentation. Lee and Pun use the wavelet packet transform to select only the dominant energy bands for use as texture features, allowing for good classification accuracy at reduced computational expense [84]. Wang has also used the wavelet packet transform in a multiwavelet environment for texture segmentation using a novel evolutionary algorithm [85]. In this work, the extrema density of the wavelet coefficients is used as a measure of the coarseness of a texture at a given location, and provides good segmentation results.

By removing the downsampling step of the FWT, a translation invariant, over-

complete wavelet decomposition of a signal is obtained. Using such a representation when extracting features for texture analysis has the advantages of greater spatial resolution, more robustness against translation, and allowing greater confidence when extracting statistical features due to a larger number of coefficients. Using such a transform, Van de Wouwer *et. al.* extract first and second order statistical features from each band of the decomposition, and use these features for a classification task with excellent results [83].

3.5 Applications of Texture Analysis

Texture analysis has been used to date in a wide variety of applications, including remote sensing, surface inspection and flaw detection, medical image analysis, computer graphics and document processing. The role that texture analysis plays in these applications is widely varying. For example, texture classification has been successfully used to classify regions of SAR imagery into categories such as water, farmland, forest, etc. In the field of medical image analysis, texture information has been employed to accurately segment ultrasound images into various regions based upon textural properties. In the field of computer graphics, texture compression, modelling, and synthesis have all been used to great effect to create realistic artificial scenes. Texture analysis techniques have also been used in the more general task of object recognition.

3.5.1 Automated Flaw Detection

To date, there have been a number of applications which employ texture analysis techniques to eliminate the need for human inspection of various materials. Such materials include carpets, automobile paints and most commonly textiles. Dewaele *et al.* [86] uses an adaptive filter-bank implementation to detect point and line defects in textile images. Chetverikov [87] detects boundaries of defects

in textures by using a simple window difference operator over texture features obtained by filtering. Chen and Jain [88] use a structural approach to detect flaws. A skeleton representation of the texture is first extracted, then statistical anomalies in these skeletons are used to detect possible defects. More recently, Bodnarova [89] has successfully used Gabor filterbanks to detect a number of defect types in textiles.

Another application of flaw detection is in the automatic detection and classification of flaws in wood samples. Conners *et al.* [90] have proposed an algorithm by which an image of wood is divided into regions, each of which is classified into categories depending on the defect type present. Such flaws include knots, decay, and mineral streaks. The texture features used to perform this classification are a combination of grey-level statistical features such as mean, variance, skew and kurtosis, as well as second-order statistical features extracted from co-occurrence matrices.

In the area of quality control, Siew *et al.* [91] use second-order grey-level dependency statistics as well as first-order difference statistics to assess carpet wear. Jain [92] uses texture features calculated from a Gabor filterbank to assess the quality of a painted automobile surface.

3.5.2 Medical Imaging

In recent times, the automatic analysis of medical images has lead to a vast improvement in both the speed and accuracy of the diagnosis of many diseases. Medical images, such as ultrasound, MRI and X-ray images, are often unclear and distorted, and in general do not contain clear separation of important objects. These characteristics make such images unsuitable for many traditional object recognition and classification techniques which rely on the detection of object boundaries. Texture analysis, with no such reliance on these features, has been shown to give good results in many medical applications.

Freeborough uses the common spatial co-occurrence matrix features over a number of distances to diagnose Alzheimer's disease from magnetic resonance (MR) images of the brain [93]. A number of other researcher have used similar features to detect microcalcifications in X-ray mammogram images with varying success [94, 95]. Due to the high number of features extracted using this technique, a feature selection scheme is required to choose those most useful for classification. Bleck uses a random field model of texture to detect microfocal lesions of the liver from ultrasound images [96]. Such a model was shown to give better results at detecting such features than the more traditionally used co-occurrence features. A structural texture analysis technique has been successfully developed by Quiang Ji *et al.* to detect cervical lesions from colposcopic images [97]. Vascular structure from the image are extracted to form line segments, which are regarded as the primitive textural elements. Properties of these elements such as length and orientation, as well as statistics of the line-map itself, are then used to classify the image into six classes with an accuracy of over 85%. Taking advantage of the low grey-level resolution of some medical images, Wang [98] uses a local binary pattern (LBP) texture analysis technique to identify various regions of interest from endoscopic images. Texture has also been used to successfully classify and detect pulmonary disease, leukemic malignancies, and white blood cell patterns, as well as to aid in the segmentation of MRI, ultrasound and X-ray images [99, 100, 101, 102, 103, 104].

3.5.3 Document Processing

Document processing is a broad field of computer vision encompassing a number of important tasks. Such tasks include postcode and address retrieval for rapid sorting of mail, indexing and retrieval of large document databases, converting printed documents into electronic form via optical character recognition (OCR), detecting and recognising company logos from scanned correspondence, and many others. Recent research has identified that printed text has a distinct texture,

and used this observation to the development of new algorithms for detecting and segmenting such regions. Texture analysis techniques have therefore shown to give good results when applied to problems of text extraction and recognition. A more thorough review of the use of texture analysis techniques in the field of document analysis is presented in Chapter 7.

3.5.4 Remote Sensing Image Analysis

Remote sensing imagery is obtained in a number of ways, from direct satellite and aerial photography to more advanced methods employing various parts of the electromagnetic spectrum. Recently, synthetic aperture radar (SAR) has emerged as an extremely powerful method of obtaining such data, by utilising the motion of a satellite through space to simulate a much larger antenna than is physically possible. SAR images are used in a number of applications, including

- Sea ice monitoring
- Cartography
- Surface deformation detection
- Glacier monitoring
- Crop production forecasting
- Forest cover mapping
- Ocean wave spectra
- Urban planning
- Coastal surveillance (erosion)
- Monitoring disasters such as forest fires, floods, volcanic eruptions, and oil spills

Due to the nature of SAR images produced by backscatter from various types of material, texture analysis techniques have proven extremely successful in their analysis [105]. The unique noise characteristics of SAR image, in particular speckle noise, means that a somewhat different approach has been taken in when analysing such textures [106, 107].

As in other applications, the texture features used to classify and segment SAR images varies significantly. Kurvonen and Hallikainen [108] use a mixture of first and second order statistical measures to classify land-cover and forest types from SAR images. Autocorrelation and GLCM features were shown in this case to produce good results. Simard *et. al.* [109] use texture features extracted from the wavelet transform to provide information on the tropical vegetation cover of rainforests. Soh *et. al.* uses features extracted from GLCM's to aid in the mapping and classification of sea ice formations.

The literature contains many more examples of the use of textural information in the field of remote sensing. More more information, the reader is referred to [110, 111, 112, 113, 114, 115].

3.6 Chapter Summary

This chapter has presented a review of the field of texture analysis. The problem of defining texture was explained, with a number of differing viewpoints presented over the last four decades listed. The various tasks within the field were then briefly summarised, including texture classification, segmentation, synthesis and compression. Following this, a review of common methods of analysing texture were presented, from early statistical and texton-based approaches to the recent random field and multi-resolution filtering techniques. A number of specific examples of each from each of these approaches were outlined in detail, with the advantages and disadvantages of each noted. Finally, a number of applica-

tion areas for texture analysis were outlined, including automated flaw detection, medical imaging, document processing and remote sensing tasks.

Chapter 4

Quantisation Strategies for Improved Classification Performance

4.1 Introduction

The literature proposes many techniques for classifying textures based on statistics of wavelet coefficients. Such methods include first order statistics such as the variance or energy of each band, second order statistics based on the co-occurrence matrices of coefficients, and random field models [81, 83, 85, 116, 117, 118, 119]. When using these algorithms, it is often required that the coefficients be first quantised to a discrete number of levels, or analysis is performed using the histograms of the coefficients. Although some strategies for performing such quantisation have been proposed, the quantitative effect of this has not been fully evaluated to date.

This chapter shows the results of a detailed study into the effect of wavelet coefficient quantisation on classification performance, and proposes two new feature

sets based on first and second order statistics of wavelet coefficients quantised on a logarithmic scale. Experimental results will show that such a quantisation strategy can better model the distribution of wavelet coefficients, and thus reduce the overall classification error rate.

4.2 Quantisation Theory

Quantisation is the process of mapping a continuous variable onto a finite range of integer values. This operation is necessary whenever data is to be stored in digital form, and is often combined with sampling in what is commonly referred to analog to digital conversion (ADC). For example, image data which is theoretically unlimited in intensity resolution is typically stored using a fixed number of grey levels. Most modern methods of capturing images perform such quantisation automatically, resulting in a final image with, for example, 256 possible intensity values.

Mathematically, the process of quantisation can be expressed in functional form, with the quantisation function $q(t)$ expressed as a sum of the input variable t and an error ε,

$$q(t) = t + \varepsilon \tag{4.1}$$

However, such a representation is of little use, and a more general description of a quantiser can be constructed by means of a set of intervals or *cells* $\mathcal{S} = \{S_i,\ i \in \mathcal{I}\}$, where $\mathcal{I}$ is usually a collection of consecutive integers, combined with a set of *reproduction values* or levels, $\mathcal{C} = \{y_i,\ i \in \mathcal{I}\}$ [120]. The value of the quantisation function $q(t)$ is thus defined as $q(t) = y_i$ for $t \in S_i$, which can be expressed concisely as

$$q(t) = \sum_i y_i 1_{S_i}(t) \tag{4.2}$$

where 1_{S_i} is an indicator function defined as

$$1_{S_i}(t) = \begin{cases} 1, & t \in S_i \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

This definition of a quantiser $q(t)$ is only valid when $\mathcal{S}$ is a true partition of the set of real numbers $\mathbb{R}$, that is, the cells are non-overlapping and exhaustive. Typically, the members of $\mathcal{S}$ are expressed as a range,

$$S_i = (\min S_i, \max S_i] \tag{4.4}$$

and for the usual case where the cells are contiguous, this can be further simplified as

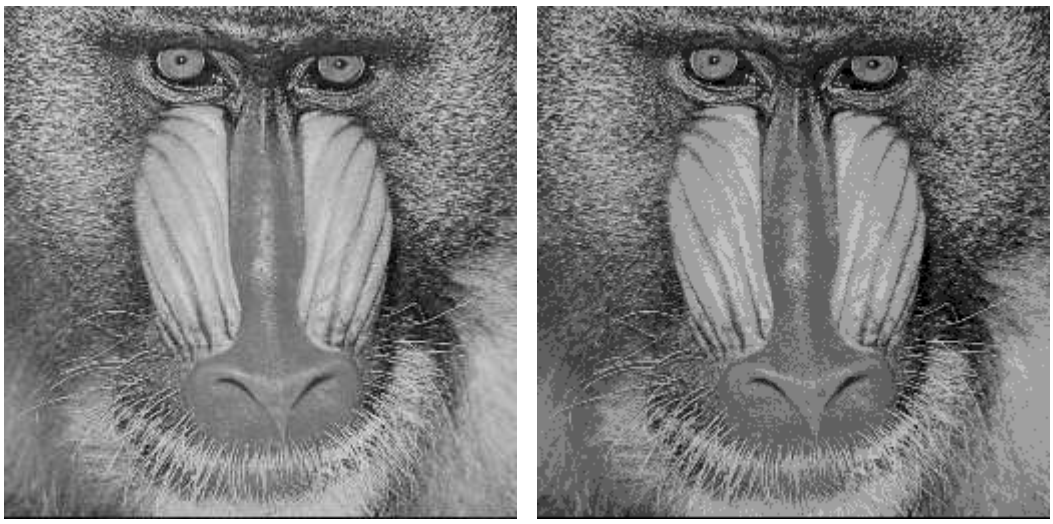
$$S_i = (a_{i-1}, a_i] \tag{4.5}$$

where $\{a_i\}$ is the set of threshold values forming an increasing sequence [120]. In the special case where these threshold values are equidistant, the resulting system is known as a *uniform quantiser*.

An important property of a quantiser is the quality of signal reconstruction that can be obtained. The most common way of measuring such quality is to calculate a distortion measure $d(x, \hat{x})$, which indicates the amount of error introduced when an element x is quantised and reconstructed to form $\hat{x}$. The most typical example of such this distortion function is the squared error,

$$d(x, \hat{x}) = |x - \hat{x}|^2 \tag{4.6}$$

however there are a number of other functions used as well. In practice, more than a single sample is quantised, and a more important measure of the quality of the quantiser is the overall signal distortion. While this can be measured practically by averaging the error from each sample, a more concise definition of the error can be formulated by viewing the signal as a random variable X with a probability density function $f(x)$. The expectation of the signal distortion $D(q)$ can then be calculated by the summation of the expected error from each cell S_i



(a) (b)

Figure 4.1: Effects of quantisation on image quality. (a) Original image with 256 grey levels, and (b) image quantised to 8 levels.

of the quantiser.

$$D(q) = E[d(X, q(X))] = \sum_i \int_{S_i} d(x, y_i) f(x) dx \tag{4.7}$$

From (4.7) it can be seen that as the cell size approaches zero, the signal distortion becomes arbitrarily small.

When dealing with the quantisation of images, perceptible distortion is an important property. Such distortion can only be measured qualitatively by visual inspection of the images, in order to determine the relative quality. Figure 4.2 shows an example of distortion introduced into an image by quantisation to 32 grey-levels.

Quantisation of a signal is required when calculating certain statistical properties. The creation of a histogram requires that the signal be divided into discrete levels, or buckets, in order to obtain the relative frequencies at each location. Calculating second-order statistics of an image, such as the GLCM and run-length matrix features, also requires quantisation, since the indices of these matrices correspond to a grey-level value, and hence these must be discrete. The effect of quantisation in these cases is significant. If too few quantisation levels are used,

then the resolution of the representation will suffer, resulting in a loss of statistical information. Conversely, using too many levels for a signal of insufficient size will result in a sparse histogram or co-occurrence matrix, making the subsequent statistics more susceptible to noise or signal artifacts.

4.3 Quantisation of Wavelet Coefficients

In general, the wavelet coefficients of an image are near-continuous in nature. Therefore, in order to calculate either histograms or second order statistics of these coefficients, some quantisation strategy must be employed. One example of such an approach to texture characterisation is proposed by Van de Wouwer *et. al.*, who use both first and second order statistics of the wavelet coefficients [83]. First order statistics are calculated by constructing a histogram of the wavelet coefficients, and modelling this histogram using a generalised Gaussian function. In order to create the histogram, uniform quantisation of the coefficients is performed.

Second order statistics of the wavelet coefficients are extracted by means of co-occurrence matrix features. Such a matrix is formed using the same procedure as that for GLCM's presented in section 3.4.3. Uniform quantisation is again performed to map the continuous valued wavelet coefficients to the discrete indices of the co-occurrence matrix, however the details of this process are not outlined. In order to reduce the total number of features, the co-occurrence matrices are averaged over the four directions $\{0°, 45°, 90°, 135°\}$, and restricted to a distance of one pixel. Thus, only one co-occurrence matrix from each detail subband is created. Extracting the 8 common co-occurrence features (inertia, energy, entropy, local homogeneity, max. probability, cluster shade, cluster prominence and information measure of correlation) from each such matrix for the first four levels of decomposition gives a total of 96 features, which the authors refer to as *co-occurrence signatures*.

4.4 Logarithmic Quantisation of WT Coefficients

It has been observed by a number of authors that the wavelet coefficients of natural and other textured images are non-Gaussian in nature, and often contain large peaks at the origin and long tails [121, 122, 123]. In spite of this, the vast majority of texture features extracted for classification of such images, such as the energy or mean deviation of each band, assume a Gaussian-like distribution of these coefficients. Examples of the distribution of wavelet coefficients of textured images and the corresponding Gaussian distributions are shown in figure 4.2. These examples show the error which are introduced using such a model. Using a generalised Gaussian function to model the distribution of coefficient, as proposed by Van de Wouwer *et. al.* can in many cases overcome such limitations, however there are still a number of cases where such functions are inadequate.

By modifying the quantisation function $q(t)$, it is possible to significantly alter the shape of the resulting histogram, while still maintaining the relevant information contained within the coefficients. In order to remove the long tails of the histogram that make modelling of the coefficients difficult, a function is required that compacts the histogram at high levels. One example of such a function is the logarithm, and this shall be the focus for further investigation.

Logarithmic quantisation implies that the size of consecutive cells in $\mathcal{S}$ increases in an exponential manner, and the threshold values can be defined by

$$a_i = ka_{i-1} \tag{4.8}$$

which can be solved as

$$a_i = e^{ki} \tag{4.9}$$

where $k > 1$ is a constant determining the relative size of the cells. Such a function leads to very small quantisation cells near the origin, which is undesirable in most cases. To overcome this, an offset value is required such that the size of the first

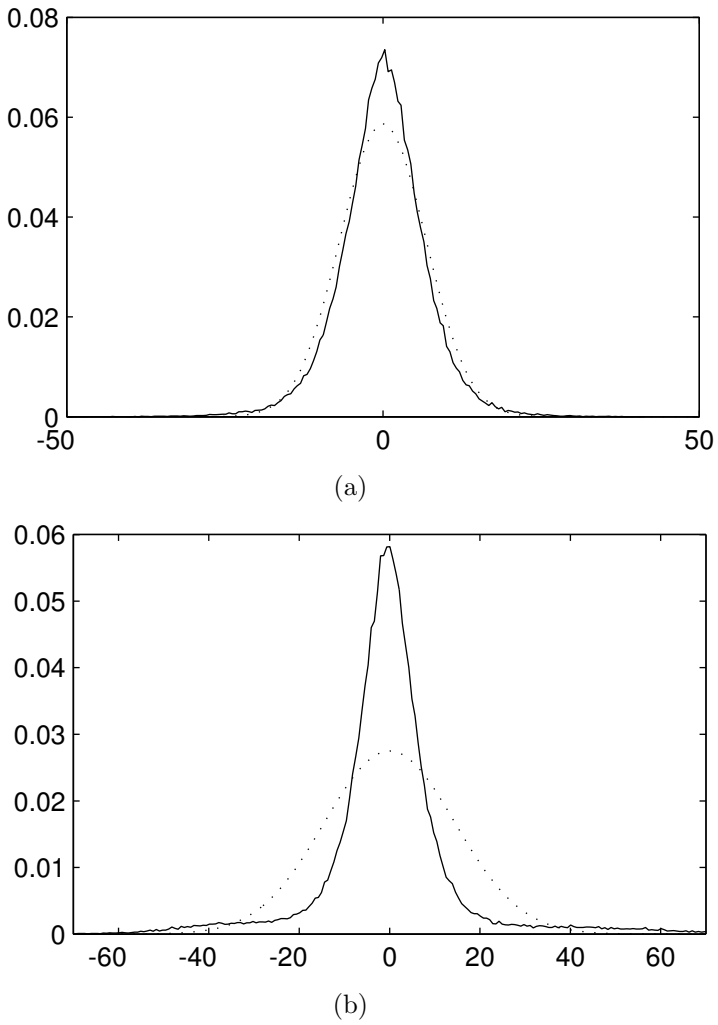


Figure 4.2: Histograms of wavelet coefficients for (a) well matched, and (b) mismatched images. The dotted line shows the Gaussian distribution of equal variance commonly used to model such distributions.

cell is fixed, and the size of every subsequent cell increases exponentially. Hence,

$$a_{i+1} - a_i = k(a_i - a_{i-1}) \tag{4.10}$$

with the solution for i_n given by

$$a_i = e^{e^k i + C} - e^C \tag{4.11}$$

By setting $k = e^k$ this can be simplified to

$$a_i = e^{ki+C} - e^C \tag{4.12}$$

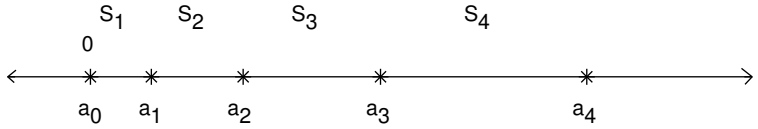


Figure 4.3: Thresholds and cell locations for the logarithmic quantisation function.

From (4.12) it can be seen that $a_0 = 0$, and that the size of cell S_i proportional by a factor e^k to the size of S_{i-1}. Figure 4.3 shows the layout of these thresholds and cells in graphical form. Clearly, (4.12) does not allow for negative thresholds, and hence cannot be applied in this form to negative coefficients. However, by defining $a_n = -a_{|n|},\ n < 0$, the thresholds for positive values of i can be mirrored, allowing for full coverage of $\mathbb{R}$.

A more efficient implementation of logarithmic quantisation of a variable x can be realised by transforming x such that uniform quantisation is achieved. An ideal case of this transform would map each threshold value a_i to the linear center of the new quantisation cell, ie $\frac{2i-1}{2}$. Simple algebra gives this function as

$$f(x) = \frac{\log\left(\frac{x+e^C}{e^C}\right)}{k} - \frac{1}{2} \tag{4.13}$$

By redefining constants, this can be rearranged to give

$$f(x) = \kappa \log\left(\frac{x}{a_I \delta} + 1\right) + 1/2 \tag{4.14}$$

where

$$\kappa = \frac{I-1}{\log\left(1/\delta + 1\right)} \tag{4.15}$$

remembering that I represents the total number of cells in the quantiser, and a_I is the saturation level. Thus, to design a logarithmic quantiser having I levels, knowing the desired saturation point a_I, it is required to choose only one parameter δ, which must be positive, and can be thought of as representing the point on the log curve which is used as a non-linearity. The value of δ is proportional to the similarity of the cell sizes of the quantiser, so that low values indicate vastly different sizes, and high values very similar. As $\delta \to \infty$, the

quantiser becomes a uniform quantiser. Quantisation of $f(x)$ is performed by a simple rounding operation. By removing the addition of $\frac{1}{2}$ from (4.14), it is possible to make the quantiser symmetric about the origin.

The non-linear transform described in (4.14) is similar to that proposed by Unser and Eden, who suggested that a logarithmic function applied to wavelet coefficients yields a more stable representation and better class separation in the context of texture segmentation [124]. In their work, a second non-linearity was also applied to the coefficients before the logarithm function in order to remove the sign of the coefficients. Since the wavelet detail coefficients are zero-mean, and their histograms generally near-symmetric about the origin (see figure 4.2) [83], the sign of each coefficient is relatively unimportant and such an operation results in little loss of useful information. In [124], Unser and Eden show that taking the square of the coefficients leads to the best characterisation of the textures. Modifying (4.14) to include this operator gives

$$f(x) = \kappa \log\left(\frac{x^2}{a_I^2\delta} + 1\right) \tag{4.16}$$

Another possibility for this rectifying function is the magnitude operator $|\cdot|$, which gives

$$f(x) = \kappa \log\left(\frac{|x|}{a_I\delta} + 1\right) \tag{4.17}$$

The effect of the proposed quantisation system on the wavelet coefficients of textured images is best described by the histograms of these coefficients. Figure 4.4 shows an example of a typical histogram of wavelet coefficients, and the effect of both uniform and logarithmic quantisation. In both cases, the squaring function is used to rectify the coefficients, and the redundant undecimated wavelet transform used in place of the usual FWT. Using the undecimated version of the transform provides a more robust representation with respect to noise, and allows for better reconstruction after quantisation [78, 83]. In addition, the removal of the decimation allows for a less sparse histogram and therefore a more accurate texture model. The histograms show that logarithmic quantisation improves the modelling of the coefficients when compared to uniform quantisation

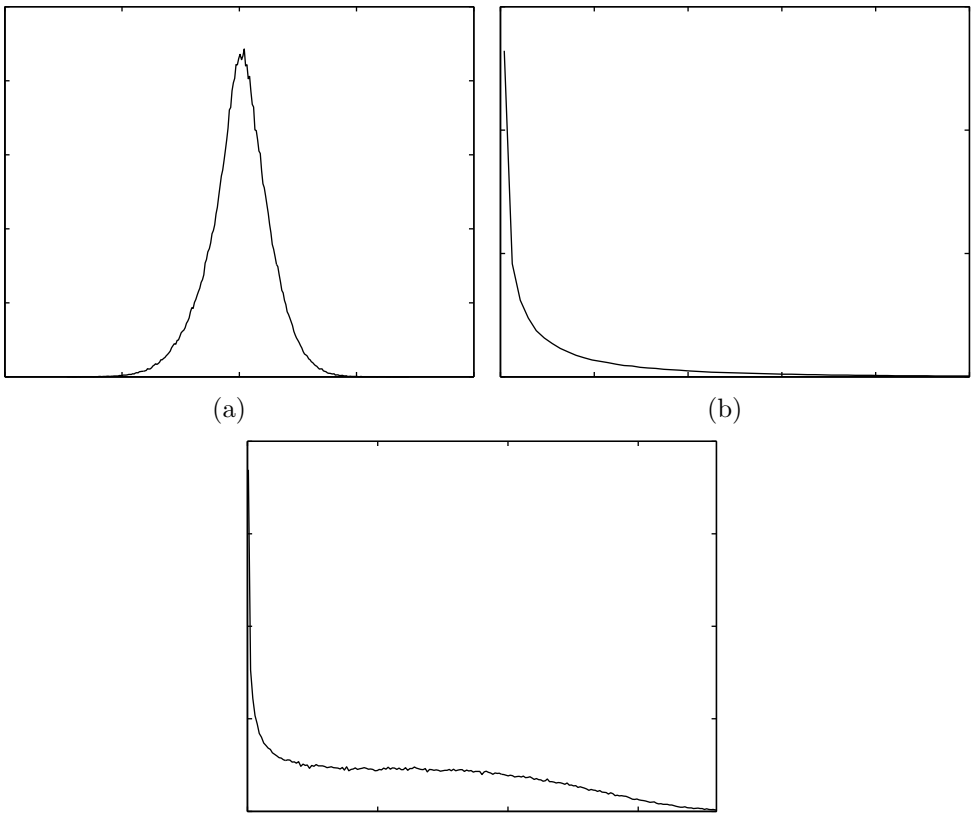


Figure 4.4: Histograms of texture wavelet coefficients. (a) Original histogram, (b) uniform quantisation, and (c) logarithmic quantisation ($\delta = 0.001$).

by shortening the tail of distribution, and reducing the height of the peak at the origin.

4.4.1 Image Distortion

The application of the logarithmic quantisation function described above to the wavelet coefficients of a textured image necessarily causes some distortion of this image if reconstruction is performed. Measurement of this distortion is possible using the peak signal to noise ratio (PSNR) which compares the original image to the distorted version. The PNSR of two images is defined as

$$PSNR = 20 \log_{10}(r/\varepsilon) \tag{4.18}$$

Quant. Levels	Uniform ($\delta = \infty$)	Logarithmic					
		$\delta = 2$	$\delta = 1$	$\delta = 0.25$	$\delta = 0.05$	$\delta = 0.01$	$\delta = 0.001$
32	41.71	43.07	43.98	46.29	47.33	48.04	48.25
16	35.82	37.20	38.18	40.80	42.01	42.80	42.96
8	29.82	31.05	31.91	34.38	35.55	35.98	36.01
4	23.46	24.59	25.35	27.18	27.58	27.41	27.39

Table 4.1: PSNR (in dB) of logarithmic compared to uniform quantisation for various quantisation levels and values of δ.

where ε is the RMS error between the two images, and r is the dynamic range of the image. A number of texture samples were quantised using the proposed technique, and the images reconstructed from the resulting coefficients. Since the sign of the coefficients is lost during the quantisation process, this information is reinstated before calculating the inverse DWT. The averaged PSNR over these images for a number of levels and values of δ is shown in table 4.1, compared with that for uniform quantisation performed using the same number of levels. It can be seen from these results that the logarithmic quantisation presented here provides a significantly better representation of the wavelet coefficients, with the PSNR increasing for lower values of δ. Figure 4.5, calculated using a typical image containing natural texture, better illustrates this relationship for a number of different quantisation levels, and indicates that a value of $\delta \approx 0.001$ gives the optimal representation of the sample textures. It must be noted that PSNR is not always an accurate measure of perceived image quality, and that this value of δ may not be optimal for use with all images or in all texture classification tasks.

4.4.2 Log-Squared Energy and Mean Deviation Signatures

The previous section has shown that logarithmic quantisation provides a more detailed representation of the wavelet coefficients of texture images, and leads to reduced distortion compared to the uniform case. The non-linearity defined

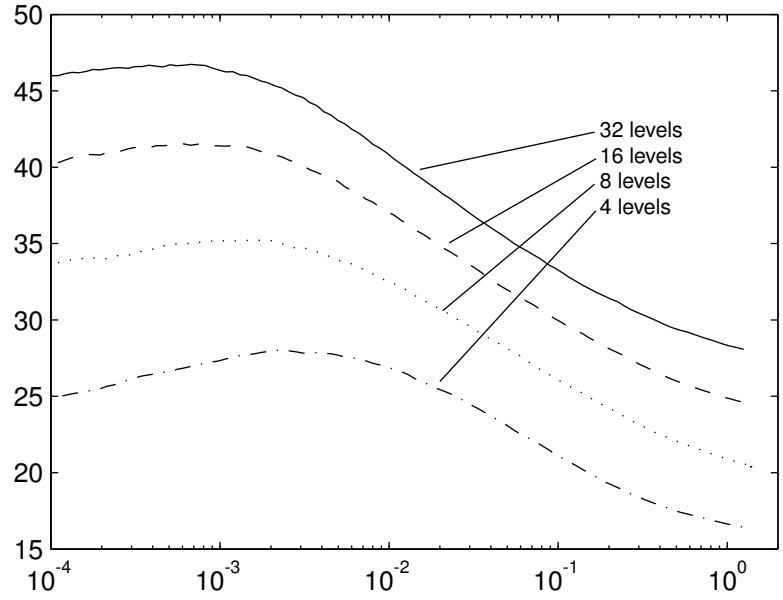


Figure 4.5: PSNR(dB) vs δ for 4, 8, 16 and 32 quantisation levels.

in (4.16) can also be used to modify the wavelet coefficients in order to calculate other statistical features. A common existing texture feature is the variance of the wavelet coefficients, often known as the energy, defined as

$$E_{jl} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} D_{jl}(m,n)^2 \tag{4.19}$$

where D_{jl} is the detail coefficient image l at resolution level j of size (M, N). For the purposes of our evaluation, these features shall be known as the *wavelet energy signatures*. If the magnitude function is used as a rectifying function, (4.19) becomes

$$MD_{jl} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} |D_{jl}(m,n)| \tag{4.20}$$

which represents the mean deviation of the coefficients. For our experiments, we shall call these the *wavelet MD signatures*.

The novel feature sets proposed in this section are calculated by applying the non-linearities of (4.16) and (4.17) to the wavelet coefficients, and then calculating the first and second order moments of the resulting distribution. The scaling constant κ is omitted from this calculation as it is identical for all samples and

thus has no effect on classification performance. These features, which we shall call the *wavelet log squared energy signatures* or LSE signatures, and the *log mean deviation signatures* or LMD signatures, are thus defined as

$$LSE_{jl} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\log\left(\frac{D_{jl}(n,m)^2}{A_j^2\delta}+1\right) \tag{4.21}$$

$$LMD_{jl} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\log\left(\frac{|D_{jl}(n,m)|}{A_j\delta}+1\right) \tag{4.22}$$

where A_j is the chosen saturation point of resolution level j. While the choice of this parameter is arbitrary and has no effect on classification performance, a good approximation of to the true maximum value will give more meaning to the parameter δ. In our algorithm, A_j was determined experimentally by studying the distributions of wavelet coefficients for a large number of textures.

LSE and LMD texture features are extracted at a total of four resolution levels, giving 24 total features. Computationally, these features are efficient to extract, with a negligible increase in complexity when compared to the standard wavelet energy signatures or mean deviation signatures. The performance of such features compared to the wavelet energy signatures and mean deviation signatures is presented in section 4.5.

4.4.3 Wavelet Log Co-occurrence Signatures

The wavelet co-occurrence signatures proposed in [83] are extracted from a co-occurrence matrix constructed using uniform quantisation of the original wavelet coefficients. In this section, a new feature set is proposed utilising the logarithmic quantisation function developed in this chapter. Three variations on this set are proposed for evaluation, with a different non-linearity applied before quantisation in each case. Thus, a *wavelet log co-occurrence matrix* is defined as

$$P_{(k,l,d,\theta)}(i,j) = \frac{|\{((r,s),(t,v)) : q_k(D_{kl}) = i, q(D_{kl}) = j\}|}{MN} \tag{4.23}$$

where j is the resolution level of the wavelet transform, $l = (1, 2, 3)$ is the index of the detail image at that level, d and θ are the distance and angle respectively between the two pixels, and $q_k(x)$ is one of three quantisation functions

$$q_1(x) = \begin{cases} \text{round}\left[\kappa \log\left(\frac{x}{A_j\delta} + 1\right)\right], & x >= 0 \\ -\text{round}\left[\kappa \log\left(\frac{|x|}{A_j\delta} + 1\right)\right], & x < 0 \end{cases} \tag{4.24}$$

$$q_2(x) = \text{round}\left[\kappa \log\left(\frac{|x|}{A_j\delta} + 1\right)\right] \tag{4.25}$$

$$q_3(x) = \text{round}\left[\kappa \log\left(\frac{x^2}{A_j^2\delta} + 1\right)\right] \tag{4.26}$$

where round(x) represents rounding to the nearest integer value. $q_1(x)$ quantises the wavelet coefficients with no rectifying function, and thus requires twice as many quantisation levels to allow for negative values. $q_2(x)$ and $q_3(x)$ use the magnitude and squaring operators respectively to rectify the coefficients before quantisation.

From such matrices, the following co-occurrence features are extracted from table 3.2: energy, entropy, inertia, local homogeneity, max. probability, cluster shade, cluster prominence and information measure of correlation. To keep the number of features manageable, the value of d is restricted to 1, and the matrices for the four directions $\{0°, 45°, 90°, 135°\}$ are averaged to form a single matrix for each detail image D_{jl}. This means that a total of 24 features, are extracted at each resolution level of the wavelet transform. For four decomposition levels, this leads to a total of 96 features. These features will be referred to as the wavelet log co-occurrence (WLC$_k$) signatures, with $k \in \{1, 2, 3\}$ indicating the quantisation function used.

4.5 Experimental Setup and Results

The performance log squared wavelet energy, mean deviation and co-occurrence features is evaluated experimentally using a selection of 25 texture images from the Brodatz album [125], which can be downloaded from

http://www.ux.his.no/~tranden/brodatz.html. Figure 4.6 shows a sample of each texture class used, along with the original plate number from the Brodatz album. These images were chosen on the basis of being relatively uniform in appearance and either non-directional or unidirectional in nature, since none of the features tested here are rotation invariant. Additionally, the classification performance using the wavelet energy signatures on this set of images is poor enough to allow for a meaningful comparison between the techniques. Each of the images was divided into two equal sized sections, one used for extracting training features, the other as a test set. From each of these regions, five sets of training and testing images, each consisting of 100 64×64 pixel samples, are randomly extracted. Classification is then performed using each of these five independent training and testing sets in order to evaluate the consistency of the results.

4.5.1 First-order Statistical Features

In the first set of experiments, the first-order features, namely the log-squared energy and log mean deviation signatures, are extracted from these samples using a number of values of δ. The wavelet energy signatures and mean deviation features are also extracted for comparison purposes. These features are a useful basis for making a meaningful comparison, as they are often used in the literature for this purpose and can thus be used to provide some measure of the performance of the proposed features against the many other texture features in existence. As the dimensionality and size of these feature sets is small, classification is performed using a k-nn classifier for various values of k, using the euclidean distance measure. In order that no feature or features dominate this distance, the mean and variance of the features are normalised to 0 and 1 respectively prior to classification, based on the estimated values from the training data. Each vector from the test set is then classified, and the resulting error rates for each of the five sets reported in tables 4.2 and 4.3. In all cases, the value of k which gave the best overall results is used.

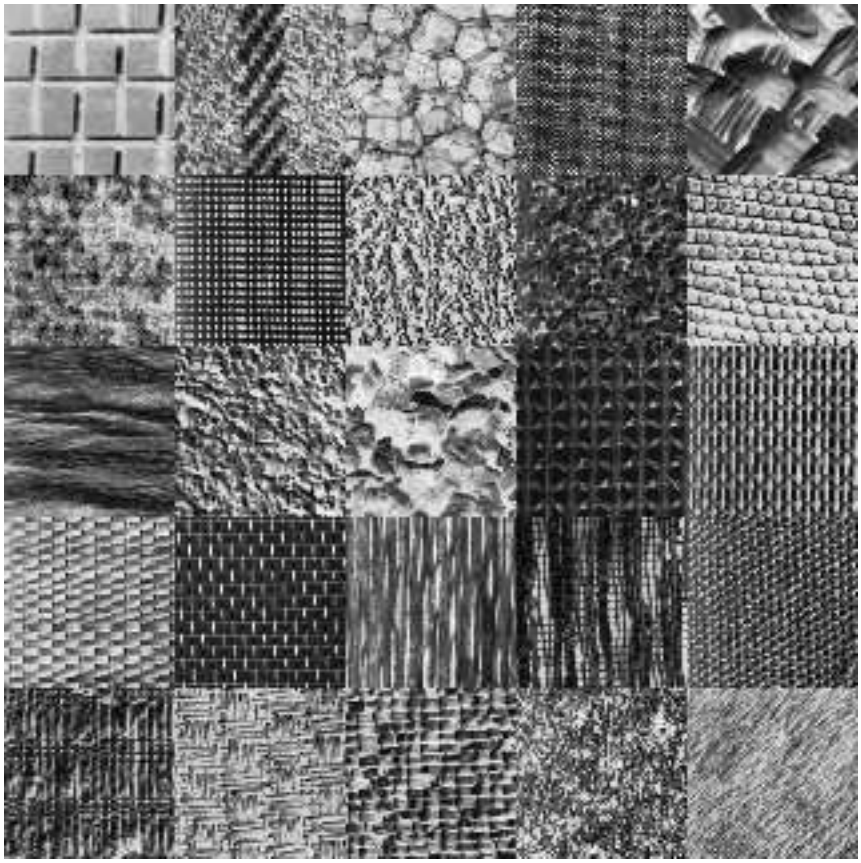


Figure 4.6: Texture images used in experiments. Original plate numbers from top-bottom, left-right: D1, D11, D112, D16, D18, D19, D21, D24, D29, D3, D37, D4, D5, D52, D53, D55, D6, D68, D76, D77, D80, D82, D84, D9, D93.

From these results, it can be seen that the proposed LSE and LMD features significantly improve performance when compared to the standard wavelet energy signatures and mean deviation features, with an average reduction in classification error of approximately 25%. For this selection of images, a value of $\delta = 0.001$ is shown to be near-optimal for the LMD features, while $\delta = 0.0001$ shows better results for the LSE features. This discrepancy is likely due to the influence of the parameter A_j when compared to the actual distribution of wavelet coefficient values.

By evaluating the performance of each feature over multiple training and test sets, a degree of confidence in these results is achieved. From table 4.2, it can be seen that the standard deviation of the overall error rates for the wavelet energy

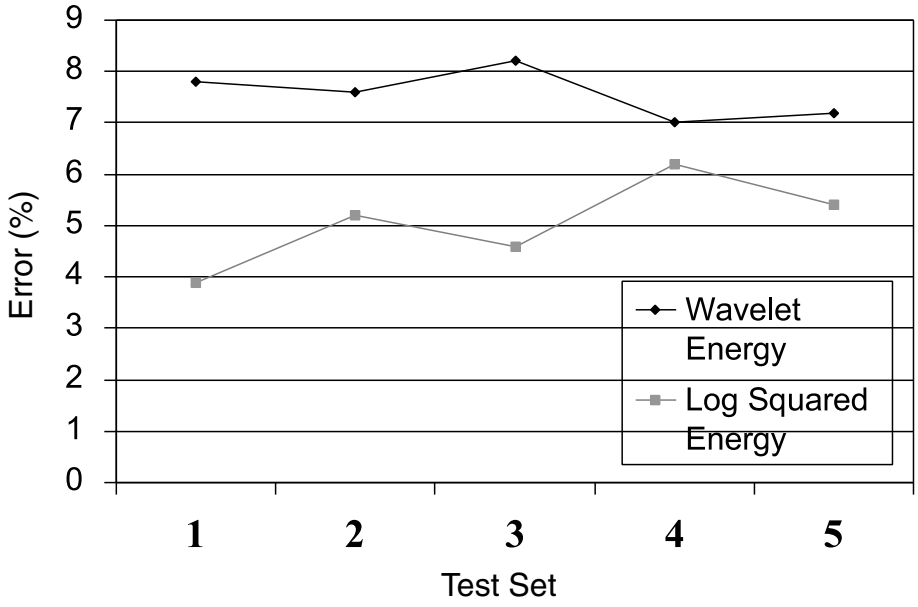


Figure 4.7: Graph of classification errors for the wavelet energy and wavelet log squared energy features ($\delta = 0.0001$) for each of the five test sets.

signatures and log-squared energy signatures (at $\delta = 0.0001$) are 0.43 and 0.77 respectively, with values of 0.40 and 0.71 for the mean deviation and log mean deviation (at $\delta = 0.001$). Even with a small sample size of five independent tests, these values show with a high degree of confidence that the features generated using the proposed transform perform considerably better on this selection of images. This can be visually seen from the error graph of figure 4.7, which shows the comparative errors with and without the logarithmic transform for each of the test sets. Similar results were also obtained for the wavelet log mean deviation features when compared to those extracted without the logarithmic transformation.

In addition to the improvement in overall error rates, the average classification error for each individual texture class was higher for only one of the tested textures for the proposed , further evidence showing that the proposed transform is beneficial when analysing a wide range of natural textures. Table 4.4 shows the error rates of each texture class for all of the tested feature sets in one of the

Test Set	Wavelet Energy	Log-Squared Energy				
		$\delta = 1$	$\delta = 0.1$	$\delta = 0.01$	$\delta = 0.001$	$\delta = 0.0001$
1	7.8%	7.3%	6.9%	5.4%	4.0%	3.9%
2	7.6%	7.1%	6.5%	5.9%	5.7%	5.2%
3	8.2%	8.0%	7.2%	6.7%	5.5%	4.6%
4	7.0%	6.8%	6.8%	6.3%	6.2%	6.2%
5	7.2%	7.0%	6.8%	6.5%	6.1%	5.4%
Avg.	7.6%	7.2%	6.8%	6.2%	5.5%	5.1%

Table 4.2: Classification errors of the wavelet LSE features compared to wavelet energy signatures for each test set and various values of δ.

Test Set	Mean Deviation	Log Mean Deviation				
		$\delta = 1$	$\delta = 0.1$	$\delta = 0.01$	$\delta = 0.001$	$\delta = 0.0001$
1	5.5%	5.2%	4.9%	4.4%	4.0%	4.2%
2	5.6%	5.4%	5.1%	4.8%	4.7%	4.8%
3	6.1%	5.8%	5.6%	5.2%	4.2%	4.4%
4	6.6%	6.1%	5.9%	5.8%	5.8%	5.9%
5	6.2%	5.7%	5.5%	5.5%	5.4%	5.8%
Avg.	6.0%	5.7%	6.1%	5.4%	4.8%	5.0%

Table 4.3: Classification errors of the wavelet MD features compared to wavelet MD signatures for each test set and various values of δ.

experimental sets, again using the best value of k, and $\delta = 0.001$, $\delta = 0.0001$ for the log-squared energy and log mean deviation features respectively.

4.5.2 Second-order Statistical Features

The experiments outlined above were repeated to evaluate the proposed second order statistical features, the wavelet log co-occurrence signatures. These features were extracted from the previously described training and testing samples using each of the three quantisation functions outlined in section 4.4.3. For comparison, the wavelet co-occurrence signatures calculated using uniform quantisation are also extracted. These features have been previously shown to provide excellent classification performance over a wide variety of textured images [83]. A k-nn

Texture Class	Mean Dev.	Wavelet Energy	Log Mean Dev. $\delta = 0.001$	Log-Sq. Energy $\delta = 0.0001$
D1	0%	0%	0%	0%
D11	0%	0%	0%	0%
D112	26%	26%	12%	12%
D16	0%	0%	0%	0%
D18	10%	10%	14%	14%
D19	14%	14%	6%	6%
D21	0%	0%	0%	0%
D24	0%	0%	0%	0%
D29	6%	6%	0%	0%
D3	28%	28%	12%	18%
D37	12%	12%	0%	2%
D4	2%	2%	0%	0%
D5	34%	34%	18%	22%
D52	2%	2%	0%	0%
D53	0%	0%	0%	0%
D55	0%	0%	0%	0%
D6	0%	0%	0%	0%
D68	0%	0%	0%	0%
D76	0%	0%	0%	0%
D77	0%	0%	0%	0%
D80	20%	20%	8%	14%
D82	0%	0%	0%	0%
D84	0%	0%	0%	0%
D9	44%	44%	46%	42%
D93	2%	2%	2%	2%

Table 4.4: Error rates for individual texture classes for wavelet mean deviation, energy, log mean deviation and log squared energy features. In all cases, the best value of k was used.

classifier is again used, and the results shown in table 4.5 for each training and test sample. Again, results for all features are shown for the optimal value of k. For each of the wavelet log co-occurrence signatures, only the near-optimum value of δ is shown. For a better illustration of the effects of δ on overall performance, the reader is referred to figure 4.9.

From these results, it can be seen that all of the WLC signatures outperform the uniformly quantised wavelet co-occurrence signatures, with the WLC_1 fea-

Test Set	**Wavelet Co-oc.**	**WLC$_1$** $\delta = 0.001$	**WLC$_2$** $\delta = 0.001$	WLC$_3$ $\delta = 0.0001$
1	2.7%	1.6%	2.3%	2.0%
2	3.2%	1.3%	2.1%	1.6%
3	3.1%	1.4%	2.6%	1.6%
4	3.4%	1.8%	1.9%	2.4%
5	2.1%	0.8%	1.7%	2.1%
Avg.	2.9%	1.4%	2.1%	1.9%

Table 4.5: Classification errors for wavelet log co-occurrence signatures compared to wavelet co-occurrence signatures extracted with uniform quantisation. The best results for each feature set are shown in bold.

tures showing error rates reduced by approximately 50% overall. The WLC$_2$ and WLC$_3$ signatures, in which the coefficients were rectified using the magnitude and squaring operators respectively, did not perform as well as the non-rectified WLC$_1$ signatures, from which it can be concluded that the sign of the coefficients is of importance when calculating second-order statistics. The relative performance of each of the features is shown in figure 4.8, which clearly illustrates the improved performance of the proposed features, with the WLC$_1$ features providing the lowest error rates for each of the test sets.

The individual error rates for each texture are listed in table 4.6. Interestingly, however, WLC$_2$ and WLC$_3$ features performed considerably *better* on the D9 texture, which contributed the most to the overall error in all cases.

As was the case with the wavelet log energy signatures, a lower value of δ was found to perform the best for the WLC$_3$ signatures. Figure 4.9 shows the relationship between classifier performance and δ for each of the WLC signatures. From this plot it can be seen that $\delta = 0.001$ is near-optimal for both the WLC$_1$ and WLC$_2$ features, with $\delta = 0.0001$ providing better results for the WLC$_3$ feature set.

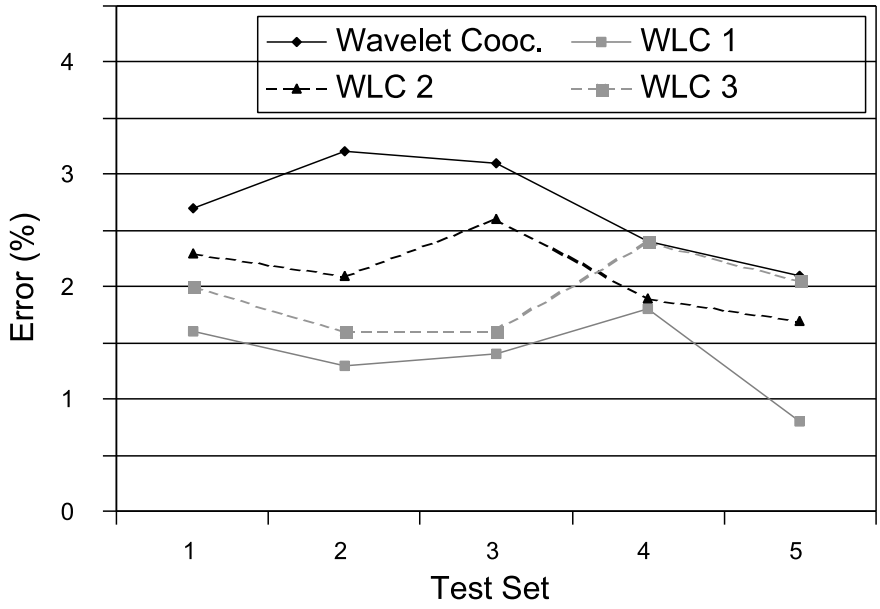


Figure 4.8: Graph of classification errors for the wavelet co-occurrence and wavelet log co-occurrence features for each of the five test sets. The values of δ used were 0.001, 0.001 and 0.0001 for the WLC_1, WLC_2 and WLC_3 features respectively.

4.5.3 Validation of Results

In order to ascertain the relative performance of the proposed feature sets in a variety of conditions, the experiments outlined above were repeated using two further sets of texture images. The first of these, designed to represent a relatively simple texture classification task, were obtained by taking a random selection of 20 textured images from those published at http://astronomy.swin.edu.au/~pbourke/texture/ (obtained 08/2003). An example of samples obtained from each of these images is shown in figure 4.10. The second validation set was taken from the Vistex collection of texture images, which has been widely used in many texture classification experiments [126]. This set of images provides a significantly more challenging classification task, due to the greater variation present in many of the images used. A total of 30 images from this database were used to generate the results presented here.

Texture Class	Wavelet Co-oc.	$\mathbf{WLC}_1$ $\delta = 0.001$	$\mathbf{WLC}_2$ $\delta = 0.001$	$\mathbf{WLC}_3$ $\delta = 0.0001$
D1	0%	0%	0%	0%
D11	0%	0%	4%	0%
D112	4%	10%	22%	14%
D16	0%	0%	0%	0%
D18	4%	0%	4%	10%
D19	0%	0%	0%	0%
D21	0%	0%	0%	0%
D24	0%	0%	10%	0%
D29	0%	0%	0%	0%
D3	4%	0%	0%	0%
D37	0%	0%	0%	0%
D4	0%	0%	0%	0%
D5	12%	0%	2%	6%
D52	0%	0%	0%	0%
D53	0%	0%	0%	0%
D55	0%	0%	0%	0%
D6	0%	0%	0%	0%
D68	0%	0%	0%	0%
D76	0%	0%	0%	0%
D77	0%	0%	0%	0%
D80	8%	6%	4%	6%
D82	0%	0%	0%	0%
D84	0%	0%	0%	0%
D9	38%	22%	10%	12%
D93	2%	0%	0%	2%

Table 4.6: Individual classification error rates for each texture for the wavelet co-occurrence and wavelet log co-occurrence signatures. In each case, the optimum value of k from 4.5 is used.

As in the first set of experiments, each image was divided into two regions of equal size, with training and testing samples taken independently from each region. Due to the smaller size of these images compared to the Brodatz images, only 50 samples are extracted for both training and testing purposes. The same sets of features were then extracted from each such sample, and a k-nn classifier used to evaluate the performance of each feature set. For the purposes of these experiment, the optimal values of δ are used for each of the proposed features, as have been previously empirically determined.

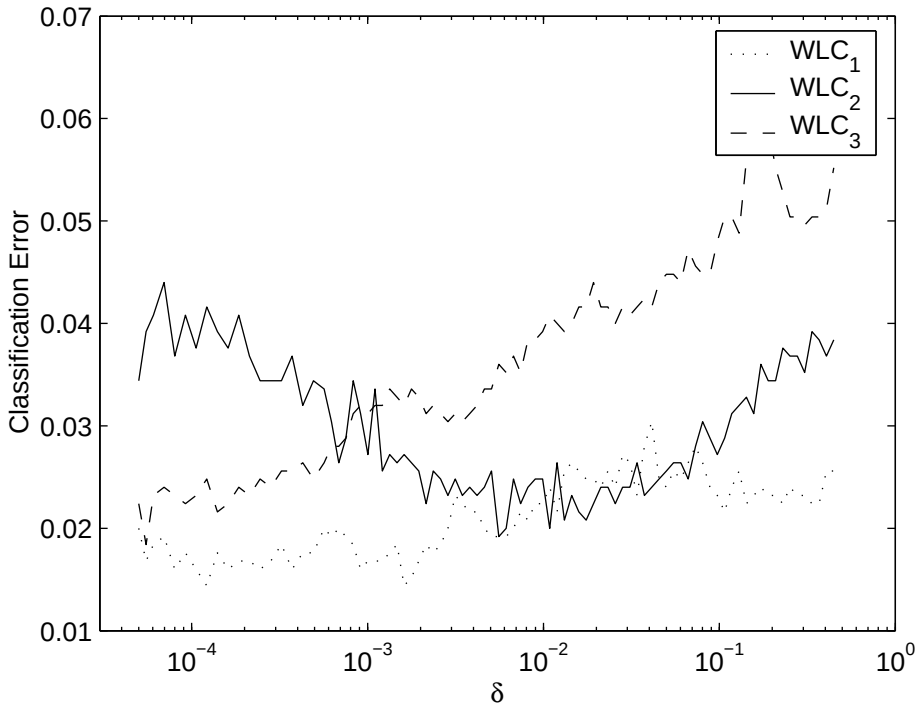


Figure 4.9: Graph of classifier error vs. δ for each of the three WLC signature features.

The results using these sets of images support those already obtained, with a similar reduction in error rates for each of the proposed feature sets when compared to the corresponding linear first and second order statistical features. This is true for both the relatively easy classification task presented by the first validation set, as shown in table 4.7, as well as the more difficult problem of the Vistex images, whose results are shown in table 4.8.

4.6 Chapter Summary

This chapter has presented two general approaches for improving the accuracy of texture classification. A framework for the quantisation of wavelet coefficients on a logarithmic scale was presented which has the potential to provide a better model for such data. Using this approach, a number of quantiser designs are

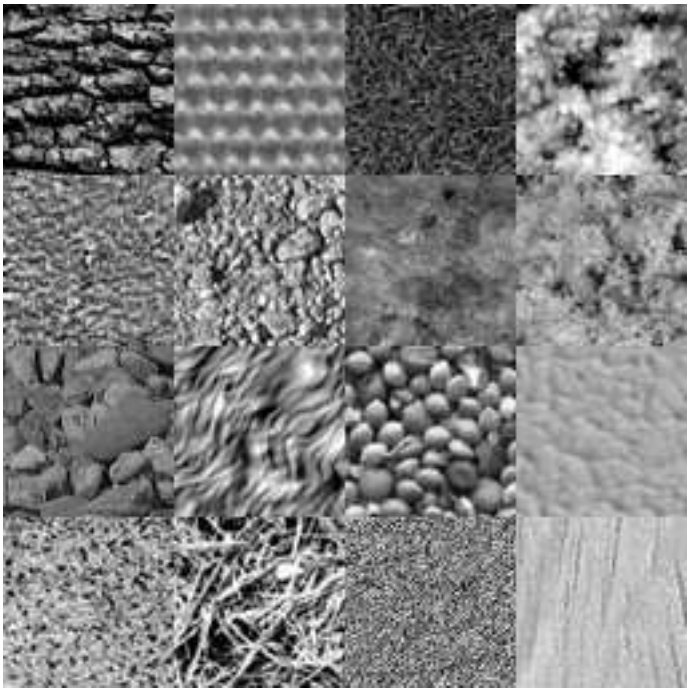


Figure 4.10: Example of texture images used to verify the performance of the proposed texture features.

proposed for use in texture analysis tasks, with experimental evidence showing that the distortion rate of such quantisers measured by the PSNR is significantly less than that produced by uniform quantisation.

Based on this framework, two novel texture features sets capturing first and second order statistics were proposed. The wavelet log-squared energy signatures and log mean deviation signatures have shown to perform better for all texture classes in capturing first order statistical information than the standard wavelet energy signatures used by many authors. For second-order statistical information, the log-squared wavelet co-occurrence signatures proposed here also show an increase in performance over co-occurrence signatures extracted using uniform quantisation.

Experimental evaluation of both of the proposed feature sets was performed using a selection a images from the Brodatz album, from which five sets of independent

Feature Set	Error
Wavelet MD	3.25%
Wavelet Energy	3.5%
Wavelet Log MD	1.5%
Wavelet LSE	1.7%
Wavelet Cooc.	1.0%
WLC_1	0.4%
WLC_2	0.9%
WLC_3	0.3%

Table 4.7: Classification errors for all features obtained using the first set of validation textures.

Feature Set	Error
Wavelet MD	18.2%
Wavelet Energy	21.9%
Wavelet Log MD	11.4%
Wavelet LSE	10.9%
Wavelet Cooc.	7.9%
WLC_1	3.1%
WLC_2	4.5%
WLC_3	3.9%

Table 4.8: Classification errors for all features obtained using a selection of images from the Vistex database.

training and testing samples were extracted. In each such set, the proposed features were shown to significantly outperform the features extracted on a linear scale with a high degree of confidence. The classification error of each individual texture class was lower for the proposed features in all but one case, showing that the technique is applicable to a wide range of textured images. For the purpose of further validation of these results, testing was also performed using two separate sets of textures which provide a significantly easier and more difficult classification challenge, with similar improvement in the overall error rates obtained.

Chapter 5

Texture Feature Reduction and Classification

5.1 Introduction

An important consideration in the problem of texture classification is that of classifier design. Most evaluations of texture features rely on simple classifiers, such as the minimum distance and nearest neighbour classifiers, while other researchers have used artificial neural networks to good effect. Often such classifiers make assumptions regarding the nature of the input features, typically regarding the correlations between features. In practice such assumptions are not generally valid, and may cause a loss of classification accuracy. In addition, many texture analysis methodologies produce feature vectors of large lengths, which are unsuitable for many classifiers. A means of reducing the dimensionality of this data is thus required, either through selection of a subset of features, or by extracting from the vector the principal components of variation. Numerous examples of these techniques are found in the literature, and will be reviewed briefly in this chapter.

An evaluation of a number of popular classifier designs is then conducted in the context of texture classification, with empirical results showing the relative performance of each approach. A new approach for the problem of feature reduction and classification using a combination of linear discriminate analysis and a multi-modal Gaussian mixture model is then proposed, with experimental results showing that improved classification accuracy and computational efficiency is possible using this configuration when compared to commonly used techniques.

5.2 Optimal Feature Spaces for Pattern Recognition

Well known in the field of pattern recognition is the *curse of dimensionality*, whereby increasing the dimensionality of the feature space may actually reduce classification accuracy [127]. The reason for this phenomenon is that, for a given finite number of training observations, it is only possible to adequately populate a feature space of finite dimensions. Thus, for any given feature extraction method with a finite number of training examples N and dimensionality D, there exists an optimal subset of features R for which classification accuracy is maximised. Such an approach is known as *feature selection*.

Alternatively, there exist methods of reducing the dimensionality of the feature space whilst retaining most of the information required to discriminate between the various classes of the problem domain. This is generally accomplished by means of a linear transform which maps the feature space from $D \rightarrow R$ dimensions using a transformation matrix $\mathbf{C}_{R \times D}$. Such processing is commonly known as *feature transformation*.

Both of the these techniques has advantages and disadvantages. Feature selection retains the integrity of each of the features within the set, allowing easier examination and interpretation of classification results. On the other hand, feature

transformation does not retain such integrity, resulting in transformed features which often do not have a specific meaning, and cannot be easily related to those from which they were derived. Another advantage of the feature selection approach is that features which are not selected need not be computed at all for future unknown observations, which in some cases can save considerable computation time. In terms of information retention, feature transformation allows a greater scope for reducing dimensionality whilst retaining the important information. Feature selection can be regarding as a special case of transformation, where the transformation matrix is purely diagonal with 1's on the rows which are to be kept, and 0's otherwise. Because of this inherent limitation, the discriminatory power of a subset derived by feature selection will in most cases be inferior to one of the same dimensionality calculated via a transformation algorithm.

The following sections provide a brief overview of common feature selection algorithms, as well as two of the most commonly used feature transformation techniques, principal component analysis (PCA) and linear discriminate analysis (LDA).

5.2.1 Feature Selection Algorithms

Feature selection can be described as the task of choosing a subset X of a set of features Y, such that the performance of the feature set is maximised in some sense [128]. The metric used to measure the performance of the selected set may be the average inter-class distance as measured by some distance metric, the actual performance using a particular classifier, or even the computational expense in performing classification. This metric can be expressed as a function $J(X)$ of the subset, where a higher value indicates a better subset in respect of this evaluation criteria.

Feature selection is of critical importance in many types of problems, such as:

- Applications where data from multiple sensors is fused into a single feature vector for classification.
- Applications where multiple models are used to generate separate feature sets, and combined to form a single vector.
- Feature vectors of high dimensionality, where insufficient training data exists to describe an adequate model in this high dimensional feature space.

There exists in the literature a large number of feature selection algorithms, which can initially be distinguished as either *optimal* or *sub-optimal* in nature. An optimal feature selection algorithm is guaranteed to find the best possible subset X of dimensionality R from the given feature space Y of dimensionality D. The most basic of such approaches is the exhaustive search, which evaluates all of the $\binom{D}{R}$ possible combinations in order to find the global optimum. Narendra and Fukunaga have proposed the branch-and-bound (BB) feature selection algorithm, which can find the optimum subset much more quickly than the exhaustive search [129]. However, this approach requires that the subset evaluation function $J(X)$ be positively monotonic, that is,

$$J(A \cup B) \geq J(A), \quad \forall A, B \subseteq Y \tag{5.1}$$

Due to the curse of dimensionality problem, this requirement is difficult to achieve in practice, and hence the usefulness of this technique is somewhat limited in scope. Additionally, the computational complexity of the branch-and-bound algorithm tends to exponential in the worst case, making it infeasible for applications with high dimensional feature spaces.

Non-optimal feature selection algorithms are not guaranteed to find a global maximum for $J(X)$, but rather continue to search for a local optimum until some termination condition is met. Such algorithms can be divided into to broad categories, *deterministic* and *stochastic*. Deterministic algorithms, given the same data and parameters, will always arrive at the same solution. Conversely, stochas-

tic methods are partially random, thus repeated executions may result in many different subsets.

A typical example of a stochastic feature selection technique is the so-called Las Vegas feature selection algorithm [130]. Such an approach randomly selects a subset X and compares it to the current 'best' set using the evaluation function $J(X)$, replacing it if better. This step is repeated until either a maximum number of iterations is reached, or a specified threshold for the evaluation function is exceeded. In order to improve the purely random nature of this algorithm, Chen and Liu have proposed an adaptation which weights the probabilities of each feature being selected based on past results [131]. Thus, features which were selected in subsets with high values of $J(X)$ are more likely to be selected in future iterations. Another class of stochastic feature selection techniques are the *genetic algorithms*. By randomly making small modifications, or *mutations* to a group of feature subsets, and pruning those that perform poorly, a near-optimal subset of features can be arrived at in significantly less time than required for an exhaustive search [132, 133, 134]. The advantages of stochastic feature selection methods are that a result can be obtained at any stage of the processing, making them ideal for time-limited applications, and because of their random nature, it is possible that they will find solutions which will never be reached by deterministic approaches.

Two common techniques of deterministic feature selection are the *sequential forward selection* or *sequential backward selection*, which initialise the subset as either an empty set or the entire set, then sequentially add or remove features, maximising the performance function $J(X)$ at each step, until the desired dimensionality is achieved [135]. Since such add and subtract features individually, the optimal *combination* of features is often not found. To overcome this, the so-called (l, r) and generalised (l, r) algorithms were developed, whereby at each iteration l features are included and r removed. The performance of these algorithms, while substantially better than the basic sequential forward and backward

selection approaches, depends heavily on the choice of l and r, and to date no known method exists to predict the values of these parameters which will yield the best feature set.

The deterministic feature selection algorithm which has shown to provide the best results in a number of studies is the floating forward feature selection algorithm proposed by Pudil [128, 136]. This algorithm is more flexible than other techniques, as it allows the parameters l and r to float, rather than being fixed. Consequently, the dimensionality of intermediate stages of the algorithm do not change monotonically, but rather float upwards and downwards, allowing greater scope for finding the globally optimal solution. Additionally, the algorithm does not stop when the desired dimensionality is reached, but rather continues until a further number of features δ have been added. By doing this, it is possible that a better final subset will be reached. A similar method which starts with the full feature set and iteratively removes and adds features is also proposed, known as the floating backwards feature selection algorithm [136]. While both of these algorithms consistently outperform other sequential feature selection techniques, the computational time required is significantly higher for a given problem, as more nodes search tree are required to be traversed. Unlike random techniques, a 'best so far' estimate is not available until the algorithm has finished, or at least reach the desired number of features, making this approach infeasible for applications that require a subset to be selected in a fixed period of time.

Choice of Evaluation Function

The function $J(X)$ is used to evaluate the performance of a feature subset X at each step of the feature selection algorithms described above, and as such, can therefore strongly influence the final result of the algorithm. There are two general methods of forming this function, known as the *filter* and *wrapper* methods.

Using the filter approach, the function $J(X)$ is calculated independently from

the classifier to be used in making class determinations. Such functions include distance metrics, discriminate functions, and other heuristic measures for representing the information value of the feature subset X. The Maholanobis distance is one such commonly used metric. More recently, Liu *et. al.* have proposed a evaluation function called the inconsistency rate, which is monotonic in respect to the size of X [137]. The primary advantages of using the filter approach to feature evaluation are that the resulting feature subset is not dependent upon the choice of classifier used, and should perform equally well regardless of this choice. In general, such functions are also significantly faster to evaluate than wrapper methods.

The wrapper approach to evaluating a feature subset uses direct knowledge of the classifier to determine the evaluation function $J(X)$. The most common wrapper approach is to train a classifier using the selected subset, and use the obtained classification accuracy of a small test set as the evaluation metric. While this approach ensures that the final feature subset is well-matched to the chosen classifier, and thus has the potential for better results, training and evaluating the classifier for each subset can be computationally expensive. In addition, the feature subsets obtained using the wrapper method are necessarily biased towards the particular classifier used, which may be undesirable in some data mining tasks.

To overcome the disadvantages of these approaches, Yuan *et. al.* have proposed a technique which utilises a combination of the two approaches [138]. Initially, the inconsistency metric of [137] is used to quickly reduce the dimensionality of the feature set to a predetermined level. Following this, a neural network wrapper approach is used to adapt this set further for classification.

5.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a widely used technique for reducing the dimensionality of a feature vector by retaining only information related to the

principal modes of variation within the feature space. Because of this approach, PCA is optimal in a mean-squared error sense, and is very useful in applications where data loss is to be kept to a minimum, such as signal compression.

Using PCA, it is possible to transform an observation set $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$ of dimensionality D with zero mean without loss of of information by

$$\mathbf{P}_{(N\times T)} = \mathbf{\Phi}_{(N\times D)}\mathbf{O}_{(D\times T)} \tag{5.2}$$

where $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$ are the N eigenvectors of the covariance matrix of $\mathbf{O}$, calculated by

$$\mathbf{\Sigma_O} = \frac{1}{T-1}\mathbf{O}'\mathbf{O} \tag{5.3}$$

Typically, the number of eigenvectors of the covariance matrix will be equal to the original dimensionality D, resulting in no feature reduction at all. This reduction can be achieved by retaining only the R eigenvectors with the highest corresponding eigenvalues, so that

$$\mathbf{P}_{(R\times T)} \approx \mathbf{\Phi}_{(R\times D)}\mathbf{O}_{(D\times T)} \tag{5.4}$$

where $\mathbf{\Phi}_{(R\times D)}$ are the eigenvectors corresponding to the R largest eigenvalues.

With the exception of special cases where there are zero valued eigenvalues, such a transformation will result in a loss of data, and as such there exists a tradeoff between the dimensionality reduction achieved and the reconstruction error.

There are a number of practical considerations which must be addressed with performing PCA. Calculating the eigenvectors of a covariance matrix of a large dimension is computationally very costly, and in some cases intractable. If there is insufficient training data, the covariance matrix $\mathbf{\Sigma_O}$ may not be of full rank. To address these problems, Roweis has proposed a techniques using the EM algorithm to find only the R largest eigenvalues and eigenvectors of $\mathbf{\Sigma_O}$ [139]. In the majority of practical cases, this algorithm will provide a close approximation to the R largest modes of variation in around 20 to 30 iterations.

5.2.3 Linear Discriminate Analysis

Principal Component Analysis is optimal in the sense of preserving the energy of a given set of feature vectors. No regard is given, however, to ensuring that discrimination between classes within the feature space is maintained. Linear discriminate analysis (LDA) another form of dimensionality reduction which aims to maximise the class separability rather than the energy of the features. By using the within- and between-class scatter matrices, a transform matrix $\mathbf{C}_{(\mathbf{R}\times\mathbf{D})}$ is defined, and used to transform the given observations $\mathbf{O}$ such that

$$\mathbf{P}_{(R\times T)} = \mathbf{C}_{(R\times D)}\mathbf{O}_{(D\times T)} \tag{5.5}$$

The within-class scatter matrices represents the scatter of the observations of each class around their respective means, and can be expressed as

$$\mathbf{S}_w = \sum_{i=1}^{L} c_i \boldsymbol{\Sigma}_i \tag{5.6}$$

where L is the number of classes, c_i is the weighting of class i, and $\boldsymbol{\Sigma}_i$ is the covariance matrix of class i. The between-class scatter matrix is the scatter of the class means around the global mean, given by

$$\mathbf{S}_b = \sum_{i=1}^{L} c_i(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_0)' \tag{5.7}$$

where $\boldsymbol{\mu}_i$ is the mean of class i and $\boldsymbol{\mu}_0$ the global mean, defined as

$$\boldsymbol{\mu}_0 = \sum_{i=1}^{L} c_i \boldsymbol{\mu}_i \tag{5.8}$$

Given these definitions, the aim of LDA is to determine a transform matrix $\boldsymbol{\Phi}$ such that the within scatter is minimised while the between scatter is maximised. One common cost function used to determine this is $tr(\mathbf{C}\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{C}')$, which can be maximised by $\mathbf{C} = \boldsymbol{\Phi}$, where $\boldsymbol{\Phi}$ represents the R greatest eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ [140]. Since $\mathbf{S}_w^{-1}\mathbf{S}_b$ is not guaranteed to be symmetric, normal eigen-decomposition is

impossible, however simultaneous diagonalisation may be used to accomplish this, leading to [140]

$$\mathbf{C}'\mathbf{S}_w^{-1}\mathbf{C} = \mathbf{I} \tag{5.9}$$

$$\mathbf{C}'\mathbf{S}_b\mathbf{C} = \mathbf{\Lambda} \tag{5.10}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The eigenvectors $\mathbf{C}$ obtained from this process are not orthogonal, and as such the transform does not preserve the energy of the features.

While LDA is superior to PCA in maintaining class separability, it has a number of disadvantages. Firstly, the use of a single within-class scatter matrix assumes that each class is defined by the same covariance matrix, which is often not true in many applications. Additionally, the size of the between-class scatter matrix is limited to $\leq L-1$, which puts a similar restriction on the final dimensionality R.

5.3 Classification for Texture Analysis

Classifiers for pattern recognition tasks have been widely studied. In theory, the Bayes classifier is optimal for any problem, as it minimises the probability of error. The true *a posteriori* probability that an observation $\mathbf{o}$ belongs to a class ω_i is given by Bayes rule [140],

$$Pr(\omega_i|\mathbf{o}) = \frac{P(\omega_i)p(\mathbf{o}|\omega_i)}{\sum_{n=1}^{N} P(\omega_n)p(\mathbf{o}|\omega_n)} \tag{5.11}$$

where N is the total number of classes, $P(\omega_i)$ is the *a priori* probability of being in class ω_i, and $p(\mathbf{o}|\omega_i)$ is the true conditional density function for the class ω_i.

In practical applications, this true *a posteriori* probability can never be achieved, since the true conditional density functions $p(\mathbf{o}|\omega_n)$ cannot be known with finite training data in all but the most trivial case. Rather, an estimate of this

probability is given, such that

$$\hat{Pr}(\omega_i|\mathbf{o}) = Pr(\omega_i|\mathbf{o}) + \epsilon(\mathbf{o}) \tag{5.12}$$

where $\epsilon(\mathbf{o})$ is an error term due to the limitations stated above. Given these constraints, the aim of a classifier system is to provide an estimate of $p(\mathbf{o}|\omega_i)$ such that this error is minimised. This typically involves making assumptions about the form of these density functions and as such is very dependant upon the problem itself. The problem of classifier design can thus be interpreted as defining an accurate model for $p(\mathbf{o}|\omega_i)$ in the context of the given classification problem. The following sections describe a number of general approaches to classifier design, and give examples of their use in texture analysis.

5.3.1 Non-parametric Classifiers

Non-parametric classifiers do not make any assumptions about the structure of the underlying probability density functions $p(\mathbf{o}|\omega_i$, but rather use arbitrary distributions to estimate $Pr(\omega_i|\mathbf{o})$. This approach somewhat avoids the problem of not knowing the form of $p(\mathbf{o}|\omega_i)$.

Common implementations of non-parametric classifiers involve Parzen windows or k_n nearest neighbour estimation [140, 141]. The basic premise of such techniques is that if a volume V is placed around the given observation $\mathbf{o}$, and within V exist n training samples of which k are of class ω_i, the density function can be estimated as

$$p(\mathbf{o}|\omega_i) \approx \frac{k/n}{V} \tag{5.13}$$

As $n \to \infty$ and $V \to 0$, this estimate of $p(\mathbf{o}|\omega_i)$ becomes increasingly accurate.

A common implementation of a non-parametric classifier is the *nearest neighbour* classifier. Given a set of training observations $\{\mathbf{o}\}$, such a classifier seeks to find the the closest to the test observation $\mathbf{o}^*$ with respect to a distance metric $D(\mathbf{a}, \mathbf{b})$.

Common choices for $D(\mathbf{a}, \mathbf{b})$ are the simple Euclidean distance,

$$D_E(\mathbf{a}, \mathbf{b}) = (\mathbf{a} - \mathbf{b})'(\mathbf{a} - \mathbf{b}) \tag{5.14}$$

or the Mahalanobis distance, given by

$$D_M(\mathbf{a}, \mathbf{b}) = (\mathbf{a} - \mathbf{b})'\mathbf{W}^{-1}(\mathbf{a} - \mathbf{b}) \tag{5.15}$$

where $\mathbf{W}$ is a weighting matrix representing the relative importance of each feature. The covariance matrix $\mathbf{\Sigma}$ is commonly used for this purpose, although any *a priori* knowledge may be substituted. For small training sets, the choice of distance metric can be crucial in determining the overall performance of the classifier. As the size of the training set becomes infinitely large, $D(\mathbf{a}, \mathbf{b})$ becomes less important, and the classification error of the nearest neighbour classifier has been shown to approach twice the Bayes error [140, 141].

Due to their simplicity, non-parametric classifiers are a popular choice for many applications, and are particularly attractive in situations where limited training data is available, and the creation of a parametric model or adequate discriminate boundaries are infeasible. In addition, no retraining of discriminate boundaries or models is required with the addition of more training data. The disadvantages of this type of classifiers is that they require storage of all training vectors, and must calculate distance measures for each training vector when classifying each sample. This can often lead to large computational and storage requirements when large amounts of training data are used, making non-parametric classifiers unsuitable for some tasks. Non-parametric classifiers have been widely used in the context of texture analysis for both their simplicity and their ability to model distributions with a poorly defined or random structure. The k-nn classifier, in particular, has been widely used for these reasons.

5.3.2 Artificial Neural Networks and Discriminate Classifiers

A *discriminate* classifier is one which functions by generating decision boundaries between the classes which are to be classified, by means of a set of discriminate functions $g_i(\mathbf{o})$. Traditionally, such boundaries are generated by knowledge of $p(\mathbf{o}|\omega_i)$, but in practice the uncertain nature of this probability density means that often the form of the discriminate function $g_i(\mathbf{o})$ is assumed *a priori*.

An artificial neural net (ANN) can be interpreted as a type of discriminate classifier, with the form of the discriminate function modelled on the primitive level processing of the human brain [141, 142]. Although countless variations of this type of classifier have been proposed, perhaps the best known and most widely used is the multi-layer perceptron (MLP). Such a configuration typically consists of three (or more) layers of neurons, namely the input layer, output layer, and one or more hidden layers. Each neuron in input layer is connected via weighting matrix to each neuron in the hidden layer, which are in turn connected via a second weighting matrix to the output layers. In this way, the inputs into the network, which typically correspond to the individual elements of the feature vector to be classified, are converted into one or more binary output signals, generally one for each class. This structure is illustrated in figure 5.1.

The behaviour of each neuron in the network is rudimentary yet powerful in nature. Each consists of a number of weighted inputs which are summed, then undergo a non-linear transform. One common transform use for this purpose is the logistic function, defined as [142]

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5.16}$$

which, as can be seen in figure 5.2, saturates quickly to a binary state of -1 or 1, in a similar fashion to biological neurons. Training of the MLP can be performed in a number of ways, the most common of which is *error back-propagation* [142]. Using this method, training data is input into the network, and depending on

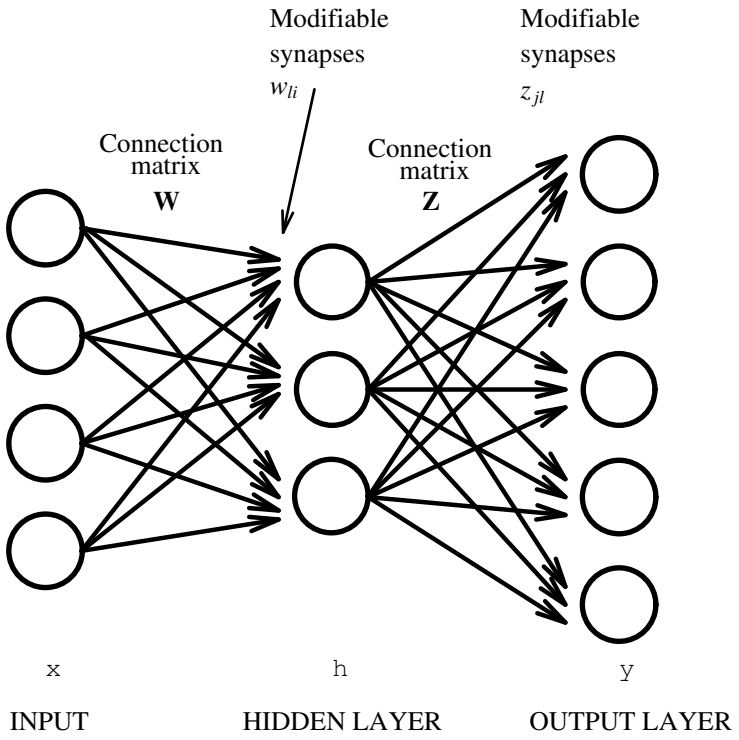


Figure 5.1: Typical structure of a multi-layer perceptron with input nodes, hidden nodes and output nodes connected by weighting matrices.

the error at each output neuron, the weighting matrix **Z** is modified accordingly, given a training constant η which controls the degree of change allowed in the weightings. Since there is no defined output for the hidden layer, the error at these nodes is defined as a function of the errors at the following level, that is, the output nodes in the first instance. In this way, the weighting matrix **W** can be modified in a similar manner. For network architectures with more than a single hidden layer, this process is repeated for all layers. For a more detailed explanation of the training of MLP's, the reader is referred to [142].

When provided with sufficient neurons and training data, a MLP can approximate virtually any function with almost any degree of accuracy [140]. Such a classifier is useful when the form of the density function is unknown, and there is sufficient training data to adequately define the network. MLP's have been frequently used in texture classification algorithms, with examples cited in [143, 144, 145, 146,

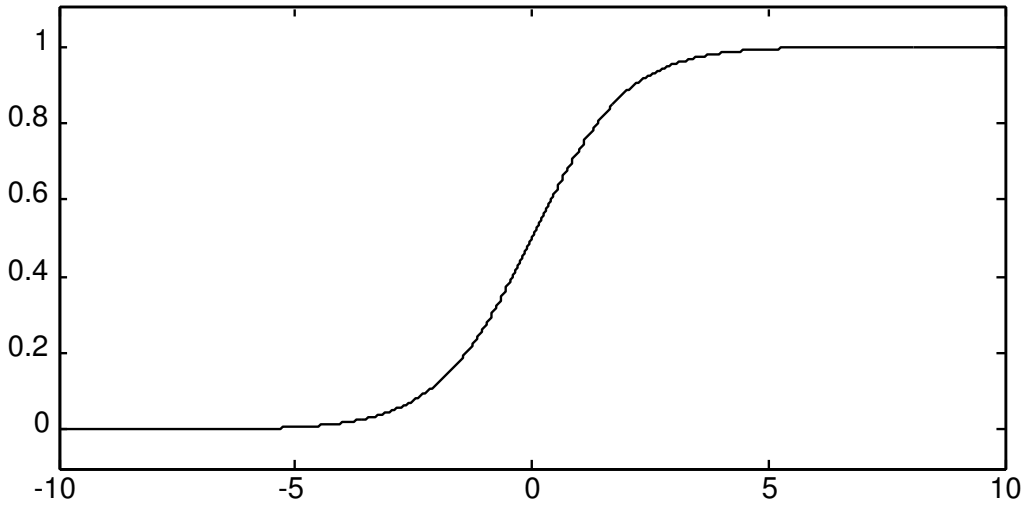


Figure 5.2: The logistic function commonly used in neurons of ANN's.

147].

Another type of discriminant classifier is the support vector machine (SVM), which are limited to two classes. This type of classifier functions by defining a class boundary using the so-called *support vectors*, which map the input to a higher-dimensional space using non-linear functions [148]. The advantage of SVMs is that the complexity of the classifier is defined not by the dimensionality of the transformed space, but rather by the number of support vectors used to define the boundary hyperplane. Because of this, these classifiers are often less prone to overtraining than some other methods. The training of SVMs is a non-trivial problem and is an active topic of research [149, 150, 151]. Such training is often time consuming and this, combined with the fact that a separate SVM must be trained for each class, can make this classifier unsuitable for many applications. Recently, SVMs have used with some success in the field of texture analysis [152, 153, 154, 155].

Discriminative classifier are most useful when there is a small, fixed number of classes with well defined class boundaries. If a binary decision is required, SVMs are a good choice of classifier as they can accurately model decision boundaries of

complex probability densities with sufficient training data. The major disadvantage of this type of classifier is their inter-class dependence. In order to formulate the decision boundary for a a class ω_i, it is necessary to use the data from all other classes. In large applications with a high number of classes, this limitation can mean that training times for the classifier become very large. Such classifiers also require retraining and, in the case of MLPs, redesigning, whenever classes are added or removed from the problem domain, in order to recalculate the discriminate functions $g_i(\mathbf{o})$. This makes discriminate classifiers a poor choice for applications where the number of classes is constantly varying, such as biometric authentication or recognition tasks. Due to the architecture of the classifiers, it is also impossible to easily modify the decision boundaries to account for changes in environmental conditions.

5.3.3 Parametric Classifiers

From Bayes rule 5.11, it can be seen that the problem of classifier design is essentially that of estimating the probabilities $P(\omega_i)$ and the probability densities $p(\mathbf{o}|\omega_i)$. Given a sufficient amount of training data, the estimation of $P(\omega_i)$ is trivial. To accurately model $p(\mathbf{o}|\omega_i)$, however, vast amounts of training data would be required. Even if such data is available, accurately describing the form of these distributions for anything other than the simplest densities is close to impossible. Parametric classifiers attempt to overcome this limitation by assuming that the conditional probability density functions $p(\mathbf{o}|\omega_i)$ have a known parametric form, and use the available training data to estimate such parameters. Typically, a maximum likelihood (ML) approach is used to perform this estimation [140, 141].

Using a parametric classifier, it is assumed that $p(\mathbf{o}|\omega_i)$ has a known parametric form $\boldsymbol{\lambda}_i$, and there exists a collection fo training observations $\mathcal{S}_i\{\mathbf{o}\},\ i = 1 \ldots N$ which have been independently drawn according the probability densities $p(\mathbf{o}|\omega_i)$.

It is also assumed that the observations in $\mathcal{S}_i$ give no information about $\boldsymbol{\lambda}_j$ where $i \neq j$. Given this independence, class distinctions are removed in order to simplify notation.

Given an estimate of $\boldsymbol{\lambda}$, it is possible to calculate the likelihood this value with respect to the set of training data $\mathcal{S}_i$ as the product of the likelihoods of each observation, giving

$$p(\mathcal{S}|\boldsymbol{\lambda}) = \prod_{r=1}^{R} p(\mathbf{o}_r|\boldsymbol{\lambda}) \tag{5.17}$$

where R is the number of training observations from the given set $\mathcal{S}$. For numerical reasons, it is more common to deal in log-likelihoods, such that

$$l(\boldsymbol{\lambda}) = \sum_{r=1}^{R} = \log p(\mathbf{o}_r|\boldsymbol{\lambda}) \tag{5.18}$$

Given these definitions, the ML estimate $\hat{\boldsymbol{\lambda}}$ is defined as the value which maximises $l(\boldsymbol{\lambda})$. Since the logarithm is a monotonically increasing function, this estimate will be identical to that which maximises (5.17). Thus,

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} l(\boldsymbol{\lambda}) \tag{5.19}$$

Theoretically, this equation can be solved using traditional differential calculus, such that

$$\nabla_{\boldsymbol{\lambda}} l(\boldsymbol{\lambda}) = \mathbf{0} \tag{5.20}$$

In practice, however, a global solution to $\hat{\boldsymbol{\lambda}}$ can only be found in the most trivial of cases. It is usually only possible to find a local optimum, which can be accomplished using the expecation maximisation (EM) algorithm. This algorithm iteratively estimates the likelihood of the training observations, and can be expressed as follows [156],

1. Intitalise $\boldsymbol{\lambda}^{\{0\}}$ to some initial value
2. Expectation: compute log-likelihood $l\left(\boldsymbol{\lambda}^{\{i\}}\right)$
3. Maximisation: $\boldsymbol{\lambda}^{\{i+1\}} = \arg\max_{\boldsymbol{\lambda}} l\left(\boldsymbol{\lambda}^{\{i\}}\right)$

4. Repeat 2-3 until $l\left(\boldsymbol{\lambda}^{\{i\}}\right) - l\left(\boldsymbol{\lambda}^{\{i-1\}}\right) \leq \tau$ or $i \geq N$

where τ is predefined convergence threshold, and N is the maximum allowed number of iterations.

The EM algorithm operates somewhat differently from other optimisation techniques, such as the gradient ascent algorithm, and can often find solutions which would not necessarily be arrived at by other methods. Although the final estimate of $\hat{\boldsymbol{\lambda}}$ is not assured to be the global maximum, the EM algorithm has shown to give a good approximation when used in the context of Gaussian mixture models and hidden Markov models in many practical situations [157].

5.3.4 Gaussian Mixture Models

Gaussian distributions naturally occur in many systems, and have been widely used to model many different phenomena. The reason for this proliferation can be explained by means of the Central Limit Theorem, which states that the aggregate of a large number of random disturbances will approach a Gaussian distribution [141]. In complicated systems, the distributions of variables can sometimes be more accurately modelled by using a *mixture* of Gaussian functions, leading to the development of a classifier known as the Gaussian mixture model (GMM). Using the summation of a number of Gaussian functions, it is possible to approximate a wide range of arbitrarily shaped density functions, with few restraints on the form of such a function. Figure 5.3 shows a one dimensional example of this ability, with a number of Gaussian distributions combining to approximate an irregular density function.

Mathematically, the GMM approximation of a probability density function using M mixtures is given by

$$p(\mathbf{o}|\boldsymbol{\lambda}) = \sum_{i=1}^{M} c_i b_i(\mathbf{o}) \tag{5.21}$$

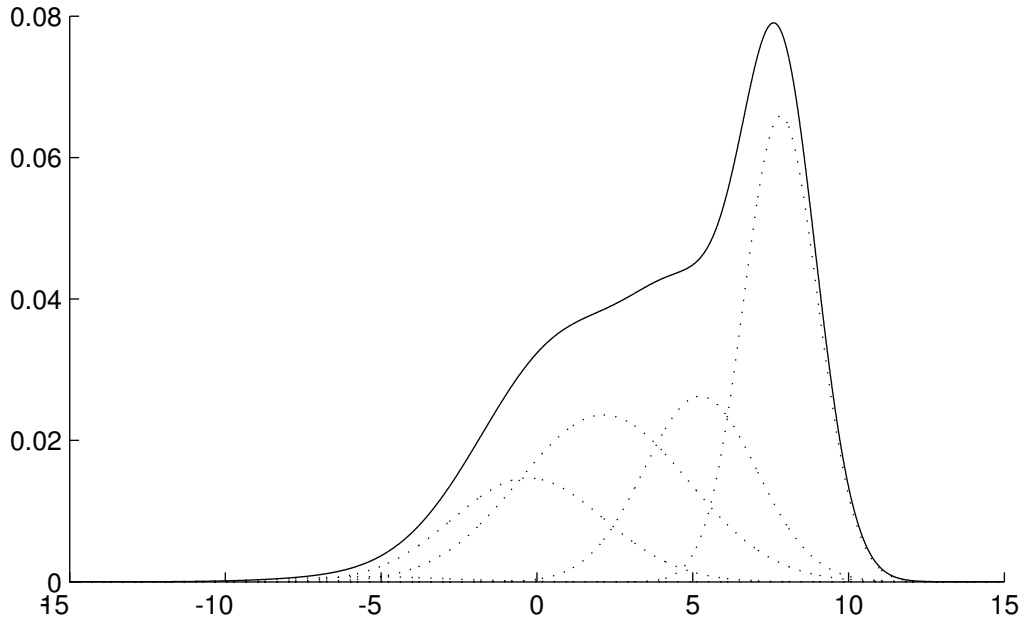


Figure 5.3: Example of GMM, showing the use of 5 Gaussian mixtures (dotted lines) to approximate an arbitrary random density function (solid line).

where $b_i(\mathbf{o})$ is the Gaussian density function of component i, and c_i is the weighting of this function, with $\sum_{i=1}^{M} c_i = 1$. The individual density functions b_i are described as [141]

$$b_i(\mathbf{o}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} e^{\frac{-(\mathbf{o}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{o}-\boldsymbol{\mu}_i)}{2}} \tag{5.22}$$

where D is the dimensionality of the feature vector $\mathbf{o}$, and $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean and covariance of $b_i(\mathbf{o})$ respectively. Given these definitions, it can be seen that the parametric form $\boldsymbol{\lambda}$ of the GMM classifier is definined by c_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, for the given number of mixtures M.

Ideally, using a high number of mixtures M, and fully describing the covariance matrices for all such mixtures will give the best approximation of the desired density function. In the absence of sufficient training data, however, it is not possible to accurately compute a high number of parameters, and the number of degrees of freedom of $\boldsymbol{\lambda}$ must be reduced, either by reducing the number of mixture components M, or making assumptions regarding the form of the covariance matrices $\boldsymbol{\Sigma}_i$. Thus, depending on the nature of the classification task and the amount of training data available, it is necessary to choose an appropriate

GMM topology. Typically, one of three distinct forms of specifying the covariance matrices is used, being [141]

Nodal covariance: each mixture component of every class has its own covariance matrix.

Grand covariance: a single covariance matrix $\mathbf{\Sigma}_i$ is used for *all* mixtures of a given class i.

Global covariance: a single covariance matrix $\mathbf{\Sigma}$ is used for *all* mixture components of *all* classes.

Nodal covariances allow the greatest freedom in describing the form of the density function, and are the most common choice. However, in applications where training data is limited, or *a priori* information regarding the nature of the density functions exists, grand and global covariance matrices may be employed to provide a better approximation of $p(\mathbf{o}|\omega_i)$. Additionally, the covariance matrices may be represented as full or diagonal only matrices. Using the full matrices leads to a more accurate representation, however requires significantly more training data, and is computationally more expensive. For most applications, diagonal nodal covariance matrices are used [141].

The choice of M, the number of mixture components, is also of critical importance, with no simple solution. Typically such a choice is made using knowledge of the problem domain, and empirical and heuristic techniques. Using more mixture components can possibly give a better representation, with more training data required as a consequence.

Estimating the parameters $\hat{c}_i$, $\hat{\boldsymbol{\mu}}_i$ and $\hat{\mathbf{\Sigma}}_i$ of a GMM can be performed using the EM algorithm described above. Using the normalised likelihood

$$L_i(r) = \frac{c_i b_i(\mathbf{o}_r)}{\sum_{k=1}^{M} c_k b_k(\mathbf{o}_r)} \tag{5.23}$$

the following estimates can be made at each iteration of the EM algorithm

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{r=1}^{R} L_i(r)\mathbf{o}_r}{\sum_{r=1}^{R} L_i(r)} \tag{5.24}$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{r=1}^{R} L_i(r)(\mathbf{o}_r - \hat{\boldsymbol{\mu}}_i)(\mathbf{o}_r - \hat{\boldsymbol{\mu}}_i)'}{\sum_{r=1}^{R} L_i(r)} \tag{5.25}$$

$$\hat{c}_i = \frac{1}{R}\sum_{r=1}^{R} L_i(r) \tag{5.26}$$

In practice, the values calculated in these equations will usually converge in a small number (< 20) of iterations.

Although the EM algorithm is guaranteed to converge to a local maximum, this may not be close to the desired global maximum. The quality of the final value of $\hat{\boldsymbol{\lambda}}$ is highly dependant upon the starting estimate $\boldsymbol{\lambda}^{\{0\}}$. Three techniques for this initialisation which have been shown to perform well in practical applications are [157]

Pre-labelled: the training set $\mathcal{S}$ is pre-labelled into mixture components via some *a priori* information regarding the known structure of the training data. This approach requires good knowledge of the problem domain, and cannot compensate for mismatches in the expected and actual environmental conditions. For an application such as texture analysis, where the structure of the feature vectors is highly random in nature, such an approach is infeasible.

Random: M random observations from the training set are chosen as the means, and the identity matrix $\mathbf{I}$ as the covariance, for each mixture component. Although this approach allows for a variable number of components, the performance is highly dependant upon the actual observations chosen as the initial means.

K-means: Using k-means clustering, the initial observations are grouped into M clusters based on some distance metric [158]. Using this clustering technique

guarantees the the clusters have minimum inter-cluster distance based on the chosen metric, and provides a stable and consistent starting point for the EM algorithm.

Experimental evidence suggests that the k-means clustering initialisation techniques usually gives the best overall classifier performance [157], and as such will be used as the basis for all GMMs in this work.

GMMs have been widely used in many areas of pattern recognition and classification, with great success in the area of speaker identification and verification [157, 159]. To date, there has been little use of GMMs in the field of texture analysis. Given the highly random nature of textured images, and the ability of the GMM to accurately describe a wide variety of density functions, it is proposed that using such a classifier will provide an improvement in performance when compared to other commonly used techniques.

5.4 Proposed Classifier Design

An optimal set of features for any pattern recognition task is one whereby the observations of each class are tightly clustered in feature space, whilst the clusters for each class are well separated. Ideally, the individual features should also be uncorrelated and non-redundant. In practice, such properties are rarely obtained by any extracted features, which leads to reduced classifier performance. Specifically, texture features extracted from bands of the wavelet transform are generally highly correlated and contain significant redundancy. To some extent, such deficiencies can be addressed by a linear transformation of the feature space. LDA describes a transformation whereby the discrimination between the classes is maximised, and as such can improve classification accuracy even in cases where feature reduction is not required. Additionally, it has been shown that LDA performs more effectively if PCA is first used to map the initial features to a lower

subspace [160, 161]. The reason for this improvement is that PCA can remove low energy components of the feature space which may effect the ability of LDA to effectively discriminate between the classes.

Due to the wide range of possible textures, the transformed features after applying LDA are in general not ideal. Due to the assumption of equal covariance, the transform densities are often non-Gaussian in nature. Because of this, classifiers which make such assumptions, for example the simple minimum distance classifier which assumes a Gaussian distribution in feature space, have been shown to perform quite poorly in many situations. Non-parametric classifiers can typically perform well in such conditions and have been used extensively for texture classification, with the k nearest-neighbour classifier using either the Euclidean or Mahalanobis distance metric a common choice. While this classifier has the advantage of simplicity, and makes no assumptions regarding the distributions, the theoretical error, shown to be twice the Bayes error for infinite training data, is quite high [140, 141].

Neural networks have also been commonly used to classify textures, with some authors showing improved performance over non-parametric techniques such as the k-nn classifier [143, 146]. Although neural networks can model a wide variety of distributions, they require retraining when new classes are added, and have been shown to model multi-modal distributions poorly in some instances. Support vector machines also suffer this disadvantage, as the support vectors are unable to adequately describe the class boundaries in these cases.

GMMs, because of their ability to approximate a wide range of distributions, are not as susceptible to the form of the probability density function as are other classifiers, and can to some extent overcome the problems with LDA described above. Furthermore, a GMM can, given sufficient mixture components, model a multi-modal distribution as well as a uni-modal case, making it ideal for applications in which significant inter-class variation exists. Additionally, the theoretical error of a well-trained GMM has been shown to approach the Bayes error, mak-

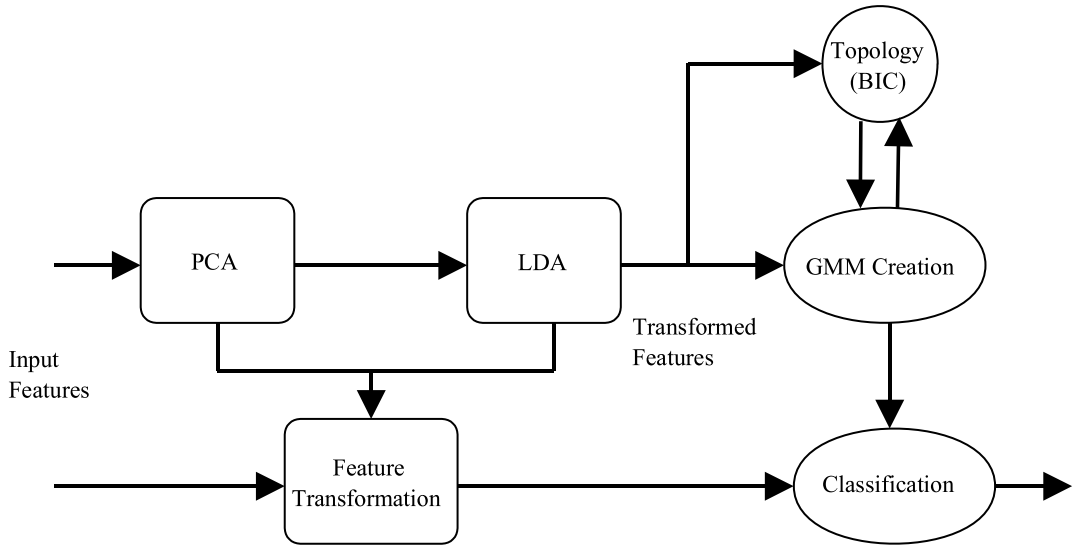


Figure 5.4: Block diagram of the proposed classifier design for texture analysis.

ing it near-optimal in this sense. For these reasons, the GMM is chosen for the classification stage in the proposed system.

Thus, the final classifier design can be described by the block diagram of figure 5.4. The mean of the combined training observations of all classes is calculated and subtracted, following which PCA is performed to map this space to one of lower dimensionality. LDA is then applied to obtain a further reduction, while optimising the discrimination between the classes. This reduced feature set is then used to train a GMM, the order of which is determined heuristically based on the number of training observations N, and the final reduced dimensionality of these observations R.

5.4.1 GMM Form and Topology

When designing a GMM, the parametric form $\boldsymbol{\lambda}$ must be defined. This typically entails choosing the number of mixture components M used to approximate the density function, and the form of the covariance matrices for each mixture. Given that few assumptions can be made regarding the form of the texture density

functions which are to be modelled, imposing the restraints of grand or global covariance will likely significantly degrade performance, and thus nodal covariance is used in all cases. These matrices are diagonal only, however this leads to no loss of accuracy since the LDA previously performed guarantees diagonal covariance matrices for all transformed features.

Having chosen the nature of the covariance matrices, the topology of a GMM classifier is limited to the number of mixtures used. In order to choose this value, the Occam's razor principle has been widely used by many authors. This theory states that a model should be complex enough to be able to capture data specifics, but simple enough for efficient computation. Additionally, using a simpler model is thought to reduce the likelihood of over-fitting the model to the training data, resulting in better generalisation.

Given a set of C classes $\{\omega_i : i = 1, \ldots, C\}$ and a set of L_i candidate models $\{M_{il} : l =, \ldots, L_i\}$, optimising the classifier design can be though of as finding the model M_{il} which maximises some criterion function $\mathcal{C}(\cdot)$. Furthermore, by viewing the model as a union of topology $\mathcal{T}_{il}$ and parametric form $\boldsymbol{\lambda}_{il}$,

Using a Bayesian framework it is possible to maximise the posterior probabilities of a model, given a set of training observations $\mathbf{X}$. If there exist two competing topologies $\mathcal{T}_{il}$ and $\mathcal{T}_{ik}$ of class ω_i, the ratios of their posterior probabilities can be given by Bayes' theorem as

$$B_{lk}^{i} = \frac{P(\mathcal{T}_{il}|\mathbf{X}_i)}{P(\mathcal{T}_{ik}|\mathbf{X}_i)} = \frac{P(\mathbf{X}_i|\mathcal{T}_{il})P(\mathcal{T}_{il})}{P(\mathbf{X}_i|\mathcal{T}_{ik})P(\mathcal{T}_{ik})} \tag{5.27}$$

Assuming that all topologies are equally likely, this can be simplified to

$$B_{lk}^{i} = \frac{P(\mathbf{X}_i|\mathcal{T}_{il})}{P(\mathbf{X}_i|\mathcal{T}_{ik})} \tag{5.28}$$

which is known as the *Bayes factor* [162].

Extending this, the Bayes factor criterion (BFC) is a measure of the performance of a given topology $\mathcal{T}_{il}$ against all competing topologies, calculated by the geo-

metric mean of the individual Bayes factors, ie

$$BFC = \log \left\{ \prod_{k=1,k\neq l}^{L_i} B_{lk}^{i} \right\}^{\frac{1}{L_i-1}} \tag{5.29}$$

$$= \log P(\mathbf{X}_i|\mathcal{T}_{il}) - \frac{\sum_{k=1,k\neq l}^{L_i} \log P(\mathbf{X}_i|\mathcal{T}_{ik})}{L_i - 1} \tag{5.30}$$

From (5.30) it can be seen that the Bayes factor criterion is the difference between two terms, the first of which is known as the *evidence* of the topology, and the second which is the average of the evidence for each competing topology in the same class. Clearly, the topology which has the highest evidence value will also have the highest BFC, so choosing the 'best' model can be viewed as finding the maximum of the function $P(\mathbf{X}_i|\mathcal{T}_{il})$.

Calculating the evidence of a topology can be accomplished by integration over the entire set of parameters $\boldsymbol{\lambda}_{il}$, giving

$$P(\mathbf{X}_i|\mathcal{T}_{il}) = \int p(\mathbf{X}_i|\mathcal{T}_{il}, \boldsymbol{\lambda}_{il}) p(\boldsymbol{\lambda}_{il}|\mathcal{T}_{il}) d\boldsymbol{\lambda}_{il} \tag{5.31}$$

The calculation of this integral is often computationally intractable for a problem of any significant size, and thus can only be evaluated by numerical methods or by an approximation technique. One example of this is the Laplacian approximation, which assumes that the function $p(\mathbf{X}_i|\mathcal{T}_{il}, \boldsymbol{\lambda}_{il}) p(\boldsymbol{\lambda}_{il}|\mathcal{T}_{il})$ is strongly peaked around the most likely parameter set $\boldsymbol{\lambda}_{MP}$ [163, 164]. Under this assumption, the evidence can then be approximated by a Taylor expansion around this peak value, which can be shown to give a final value of the Bayes information criterion (BIC) of

$$BIC(\mathcal{T}_{il}) = \log P(\mathbf{X}_i|\mathcal{T}_{il}) \tag{5.32}$$

$$= \log p(\mathbf{X}_i|\mathcal{T}_{il}, \hat{\boldsymbol{\lambda}}_{il}) - \frac{K_{il}}{2} \log N_i \tag{5.33}$$

where $\hat{\boldsymbol{\lambda}}_{il}$ is the maximum likelihood estimate of the parameters of the chosen model, K_{il} is the number of free parameters in the model, and N_i is the number of training observations in $\mathbf{X}_i$. This equation can be viewed as the likelihood of the given model minus the penalty factor $\frac{K_{il}}{2} \log N_i$, which increases linearly

with the number of free parameters in the model. Because of the error in the probability estimates, a normalising parameter $\alpha > 0$ is usually also introduced, given the final form of the BIC as

$$BIC(\mathcal{T}_{il}) = \log p(\mathbf{X}_i|\mathcal{T}_{il}, \hat{\boldsymbol{\lambda}}_{il}) - \alpha\frac{K_{il}}{2}\log N_i \tag{5.34}$$

When using (5.34) for the purposes of calculating the optimal number of mixtures for a GMM using diagonal only nodal covariance matrices, K_{il} is easily determined as $D \times M_{il}$ where D is the dimensionality of the training observations and M_{il} is the order of the topology being tested. Using this technique is computationally feasible for a moderate number of candidate mixtures, which in the proposed system is limited to 50 or less.

5.5 Experimental Setup and Results

The combination of LDA feature reduction and the GMM classifier proposed in this chapter is experimentally evaluated using the selection of texture images from the Brodatz album shown in figure 4.6. Once again, each image is divided into two halves, and five independent training and testing sets, each of 100 64×64 samples, are extracted from each. Various features are then extracted from each such sample, and used to train a number of classifier designs. For the purposes of comparison, a non-parametric classifier (k-nn) and a multi-layer perceptron implementation of an ANN are used, as well as the proposed classifier design. The Bayes information criterion proposed in section 5.4.1 was used to determine the number of mixture components for the GMM for each class. For the k-nn classifier, classification was performed using many values of k, and the best results reported. A similar process was also used when determining the accuracy of the MLP classifier, whereby a number of different hidden nodes were tested, and those providing the best results used. Although these methods of parameter selection are not possible for practical classification tasks, they serve to optimise the results

of the k-nn and MLP classifiers, thus providing a valid measure of comparison with the proposed design.

Each of the classifiers is evaluated using texture features of both low and high dimensionality, in order to evaluate the performance of the proposed design in a number of realistic situations, and to highlight the need for feature reduction techniques when the total number of features is excessive. The amount of available training data is also varied in order to determine the suitability of the various classifiers in such environments.

5.5.1 Low Dimensionality Feature Spaces

Two good examples of low dimensional sets of texture features are the wavelet energy and mean deviation features. When calculated to N levels of wavelet decomposition, the total number of features generated for each sample is $3N$, in our case 12.

Using each of these features, each classifier was trained using the previously extracted training samples, and used to classify each set of test cases. Tables 5.1 and 5.2 show the classification results of this experiment for each classifier design when no feature reduction is performed. It can be seen from these results that there is very little difference in classifier performance using such features, with the average classification rates almost identical. This similarity in performance is likely due to the low dimensionality of the feature space used, which makes it possible for all of the classifiers to create an adequate model of the training data.

In the next experiment, the performance of each of the classifiers was tested with the training data undergoing linear discriminate analysis prior to training. No actual reduction in dimensionality is required, however the form of the features is modified such that an optimal basis for linear separation is established. Tables 5.3 and 5.4 show the results of these tests, once again with the optimal values of k

Test Set	k-nn	MLP	GMM
1	7.8%	7.8%	7.7%
2	7.6%	7.4%	7.7%
3	8.2%	8.1%	8.1%
4	7.0%	7.3%	7.2%
5	7.2%	7.2%	7.0%
Avg.	7.6%	7.6%	7.4%

Table 5.1: Results of texture classification using the k-nn, MLP and GMM classifiers with the wavelet energy features, with no feature reduction performed prior to classification.

Test Set	k-nn	MLP	GMM
1	5.5%	5.4%	5.6%
2	5.6%	5.5%	5.6%
3	6.1%	6.3%	6.3%
4	6.6%	6.5%	6.7%
5	6.2%	6.4%	6.3%
Avg.	6.0%	6.0%	6.1%

Table 5.2: Results of texture classification using the k-nn, MLP and GMM classifiers with the wavelet mean deviation features, with no feature reduction performed prior to classification.

and H used for the k-nn and MLP classifiers respectively. From these results, it can be seen that performing LDA significantly improves the performance of the GMM classifier, while the errors of the other two tested classifiers is reduced only slightly, and in a few cases increased.

This significant improvement in the results of the GMM classifier can be somewhat explained by the fact that LDA converts the training data into a form which is similar to the parametric form of the GMM, with diagonal covariance matrices and no redundancy of information, allowing the GMM to create a more accurate model of the transformed data. Another result of this is that less mixtures are required to adequately describe the probability densities of each of the classes, allowing for a more robust and generalised classifier design less prone to overtraining. This conclusion is supported by the numbers of mixtures used by the

Test Set	k-nn	MLP	GMM
1	7.5%	7.9%	5.9%
2	7.5%	7.5%	6.1%
3	8.1%	8.0%	6.2%
4	6.7%	7.5%	5.5%
5	7.0%	7.3%	5.7%
Avg.	7.4%	7.7%	5.9%

Table 5.3: Results of texture classification using the k-nn, MLP and GMM classifiers with the wavelet energy features when LDA is applied before classification.

Test Set	k-nn	MLP	GMM
1	5.1%	5.2%	3.6%
2	5.5%	5.4%	4.1%
3	6.0%	6.3%	4.4%
4	6.5%	6.4%	4.9%
5	6.0%	6.2%	4.4%
Avg.	5.8%	5.9%	4.3%

Table 5.4: Results of texture classification using the k-nn, MLP and GMM classifiers with the wavelet mean deviation features when LDA is applied before classification.

GMM classifier, as determined using the BIC metric. When LDA is performed prior to training, the calculated number of mixtures is on average approximately 30% lower compared to the raw features.

5.5.2 High Dimensionality Feature Spaces

In order to evaluate the effectiveness of the feature selection and reduction techniques, a feature set of large dimensionality is required, such that for the given amount of training data it is not possible to adequately train the selected classifier. To simulate this environment, the WLC_1 feature set proposed in chapter 4 containing a total of 96 features is used, with the number of training observations limited to only 50. This experimental setup is achieved by randomly discarding half of the training samples from each of the 25 images, while retaining all 100

test cases. Reducing the amount of available training data has a significant effect on the classification error rates, as can be seen by the increase in average error from 1.4% to 5.1% when using the k-nn classifier with no feature reduction.

Within this experimental framework, the performance of the proposed LDA feature reduction is compared to that of Pudil's floating forward feature selection algorithm, which in a number of evaluations has shown to outperform other feature selection techniques. The wrapper method of subset evaluation used so that the final subset selected by this method is optimised for each classifier without being biased by the choice of filter function. Once again, the k-nn, MLP and GMM are used to perform final classification, with the results shown in table 5.5. When using Pudil's algorithm, the number of features providing the best classification accuracy is used, with this value shown in parentheses. To provide an estimate of base performance, the classification errors when no feature selection is performed are also included in these results.

From these results it can be seen that the proposed method of dimensionality reduction significantly outperforms Pudil's feature selection algorithm in all cases, with the best results obtained when using the GMM classifier. The results obtained when no feature reduction was performed illustrates well the curse of dimensionality phenomena that exists when training data is limited, with very large error rates for all classifiers when compared to the results shown for large training sets in 4.5.

5.6 Chapter Summary

This chapter has presented a review of classifier theory and feature reduction techniques, with particular attention to their application in the field of texture analysis, and presented a combination of linear discriminate analysis and Gaussian mixture model classifier which is experimentally shown to perform well in a

Feature Reduction	Test Set	k-nn	MLP	GMM
Full Set	1	4.9%	5.7%	5.9%
	2	4.2%	5.4%	5.3%
	3	5.3%	6.2%	6.0%
	4	7.2%	9.8%	8.1%
	5	4.8%	6.5%	7.1%
	Avg.	**5.3%**	**6.7%**	**6.5%**
Pudil's FFFS	1	3.6%	3.7%	3.4%
	2	3.5%	3.5%	3.7%
	3	4.8%	4.6%	3.6%
	4	5.2%	5.4%	5.1%
	5	4.2%	4.1%	4.0%
	Avg.	**4.3%**	**4.3%**	**4.1%**
PCA/LDA	1	3.2%	3.1%	2.2%
	2	3.3%	3.1%	2.4%
	3	3.9%	3.6%	3.1%
	4	4.1%	3.9%	3.5%
	5	3.5%	3.1%	2.9%
	Avg.	**3.6%**	**3.4%**	**2.8%**

Table 5.5: Classification errors for the WLC_1 feature set using limited training data and various methods of feature reduction.

texture classification context. The review of feature reduction has given a background on stochastic and deterministic methods of feature selection including the floating forward feature selection algorithm proposed by Pudil, as well as a summary of the application of PCA and LDA to the problem of dimensionality reduction. A brief discussion on Bayesian classifier theory has also been presented, along with examples of the different approaches taken when designing classifiers including nonparametric classifiers, artificial neural networks and support vector machines, and parametric classifiers including the GMM. Numerous applications of each of these types of classifiers were presented, with particular attention to those in the field of texture analysis.

A combination of linear discriminate analysis and a Gaussian mixture model classifier is presented for application to texture classification. Using a fixed number of mixtures to model texture was found to be inadequate, due to the widely vary-

ing nature of natural textures. Because of this, the Bayes information criterion (BIC) was used to dynamically determine the necessary number of mixtures from a predetermined range of possible values.

Experimental results using a selection of images from the Brodatz album show that the proposed classifier design can improve performance in most cases when compared to k-nn and MLP classifiers. These results were consistent when using both low and high dimensionality features sets, and were validated using a number of trials.

Experimentally, the performance of the LDA feature reduction was also found to be superior to the floating forward selection algorithm, which has been previously found to outperform other feature selection techniques.

Chapter 6

Scale Cooccurrence for Texture Analysis

6.1 Introduction

The coefficients of the wavelet transform have been shown to provide an excellent basis for identifying and segmenting textured images, and have been used in many applications to date [165, 166, 167]. The first and simplest of the features extracted from the wavelet coefficients were the so-called wavelet energy signatures, which were a representation of the energy contained within each band of the decomposition. Extending this, the mean deviation and other higher order moments have also been used for the purposes of texture classification and segmentation. Such features have been shown to provide good characterisation of textures in certain environments, and typically outperform single resolution techniques such as grey-level co-occurrence matrix features.

More recently, a number of new algorithms for extracting features from the coefficients of the wavelet transform have been proposed in the literature. Kim proposes a new non-separable set of wavelet filters for characterising texture which

are shown to outperform the more commonly used separable DWT [168]. Tabesh uses the zero-crossings of a wavelet frame representation to extract texture features, and has shown experimentally that these features contain information not contained within the energy signatures [169]. By combining these two feature sets, overall accuracy is improved by up to 70%. Shaffrey and Kingsbury have proposed a method of texture segmentation where each band of coefficients of a complex WT is modelled by a hidden Markov tree representation [170]. Van de Wouwer *et. al.* have proposed a set of features based on second-order statistics of the wavelet coefficients, calculated using co-occurrence matrices [83]. Numerous methods of extracting texture features using the wavelet packet transform have been proposed [79, 81, 116].

In each of the feature extraction techniques listed above, the coefficients of each band are analysed separately, with the correlations between bands of the same and different resolution levels ignored, even though it is well-known that strong relationships between neighbouring bands exists. Portilla and Simoncelli have shown that without knowledge of these correlations accurate reconstruction of the texture is not possible, indicating that this information is significant for characterising the texture [52].

This chapter proposes a novel method of characterising textures based on *wavelet scale co-occurrence matrices*, which capture information about the relationships between each band of the transform and the low frequency approximation at the corresponding level. A theoretical description of scale co-occurrence matrices is first presented, followed by a experimental evaluation of a number of distance measures comparing textures using these matrices. The mean-squared error, Kullback-Leibler distance, and the *earth mover's distance* (EMD) are evaluated in this study. The earth mover's distance is a recently proposed metric for evaluating the distance between two distributions which has shown to provide a more robust measure in some applications.

Finally, a set of wavelet scale co-occurrence features for texture classification

are proposed and evaluated using the classification system presented in chapter 5. Experimental results show that such features in many cases outperform those based on independent wavelet bands. Combining such features with those obtained independently from each band is also shown to improve performance for complicated textures, in particular those containing both macro- and micro-structures.

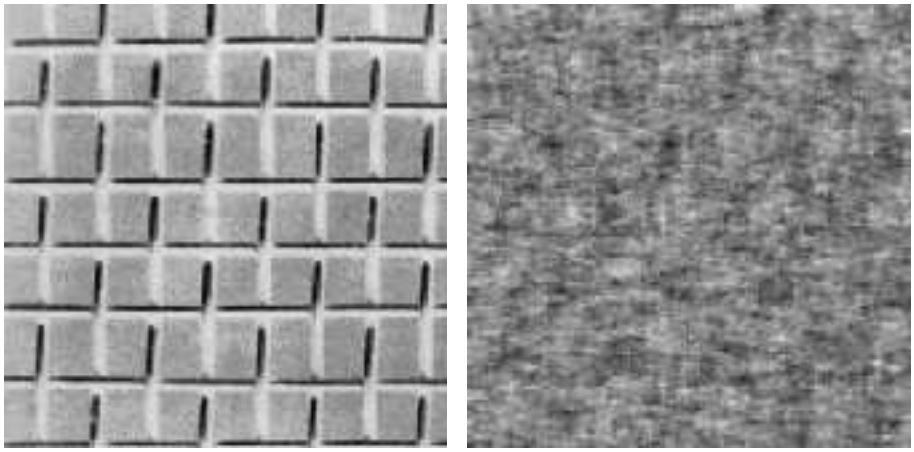
6.2 Limitations of Independent Wavelet Features

Julesz initial experiments in visual texture led him to conjecture that such images could be completely characterised by their second-order statistics [45]. Eventually, this was shown to be false, with many counter-examples presented showing visually distinct textures with identical second-order statistics. Recently, the main focus of much texture analysis research has centered around multi-scale filtering, with Gabor filters and the WT used to good effect. Common WT-based texture analysis techniques extract features from each band of the wavelet decomposition, measuring statistical information or modelling these coefficients via some parametric form. In this section, we show that it is possible for markedly different textures to have identical such statistics, indicating that they do not completely characterise texture. Examples of synthetic textures are provided to illustrate this point.

Wavelet energy signatures are one of most commonly used texture features in many applications, and can be calculated from the coefficients of the separable FWT by

$$E_{jl} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} D_{jl}(m,n)^2 \tag{6.1}$$

where D_{jl} are the wavelet detail coefficients are resolution level j, $l \in \{1,2,3\}$ indicates which of the detail images is being analysed, and M and N are the

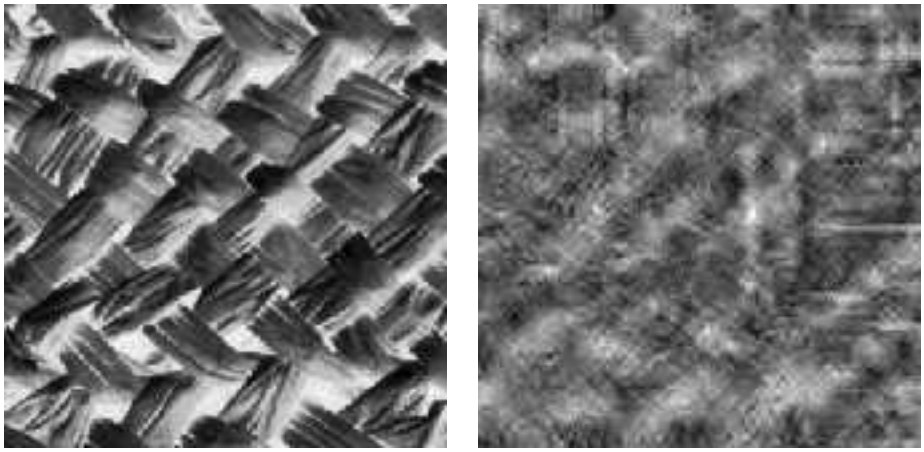


(a) (b)

Figure 6.1: Example showing the limitations of first-order statistics of wavelet coefficients as a texture descriptor. The natural texture (a) and synthesised texture (b) have identical first-order wavelet statistics of wavelet coefficients, yet are clearly distinguished by human observers.

dimensions of the coefficient matrix. Such features have been shown to perform well in various texture analysis applications, however they are not sufficient to fully characterise a texture. Figure 6.1 shows an example of a texture from the Brodatz album [125], along with a synthesised texture with identical wavelet energy signatures to 4 levels which was created using an identical low resolution approximation image. Clearly, these two texture can be easily discriminated by any human observer, and can by no definition of texture be considered identical. From these examples, it can be seen that the wavelet energy signatures fail to capture information regarding macro-structures in the texture such as the positioning, length and orientation of lines and edges.

In order to improve upon the performance of the first-order energy features, a method of modelling textures using both the first and second order statistics of wavelet detail coefficients has been proposed by Van de Wouwer *et. al.* [83]. Co-occurrence matrices are generated from each band of a redundant wavelet frame decomposition, and a set of eight standard co-occurrence features extracted to represent the second-order statistics of the texture. While this set of features



(a) (b)

Figure 6.2: Example showing the limitations of second-order statistics of wavelet coefficients as a texture descriptor. The natural texture (a) and synthesised texture (b) have identical first and second-order wavelet statistics of wavelet coefficients, yet are clearly distinguished by human observers.

improves performance compared to energy signatures, figure 6.2 shows that such a representation is still insufficient to completely characterise a texture. Again, the synthetic image is generated having equal first and second order statistics of wavelet coefficients for the first 4 decomposition levels, and using the same low resolution approximation image. While this artificial texture is a closer approximation than the example of figure 6.1, these two textures are clearly different. Without information to describe the relationships between each band of the WT, visual artifacts of the texture which contain elements at numerous scales are not adequately represented.

6.3 Wavelet Scale Co-occurrence Matrices

From the examples shown in the previous section, it is clear that features obtained independently from each band of the wavelet decomposition are not sufficient to fully characterise textured images. Portilla and Simoncelli have shown that relationships between direction and scale bands of the wavelet transform are in

many cases critical for adequate reconstruction of a textured image [52, 123], and use the correlations between the coefficients of a complex steerable pyramid decomposition to characterise texture for the purpose of synthesis. These features quantitatively measure the correlation between each orientation band at a given resolution level j, as well as between each detail image and the detail images at neighbouring resolution resolution levels. These parameters are then used, along with autocorrelation features of both the magnitude and raw coefficient values and various statistics of the grey-level values, to generate synthetic texture images by restraint enforcement.

The total number of parameters used in this approach is quite large, with more than 700 distinct features. For the purposes of classification and comparison, such a large number of features is often counter-productive, as it is difficult to effectively determine the relative importance of each feature is visually differentiating between two textures. It is proposed that a model of reduced complexity can more robustly characterise texture for classification, enabling effective discrimination between texture classes while allowing for small variations within these classes. For this purpose, a *scale co-occurrence matrix* $S(k,l)$ is defined as the probability of a detail coefficient $D(x,y)$ having a quantised value k while the approximation coefficient $A(x,y)$ at the same spatial position has a quantisedvalue of l. For an image of size $N \times M$, this can be expressed as

$$S_{ji}(k,l) = \frac{|\{(u,v) : q_1(D_{ji}(u,v)) = k, q_2(A_j(u,v)) = l\}|}{NM} \tag{6.2}$$

where A_j is the approximation image at resolution level j, D_{ji} are the three detail images, and $q_1(x)$ and $q_2(x)$ are the quantisation functions for the detail and approximation coefficients respectively. An overcomplete wavelet frame decomposition is used in our experiments in order to provide translation invariance and a higher robustness against nose, give higher spatial resolution, and avoid an overly sparse matrix at the lower resolutions.

The scale co-occurrence captures first order statistics of both the detail and approximation coefficients, and can be seen to fully encompass the wavelet mean

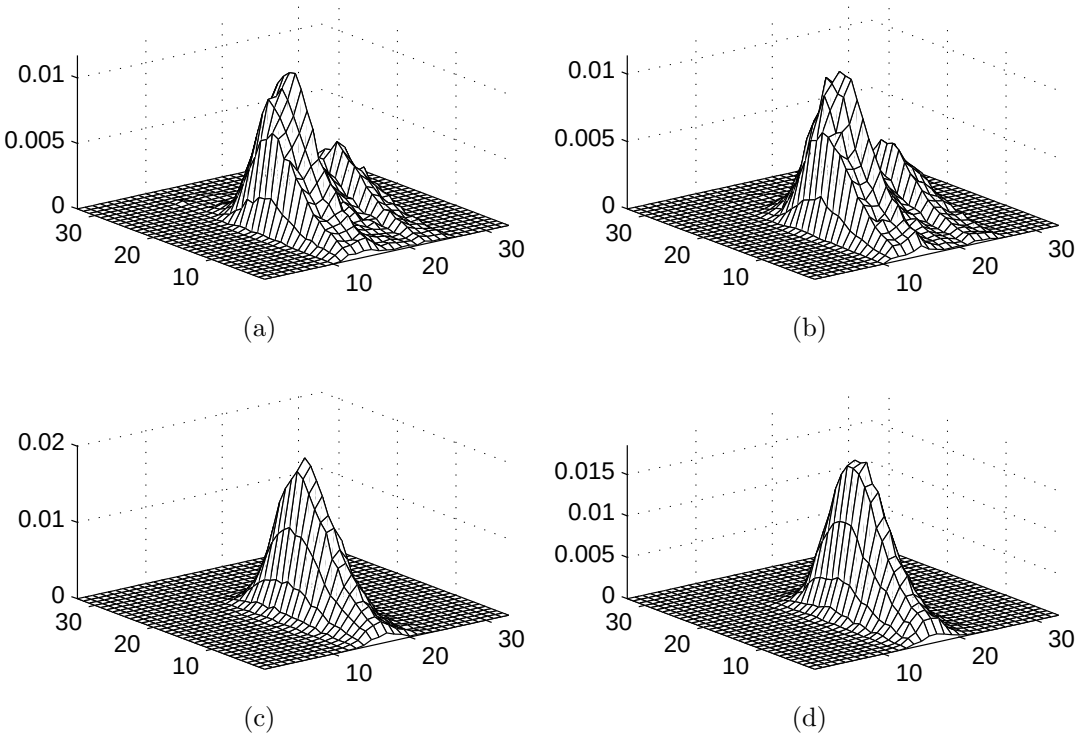


Figure 6.3: Examples of two wavelet scale co-occurrence matrices for each of the textures of figure 6.2, showing considerable differences. (a) and (b) show the horizontal and vertical scale co-occurrence matrices respectively for the first level decomposition of the texture of figure 6.2(a), while (c) and (d) show the same information for figure 6.2(b).

deviation signatures. More importantly, by defining the relationships between the low and high frequency information at each scale, much information regarding structural components of the texture such as lines and edges can be extracted. The scale co-occurrence matrices overcome many of the limitations of features modelling the first and second order statistics of wavelet coefficients, as can be seen in figure 6.3, which shows two of the scale co-occurrence matrices extracted from the textures of figure 6.2, which have identical first and second order statistics. These matrices are clearly distinguishable, and provide the discrimination power lacking in other wavelet texture features.

6.3.1 Pre-processing of Images

In order to ensure robustness of the scale co-occurrence matrix in the presence of variations in lighting and noise, it is necessary to pre-process each image before performing the wavelet decomposition. The approximation images, in particular, are sensitive to such variations, with changes causing linear translations and/or scaling in the resulting scale co-occurrence matrix. Although these changes do no cause significant visual changes, they can lead to significant errors when calculating distance metrics, particularly when using the mean squared error.

There exist a number of techniques to minimise the effects of variations in lighting changes and noise when extracting information from images. A simple method of normalising a set of images is mean subtraction, such that each image to be processed is guaranteed to have a zero mean, which for an image $I(x, y)$ of size $M \times N$ is defined as

$$\hat{I}(x, y) = I(x, y) - \frac{\sum_{v=1}^{M} \sum_{w=1}^{N} I(v, w)}{MN} \tag{6.3}$$

This is an attractive option since it is computationally inexpensive, and does not distort the image in any form. The frequency spectrum, with the exception of the DC component, is also unaffected, which means that only the low-frequency approximation of the wavelet transform coefficients will be altered. This approach does not however compensate for such environmental conditions as lighting changes, shadows, different image capture devices or noise, making it suitable only for applications with carefully controlled conditions.

Another common approach is contrast normalisation, which attempts to compress or expand the dynamic range of the image's grey level values into a specified range. This can be done using either fixed upper and lower limits, or by applying constraints to various statistical properties of the image, such as the standard deviation. This operation has a significant effect on the wavelet coefficients of the image, with each being modified by the same factor used to normalise the

contrast of the image. For a normalisation operation

$$\hat{I}(x,y) = \frac{I(x,y) - \mu_I}{\sigma_I} \tag{6.4}$$

where μ_I and σ_I are the mean and standard deviation of the image respectively, each wavelet detail image $D_{j,k}$ is scaled by a factor of σ_I.

Histogram equalisation is another common pre-processing technique whereby the histogram of the resulting image is approximately flat, meaning that the likelihoods of each of the pixel value in the desired range are equal. This operation can increase the contrast of degraded images, and often provides a visual improvement in image quality when applied to such images. The effect of this operation on the wavelet coefficients is significant and unpredictable, with a strong likelihood that different bands and regions of the image will be affected by varying amounts. Using this operation before extracting the scale co-occurrence representation of a texture has experimentally shown to decrease performance on many images, as it removes much of the significant information contained in each approximation band.

Binarisation of an image converts the original greyscale lattice into a binary representation, where each pixel can have one of only two distinct values, 0 or 1. Clearly, such an operation will significantly distort the image, and is useful only in a small subset of applications. The best example of such an application is the field of document processing, in which the images to be processed are often binary or near-binary to begin with, and consequently the binarisation process may restore contrast which is lost through the printing and scanning processes or due to noise. A more thorough review of the advantages and disadvantages of binarisation as well as examples of various approaches to this problem are given in section 7.4.2.

Experimentally, mean subtraction and contrast normalisation were found to provide the most improvement to the overall performance of the scale co-occurrence representation. Because of its lower computation cost, simple mean subtraction

has been used in all experiments in this chapter.

6.3.2 Quantisation of Approximation and Detail Coefficients

Chapter 4 presented a number of quantisation strategies which in many cases can significantly improve the performance of texture features. In particular, using logarithmic quantisation on the detail coefficients of the wavelet transform was shown to provide a better characterisation of many textured images, with improved classification results when using first and second order statistical features of each wavelet band. Because of these advantages, this method is again used to quantise the detail coefficients when constructing each matrix of the scale co-occurrence representation.

Section 4.4.3 presented three different quantisation functions which can be used when calculating the spatial co-occurrence matrices of each independent wavelet band. In this context, the function $q_1(x)$ was found to give the best results, largely due to its retention of the sign of each coefficient, which is important when considering spatial relationships at small distances. When constructing scale co-occurrence matrices, the sign of the coefficients is of lesser importance, as the histograms of such coefficients are generally symmetric or near-symmetric about the origin. Therefore, $q_2(x)$ and $q_3(x)$ will likely be better choices as they will provide greater resolution for the same number of quantisation levels. Due to the slightly higher classification accuracy of the wavelet log mean deviation features, $q_2(x)$ will be used for the quantisation of the detail coefficients when constructing the scale co-occurrence matrices, with $\delta = 0.001$.

The histograms of the wavelet approximation coefficients of a typical texture sample do not have the long tails that are often found in the detail images, and will generally be similar to that of the original image. As such, using a logarith-

mic quantisation function on these coefficients will in general not improve their characterision. For this reason, a uniform quantiser is used for this operation.

6.4 Scale Co-occurrence Matrices for Similarity Measure

Using the scale co-occurrence matrices defined previously, it is possible to calculate a similarity measure between two textures

$$SM = \frac{1}{\sum_{j=1}^{J}\sum_{i=1}^{3} d(S1_{ji}, S2_{ji})} \tag{6.5}$$

where $d(x)$ is a distance metric used to determine the difference of the two distributions $S1$ and $S2$. Such a similarity measure can be used in image retrieval tasks, and also in classification and segmentation problems, where a candidate image is assigned to the class with the highest similarity measure, or in verification tasks whereby an image is either accepted or rejected based on the score [171].

A number of different metrics are available for calculating the distance between two distributions, of which a few are outlined in the following sections.

6.4.1 Mean-Squared Error

A simple and computationally inexpensive method of calculating the distance between two distributions is the mean-squared error, defined mathematically as

$$d_{MS}(S_1, S_2) = \sqrt{\frac{\sum_{x=1}^{X}\sum_{y=1}^{Y}(S_1(x,y) - S_2(x,y))^2}{XY}} \tag{6.6}$$

where X and Y are the dimensions of the matrices. A similar metric which uses the absolute differences rather than the square is also commonly used, and is defined as

$$d_{MS}(S_1, S_2) = \frac{\sum_{x=1}^{X}\sum_{y=1}^{Y}|S_1(x,y) - S_2(x,y)|}{XY} \tag{6.7}$$

The mean squared and mean difference error metrics provide an estimate of the difference between two distributions based entirely upon the differences in the relative frequencies of each histogram bin, and does not consider the distance between bins, known as the *ground distance*, at all. In (6.7), the ground distance can be shown to be 0 when the locations of the bins are equal, and 2 otherwise. Because of this, these metrics are in many cases misleading, since large differences in the modes of two distributions can result in the same error as much smaller differences. This is especially noticeable when the histograms being compared are sparse in nature, as the probabilities of exact bin matches is decreased.

6.4.2 Kullback-Leibler Distance

An extension of the mean-squared error is the Kullback-Leibler distance, which can be regarded as an entropy measure, and is closely related to the cross entropy, or information of discrimination [141]. Given two distributions $q(x)$ and $p(x)$, this distance is defined as

$$d_{KL} = \sum_x q(x) ln \frac{q(x)}{p(x)} \tag{6.8}$$

While $d_{KL}(q(x), p(x)) = 0$ if and only if $q(x) = p(x)$, the Kullback-Leibler distance is not a true metric, as is it not necessarily symmetric. Because of this, the Kullback-Leibler distance is not an appropriate choice in some applications. Furthermore, because the distributions are compared only in terms of the relative frequencies of each bin, the ground distances are again disregarded in a similar fashion to the mean-squared error.

6.4.3 Mahalanobis Distance

Using the Mahalanobis distance it is possible to overcome some of the limitations of the mean-squared error and Kullback-Leibler distance to a certain extent. By taking into consideration the covariance of each variable within the distributions,

an estimate of the ground distance is obtained, and can be used to make a direct comparison of the means. This is mathematically defined as [141]

$$d_M = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)} \tag{6.9}$$

where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the means of the two distributions, and $\boldsymbol{\Sigma}$ is the covariance matrix of the variables involved. This covariance matrix can be obtained either by *a priori* knowledge from previous data, or by direct calculation from the two distributions. If the variables of each distribution are independent, $\boldsymbol{\Sigma}$ becomes diagonal, and the Mahalanobis distance is equivalent to the variance-normalised Euclidean distance.

While it incorporates some information regarding the ground distances and is a true metric, the Mahalanobis distance does have a number of limitations. Most notably, a strong assumption regarding the nature of each distribution is made, and data which is not Gaussian in nature will often yield poor results. It is also assumed that the covariance matrices of the distributions are equal, leading to misleading results in situations where this does not hold.

6.4.4 Earth Mover's Distance

The earth mover's distance (EMD) is a relatively new metric for representing the distance between two distributions in which the minimum amount of *work* required to transform one distribution to the other is calculated, given a set of ground distances between each point of the matrix [172]. Calculating this minimum amount can be viewed as a case of the *network transportation problem*, where one distribution is considered as a set of suppliers, and the other as a set of consumers. For the case of the scale co-occurrence matrix, each element of the matrix corresponds to a supplier or consumer.

Formally, the earth mover's distance can then be expressed as the minimum of

the cost function of a set of weighted flows f_{ij} given by

$$EMD = \min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} f_{ij} \tag{6.10}$$

where $\mathcal{I}$ and $\mathcal{J}$ are the sets of suppliers and consumers, and c_{ij} is the ground distance between bins i and j. To ensure a valid solution, the following restraints are also applied:

$$f_{ij} \geq 0 \tag{6.11}$$

$$\sum_{i \in \mathcal{I}} f_{ij} = y_j \tag{6.12}$$

$$\sum_{j \in \mathcal{J}} f_{ij} = x_i \tag{6.13}$$

where x_i is the total supply the supplier i and y_j the total capacity of consumer j, which in our case are represented by the values of each element of the scale co-occurrence matrices.

A solution to the transportation problem of finding f_{ij} can be achieved using the simplex algorithm [173], an iterative method which will eventually converge to a local minimum. Russell has proposed an algorithm to determine a near-optimal starting point for this algorithm, which is used to ensure that the final value is close to the global minimum.

Using the earth mover's distance on the full scale co-occurrence matrices of two textures involves solving a transportation optimisation problem, which is of computational complexity $O(N^2)$, for more than 1000 suppliers and consumers. Even for a relatively low number of iterations, this computational time can easily become excessive in all but the most trivial of applications. To improve this performance, it is necessary to significantly reduce the size of the flow optimsation problem without adversely affecting the accuracy of the distance metric. This can be accomplished by using *signatures* to represent the scale co-occurrence data rather than the traditional matrix form. Signatures, rather than representing fixed intervals, model a distribution using a set of clusters, and are defined

as [172]

$$\{\mathbf{s}_i = (\mathbf{m}_i, v_i)\} \tag{6.14}$$

where $\mathbf{m}_i$ are the means of each cluster, and v_i the weightings. If sufficient clusters are used, it is possible to represent any distribution with arbitrary accuracy. It can also be shown that the histogram or matrix representation is actually a special case of a signature in which the clusters are set at fixed equidistant intervals in the underlying space. Because of the possibility the each bin mean is not necessarily the mean of the distribution within it, it is possible that the signature representation can provide a more accurate model of the underlying data than the corresponding histogram form.

Calculating the signature of a distribution can be easily done by any one of a number of data clustering techniques. Using the k-means clustering algorithm has shown to produce acceptable results using a fixed number of clusters. In order to determine the number of clusters required, the mean-squared error between the scale co-occurrence matrix and signature representation was evaluated for a number of different textures, scales and number of clusters, with results showing that approximately 50 clusters is sufficient to adequately characterise the distribution for most textures. More information on the k-means algorithm can be found in [140].

Determining the Ground Distances

The optimal flows f_{ij} and thus the final value of the EMD is highly dependent on the set of ground distances c_{ij}. Generally, these values are expressed as a function of (i, j), which in our case are the indices (k, l) of the scale co-occurrence matrices S_1 and S_2, and thus a two-dimensional vector. One commonly used metric for the ground distance between 2D points is the Euclidean distance

$$d(k_1, l_1, k_2, l_2) = \sqrt{(k_1 - k_2)^2 + (l_2 - l_2)^2} \tag{6.15}$$

where (k_1, l_1) and (k_2, l_2) are the indices of the scale co-occurrence matrices S_1 and S_2 respectively. Other metrics include the city block or Manhattan distance

$$d(k_1, l_1, k_2, l_2) = |k_1 - k_2| + |l_2 - l_2| \tag{6.16}$$

which is a summation of the distance of each dimension, and the maximum distance

$$d(k_1, l_1, k_2, l_2) = \max(|k_1 - k_2|, |l_2 - l_2|) \tag{6.17}$$

which considers only the greatest of the differences over all dimensions.

Experimentally, a weighted Euclidean distance defined by

$$d(k_1, l_1, k_2, l_2) = \sqrt{a_k(k_1 - k_2)^2 + a_l(l_2 - l_2)^2} \tag{6.18}$$

where a_k and a_l are the weights for each dimension, has been found to give the most robust distance metric. Values of $a_k = 2$ and $a_l = 1$ are used in our experiments, meaning that differences in the detail coefficients are considered more important than a similar difference in the approximation coefficients.

6.4.5 Computational Considerations

The wavelet scale co-occurrence signature representation is quite small, less than 1kb for an image, making it suitable for indexing large collections of textures. This size can be further reduced by traditional compression algorithms, which generally perform quite well given the relatively sparse nature of the data. Calculation of the similarity measure using the mean squared error is very fast, with more than 1000 comparisons performed per second on a 1700MHz workstation. The calculation of the EMD when using the entire scale co-occurrence matrices is comparatively much more computationally expensive, with only around 10 comparisons per second possible for typical data. Using the signatures rather than the full co-occurrence matrices provides a significant improvement to the efficiency of calculating this distance metric, and although exact times are also data

dependent, around 200 comparisons per second is typical. On large databases or when computation speed is of critical importance, however, it may be necessary to further improve the computational efficiency of the search.

A suggested technique for pruning the search tree in large databases is to estimate a lower bound of the EMD, and use this estimate to remove unlikely match candidates. One such estimate of this lower bound is the distance between the centroids of the distributions, given using the notation of (6.10)-(6.13) as [172]

$$\min(EMD) = \left\| \sum_{i \in \mathcal{I}} x_i p_i - \sum_{j \in \mathcal{J}} y_j q_j \right\| \tag{6.19}$$

where p_i and q_j are the coordinates of each cluster in the signatures $\mathcal{I}$ and $\mathcal{J}$ respectively. Such a lower bound is significantly faster to computer than the EMD, and by setting a suitable adaptive threshold a large proportion of the total candidates can be removed from the search without compromising the final result.

Another computational improvement can be realised by using a tree structured search algorithm, whereby the lowest resolution matrices or signatures are first compared, and processing continued for only those samples with the highest partial similarity measure. By combining these two methods of searching, the computation time for a typical search is reduced by approximately 95% with no noticeable affect on the quality of the retrieved matches.

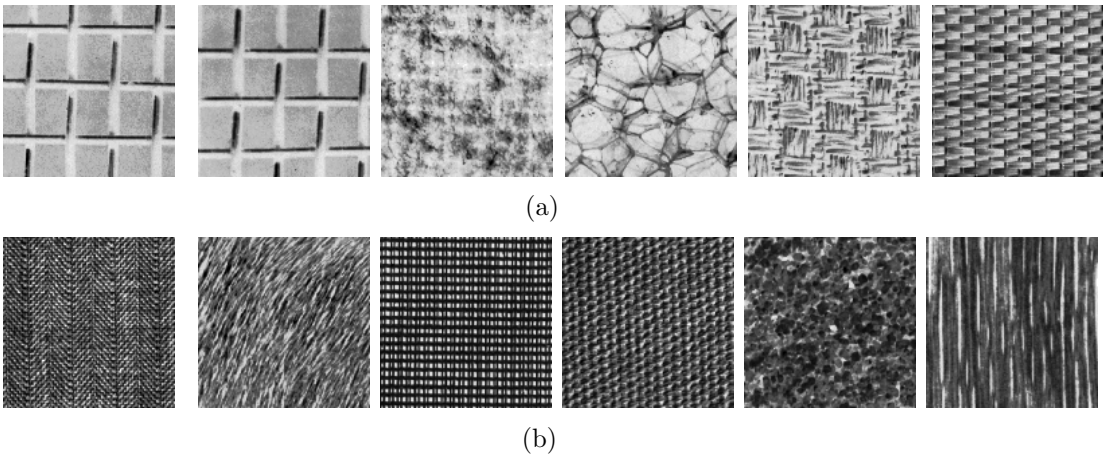
6.4.6 Texture Retrieval Results

Using the distance metrics shown above, experiments were conducted in both image retrieval and classification tasks. To simulate the retrieval of images from a database of textures, 50 images from the Brodatz album were used to form a small database of textures. The scale co-occurrence matrix representations of each of these images was extracted to 4 levels of wavelet decomposition, and used to create an index into the database. A test set of 200 images was then selected from the same 50 images, 4 from each class. In all cases, the training and test

images were extracted from separate parts of the image such that no overlap between the two sets of possible. The top 5 matches for each of these test cases were then found in the database using the proposed similarity measure using both the mean-squared and EMD metrics. Using the mean-squared error, the image of the same class as the test case was returned as the most similar image in 95.5% of cases, with 9 samples returning another class of image as the most likely. These results were improved when the EMD was used, with an image of the same class returned as the most likely match in all but 6 cases, for an overall accuracy of 97%. In all cases the image of the same class as the query image was amongst the top three matches. Tests were also conducted using textures not present in the database, in order to examine the response of the similarity measure in situations where a perfect match cannot be found.

The results of a typical query for both of these situations are shown in figure 6.4. From these images, it can be seen that the textures retrieved from the database are visually similar to the candidate image, indicating that the scale co-occurrence matrices provide a good characterisation of the visual appearance of texture. When the texture was not present, the returned images were of textures of similar scale, intensity and direction. For comparison purposes, the database search for the same two images was repeated using the wavelet energy features to calculate a distance measure. These features are pre-normalised to have zero mean and unit variance to prevent any bias due to scale. The top five returned matches using this techniques are shown in figure 6.5.

From these results it can be seen that the matches returned by the proposed scale co-occurrence representation are more visually similar to the query images than those obtained using the wavelet energy features, with distance scores more accurately reflecting these similarities. The relative stability of the distance measure is also notable, with a high correlation between the visual difference between two texture samples and the metric returned. This is in contrast to the results obtained using typical energy features, as can be seen from the vastly different



(a)

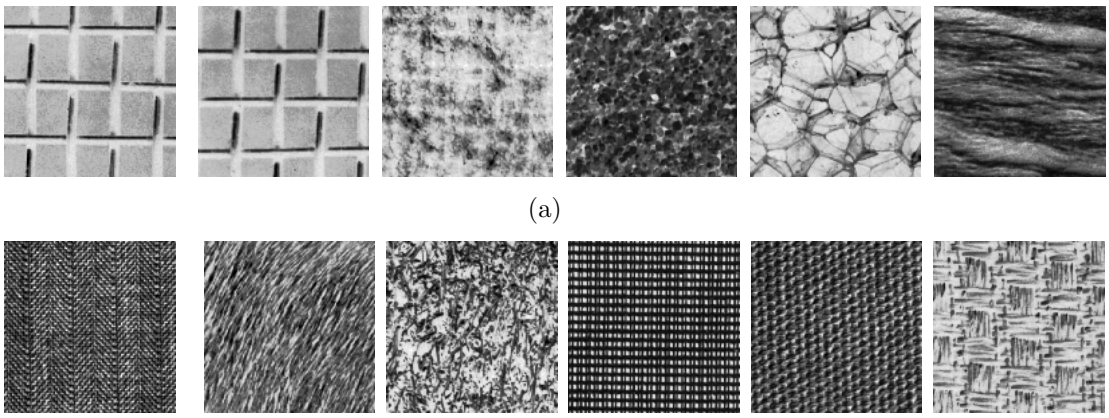
(b)

Figure 6.4: Results of two typical queries using the proposed scale co-occurrence similarity measure and the EMD. The query image (left) and top 5 matches are shown in each case. (a) Query texture is present in database, with distance measures of 42.7, 264, 274, 347.1 and 495.4 respectively, and (b) query texture is not present in the database, distances measures of 196.8, 207.8, 226.7, 401.2 and 457.1 respectively.

scores obtained for each of the two queries shown in figure 6.5. The textures returned from the second query using the scale co-occurrence representation, whilst still similar in appearance to the original image, have considerably larger distance scores than those of the first.

6.4.7 Texture Classification using Similarity Measure

Classification textures is also possibly using the proposed similarity measure, with the advantage that a meaningful estimate of the confidence of the classification decision is also provided by means of the similarity measure itself. A model for each texture class is created by calculating the average of each scale co-occurrence matrix or signature over all of the training observations of each class ω_c. An unknown sample can then be classified by comparing its scale co-occurrence representation with each such model, and choosing that with the highest similarity calculated using the proposed technique.



(a)

(b)

Figure 6.5: Results of two typical queries using wavelet energy features. The query image (left) and top 5 matches are shown in each case. (a) Query texture is present in database, with distance measures of 0.24, 1.43, 1.56, 2.95 and 3.28 respectively, and (b) query texture is not present in the database, distances measures of 40.4, 43.7, 46.4, 47.3 and 51.5 respectively.

To improve the accuracy of this classification system, weightings can be applied to the contribution of each individual scale co-occurrence matrix or signature towards the final similarity measure, reflecting the relative importance of each to the characterisation of that texture class.

Matrix Weightings

When considering a class of textures, it is common for the majority of the distinguishing features of that texture to be contained within a subset of the full wavelet decomposition, with other bands having either low energy, or containing information which is of little relevance. Noise[1] in these bands, although having little visual effect on the texture, may strongly influence the resulting distance metric. It is desirable, therefore, to reduce the contribution of such bands to dampen the effects of this noise. Additionally, it is desirable that those bands of

[1]In this context, noise is defined as any part of the image which does not play a significant role in the characterisation of the texture, or varies significantly between samples of the same texture.

a texture which naturally show significant variation within a class do not unduly skew the resulting similarity measure.

By assigning each band an importance value when performing comparisons between textures of a particular class, it is possible to somewhat achieve this goals. In terms of the proposed scale co-occurrence matrix representation of texture, bands with little interclass variation are considered to have a high importance value, while those with a relatively high degree of variance are considered to be of lesser significance. In calculating the similarity measure SM, this measure of relative importance is represented by the weighting factors w_{ji}, where $\sum_i \sum_j w_{ji} = 1$. Using these weightings, the similarity measure of (6.5) can be modified, giving

$$SM = \frac{1}{\sum_{j=1}^{J} \sum_{i=1}^{3} w_{ji} d(S1_{ji}, S2_{ji})} \tag{6.20}$$

Calculation of the weightings w_{ji} is performed separately for each class ω_c, based on a statistical analysis of the intraclass variations of each matrix. Accordingly, bands which contain little variation within the class are assigned a high weighting, while those with relatively high variation are given a lower importance. Calculating each of the weights is done using the following equations:

$$w_{jic} = \frac{1/\bar{d}_{ji}}{\sum_k \sum_l 1/\bar{d}_{kl}} \tag{6.21}$$

where $\bar{d}_{ji}$ is the average distance between the scale co-occurrence matrices S_{ji} of class c, calculated by

$$\bar{d}_{ji} = \frac{\sum_{k=1}^{N} \sum_{l=1}^{N} d(S_{kji}, S_{lji})}{N(N-1)} \tag{6.22}$$

where N is the number of training observations of class c, and S_{kji} are the scale co-occurrence matrices of the k_{th} such observation.

In applications where unsupervised classification or retrieval is required, calculating such local weightings is not possible since there is no knowledge of the class labels during training. As such, it is necessary to calculate the weightings w_{ji} using some *a priori* knowledge of the problem domain.

Metric	Classification Error
Mean-squared	5.1%
Kullback-Leibler	4.7%
Mahalanobis	7.9%
EMD	3.7%

Table 6.1: Results of classification experiments using the proposed similarity measure for both the mean-squared error and earth mover's distance metrics.

Results

Texture classification experiments using the 25 textures from the Brodatz album shown in figure 4.6 were performed using the proposed representation and similarity measure. The mean-squared error, Kullback-Leibler distance, Mahalanobis distance and EMD were each used to generate both the weightings and the final distances, and classification performed by choosing the class with the lowest overall distance. Table 6.1 shows the overall classification error rates using each of these similar functions. From these results it can be seen that the EMD outperforms the other distance measures by a significant margin, with an error rate approximately 25% lower than the Kullback-Leibler distance. The Mahalanobis distance performed poorly, which is likely due to its assumption of Gaussian-like distributions, and the fact that only the means of the

These classification results are comparable with those obtained using second-order statistics of wavelet coefficients, as well as having the advantage of providing a meaningful estimate of classifier confidence via the final similarity measure.

6.5 Wavelet Scale Co-occurrence Features

As well as being used for the purpose of calculating a similarity measure between two textures, features can be extracted directly from the scale co-occurrence

matrices for use in both classification and segmentation tasks. Table 3.2 lists a number of features extracted from spatial co-occurrence matrices, and although the meanings are somewhat different in the context of scale co-occurrence, this set can be modified to give a robust measure of the texture. Previous work in this area has shown that even unmodified, this set of features can give excellent discrimination between a wide range of textures [174].

The first two features from table 3.2, energy and entropy, measure the spread of a distribution and can be calculated for any matrix, and thus should give some meaningful information regarding the structure of the scale co-occurrence matrices. As such, these features can be used without modification. Inertia and contrast, however, have a meaning which is specific to the spatial domain from which co-occurrence matrices are generally extracted. These two features, however, have been shown experimentally to have the most discriminating power of all the co-occurrence features, and can successfully capture many aspects of the shape of a co-occurrence matrix. As such, these features will be tested in the context of scale co-occurrence matrices.

Local homogeneity measures the uniformity of a local spatial region, however can also be interpreted as a measure of the correlation between each dimension of the matrix. As such, this feature may still have some usefulness for distinguishing between scale co-occurrence matrices. The maximum probability feature is still meaningful as it measures the peak value of the distributions. However, this feature is generally considered to be of limited use in practical applications, as it is highly susceptible to small image variations and quantisation effects. The final four features of table 3.2, inverse difference moment, cluster shade, cluster prominence and information measure of correlation, all retain their meaning when applied in the new context, however these features are generally of limited value in texture discrimination tasks.

One new feature which can be calculated from scale co-occurrence matrices is the

Co-occurrence Feature	Classification Error
Energy	21.2%
Entropy	19.4%
Inertia	7.3%
Contrast	6.4%
Homogeneity	9.8%
Max. Prob.	31.1%
Inv. Diff. Mom	13.6%
Cluster Shade	18.3%
Cluster Prom.	28.4%
Inf. Meas. of Corr.	30.6%
Correlation	5.4%

Table 6.2: Performance of the individual co-occurrence features when applied to the scale co-occurrence representation of texture.

correlation, defined as

$$Corr. = \sum_i \sum_j (i - M_x)(j - M_y)P(i,j) \tag{6.23}$$

where M_x and M_y are the means of each axes of the matrix as defined in table 3.2. This feature gives an overall measure of the correlation between the approximation and detail coefficients of the wavelet transform, and is useful in providing a quantitative measure of the presence of texture structures which span multiple resolutions, such as edges and lines.

Table 6.2 shows the performance of each of the features individually on a sample set of texture images. Although these results do not necessarily reflect the true contribution of each feature when they are combined, it does give an indication of the discriminatory performance of each. From these results, it can be seen that the best discrimination is accomplished by the correlation feature, which alone outperforms the commonly used wavelet energy signatures. As expected, the maximum probability, cluster shade, cluster prominence and information measure of probability do not provide the same level of discrimination as the other features, while inertia and contrast perform quite well.

6.5.1 Classification Results

Texture classification experiments were conducted using the extracted wavelet scale co-occurrence features on the same set of 25 texture images from the Brodatz album shown in figure 4.6. Again, each image was divided into two equally sized segments, with one used for training the classifier and the other used for testing. Five independent training and testing sets, each containing 100 64×64 pixel samples, were extracted from each training and testing image, and the scale co-occurrence features described in the previous section extracted to four wavelet decomposition levels. Simple mean subtraction was used as pre-processing, and the logarithmic quantisation process described in chapter 4 used with $\delta = 0.001$ to quantise the wavelet coefficients to 32 levels. Uniform quantisation was performed on the approximation coefficients, with 32 levels also used. To provide a measure of the relative performance of these features, the wavelet energy, wavelet co-occurrence and wavelet log co-occurrence features were also extracted.

A k-nn classifier was used for classification of the test samples, with the results shown in table 6.3. As expected, the simple wavelet energy signatures give the highest overall classification error for all test sets with an average rate of 7.6%. Extracting second order statistics of the wavelet coefficients provides a considerable increase in performance, with the wavelet co-occurrence and wavelet log co-occurrence signatures resulting in average errors of 2.9% and 1.4% respectively. The proposed scale co-occurrence features further increase classification performance, with a minimum error rate of only 1.2% over the entire set of texture classes. The high performance of the scale co-occurrence features was relatively consistent over each of the five trials, being outperformed by the WLC_1 features on only one occasion.

The error rates of the individual texture classes from one of the test sets for each of the tested feature sets are shown in table 6.4. From this information it can be seen that the scale co-occurrence features significantly outperform the other

Test Set	Wavelet Energy	Wavelet Cooc.	$\mathbf{WLC}_1$ $\delta = 0.001$	Scale Cooc.
1	7.8%	2.7%	1.6%	1.2%
2	7.6%	3.2%	1.3%	1.1%
3	8.2%	3.1%	1.4%	1.4%
4	7.0%	3.4%	1.8%	1.2%
5	7.2%	2.1%	0.8%	1.1%
Avg.	7.6%	2.9%	1.4%	1.2%

Table 6.3: Results of texture classification using the scale co-occurrence features, compared to those obtained using the wavelet log co-occurrence features and wavelet energy signatures.

features in classifying the D9 class, with an error of only 9% compared 33% for the second-order features. For other texture classes, however, the scale co-occurrence features do not perform as well, with slightly increased error rates in a few classes. In no case did scale co-occurrence features result in higher error rates than the wavelet energy signatures, which provides evidence to confirm that the proposed features do indeed capture all relevant first order statistics of the wavelet bands. Figure 6.6 shows the textures which were classified with significantly different error rates by the wavelet log co-occurrence and scale co-occurrence features.

To validate these results, the same texture experiments were performed using two additional sets of images, as described in section 4.5.3. These images were again divided into two equal sized regions, with 50 samples, each of 64×64 pixels, extracted from each. The wavelet energy, wavelet co-occurrence, wavelet log co-occurrence and scale co-occurrence features were extracted from each sample, and the performance of each determined using a k-nn classifier. Once again, the value of k providing the best results is used in all cases.

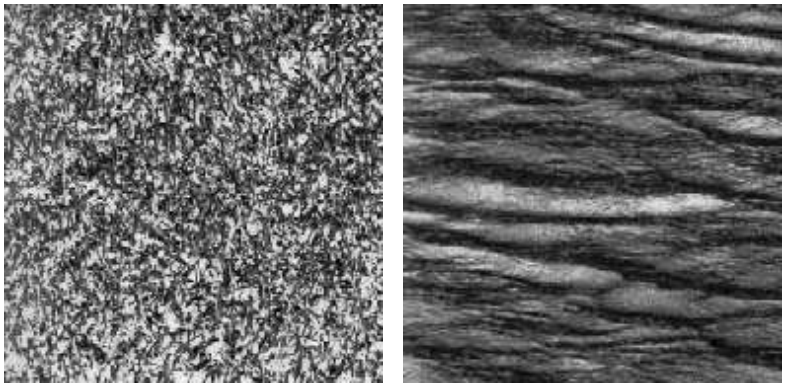
The results using these additional sets of images are shown in tables 6.5 and 6.6. These results show the robustness of the proposed scale co-occurrence features, with the scale co-occurrence features showing similar classification accuracy on the first dataset, and a significant improvement when applied to the more chal-

Texture Class	Wavelet Energy	Wavelet Cooc.	WLC_1 $\delta = 0.001$	Scale Cooc. $\delta = 0.001$
D1	2%	0%	0%	0%
D11	0%	0%	0%	1%
D112	20%	12%	4%	4%
D16	0%	0%	0%	0%
D18	3%	1%	0%	1%
D19	21%	1%	0%	1%
D21	0%	0%	0%	0%
D24	0%	0%	0%	1%
D29	13%	1%	0%	1%
D3	10%	0%	0%	2%
D37	9%	0%	0%	5%
D4	0%	0%	0%	0%
D5	24%	4%	4%	8%
D52	14%	1%	0%	0%
D53	0%	0%	0%	0%
D55	2%	0%	0%	0%
D6	1%	0%	0%	0%
D68	0%	0%	0%	0%
D76	2%	0%	0%	0%
D77	6%	0%	0%	0%
D80	1%	0%	0%	0%
D82	6%	0%	0%	0%
D84	1%	0%	0%	0%
D9	48%	32%	33%	9%
D93	2%	0%	0%	1%

Table 6.4: Individual error rates for each individual texture class for scale co-occurrence features, wavelet log co-occurrence features and wavelet energy signatures.

Feature Set	Error
Wavelet Energy	3.5%
Wavelet Cooc.	1.0%
WLC_1	0.4%
Scale Cooc.	0.5%

Table 6.5: Classification errors for all tested features using the second set of texture images.



(a) (b)

(c)

Figure 6.6: Textures which were classified with significantly different error rates by the wavelet log co-occurrence and scale co-occurrence features. (a) Texture D9 showed lower error rates using the proposed features, while those of (b) D37 and (c) D5 were higher.

lenging Vistex database. From examination of the Vistex textures, it can be seen that many of the images are not fully perpendicular to the camera plane, resulting scale changes across the image. The error rates for textures in which these scale changes were present were observed to be noticeably higher overall for all texture features with the exception of the proposed scale co-occurrence features. From these results it can be concluded that the scale co-occurrence features are more robust to small changes in scale than the wavelet energy and wavelet co-occurrence signatures which are extracted separately from each band. This property makes them a more attractive choice for many applications where such conditions cannot be strictly controlled, such as artificial vision and robotics

Feature Set	Error
Wavelet Energy	21.9%
Wavelet Cooc.	7.9%
WLC_1	3.1%
Scale Cooc.	1.9%

Table 6.6: Classification errors for all tested features using the texture obtained from the Vistex database.

devices.

The computational cost of extracting the wavelet scale co-occurrence features is relatively low, approximately half that of the wavelet co-occurrence and wavelet log co-occurrence signatures. To extract features from a single 64×64 sample takes less than 0.1s on a 1600MHz workstation, although this performance could likely be optimised further.

6.6 Fusion of Scale and Spatial Wavelet Co-occurrence Features

From table 6.4 it can be seen that although the wavelet co-occurrence and WLC_1 features have different overall error rates, the distribution of these errors amongst the classes is similar for both sets of features. From these observations it can be inferred that the information extracted by both techniques is similar in nature, with the WLC_1 features providing a somewhat better and more robust representation of this information. In contrast, the wavelet log co-occurrence features and scale co-occurrence features, which model the intra- and inter-band relationships respectively, have varying levels of performance when applied to different types of textures in different conditions. This is illustrated by the significantly different distributions of classification errors for each of these features, as well as the difference in overall accuracy when using the second set of images.

Additionally, while the independently extracted wavelet log co-occurrence features perform extremely well for well matched training and test conditions, they tend to be more adversely effected by environmental conditions such as scale changes and noise. The scale co-occurrence features, by modelling relationships between subbands, are somewhat more robust towards these changes, and are often a better choice in these situation. From these observations it can be conjectured that the information contained in each set of features is somewhat independent, and an overall improvement in accuracy may be achieved by combining this information in an optimal manner. To test if this is indeed the case, a number of combination strategies are employed and the classification results analysed in this section. In order to conduct these experiments, it is necessary to use a classifier which provides likelihood scores for each class. For this reason, the classification system proposed in chapter 5 is used for all experiments rather than the k-nn classifier.

6.6.1 Combination Strategies

In recent times there has been much research into the combination of features and classifiers for optimal performance. In essence, there are two basic approaches to this problem:

Pre-fusion involves the fusion of the feature vectors before classification. This can be accomplished by either simple concatenation of the vectors into a single vector, or by selecting a subset of each vector and combining them using other methods.

Post-fusion attempts to combine the outputs of two classifiers trained on separate, preferably independent data. In most cases, such an approach requires that each classifier output a likelihood score for each candidate class, with these scores then combined in some manner to determine the final class label.

Pre-fusion is a simple and somewhat naïve approach to the combination of two feature sets, as it simply passes the problem of deciding the most relevant features to the classifier. In addition, since the dimensionality of the combined feature set is necessarily larger (assuming no feature selection or reduction is used) than either of the two feature sets taken separately, the curse of dimensionality problem will be a factor when using all but the largest training sets.

The results of pre-fusion in the context of texture classification are shown in table 6.7. In this case, the WLC_1 features and scale co-occurrence features are combined into a single feature vector, and classified using the system proposed in chapter 5. The results of this experiment show what is known as *catastrophic fusion*, whereby the classification performance of a combined set of features is worse than either of the sets individually. This phenomenon can be explained by a number of factors. Firstly, the curse of dimensionality problem means that the models created using the combined feature set may be inadequate to describe the complexity of the fused data. Additional error is introduced when the mismatched observations of one feature set override those of the other classifier which may otherwise have provided a correct result.

Post-fusion techniques are a relatively recent advance in the field of pattern recognition, stemming from the observation that patterns misclassified by different classifiers using the same training and test observations did not necessarily overlap, suggesting that they offered complementary information about the problem [175]. Since then, a number of methods of fusing the results of multiple classifiers, trained using either the same or independently extracted features, have been proposed using a variety of methods [176, 177, 178, 179, 180].

In a Bayesian framework, given a set of Q observation vectors $\mathbf{x}_i,\ i = 1, \ldots, Q$ extracted independently from a given pattern P, should be labelled as the class ω_j which has the maximum *a posteriori* probability, ie

$$\theta = \arg\max_k P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_Q) \tag{6.24}$$

While this is theoretically the correct decision, as it takes into account each of the observations, in practice this is not always tractable as it would depend on knowing the joint probability density functions over the entire set of observations. It is possible however to simplify (6.24) such that it is expressed in terms of the independent *a posteriori* probabilities of each separate classifier, a much more tractable and practical solution. A number of these approaches are presented in the following sections.

6.6.2 Product Rule

By assuming independence between the Q observation vectors, it is possible to express the joint probability densities of each class as

$$p(\mathbf{x}, \ldots, \mathbf{x}_Q|\omega_k) = \prod_{i=1}^{Q} p(\mathbf{x}_i|\omega_k) \tag{6.25}$$

By applying Bayes theorem, and substituting (6.25), the *a posteriori* probability of each class can then be expresses as

$$P(\omega_k|\mathbf{x}, \ldots, \mathbf{x}_Q) = \frac{P(\omega_k)\prod_{i=1}^{Q} p(\mathbf{x}_i|\omega_k)}{\sum_j \left[P(\omega_j)\prod_{i=1}^{R} p(\mathbf{x}_i|\omega_k)\right]} \tag{6.26}$$

Using (6.26), (6.24) can be rewritten as

$$\theta_{PROD} = \arg\max_k P^{-(Q-1)}(\omega_k) \prod_{i=1}^{Q} P(\omega_k|\mathbf{x}_i) \tag{6.27}$$

which is known as the *product rule* of classifier combination. This can be seen as a rather severe strategy, as a low probability from a single classifier is sufficient to almost completely inhibit the selecting of a particular class. The effects of this in practice are usually undesirable, as mismatches train and test conditions can often incorrectly cause extremely low probabilities in a single classifier.

6.6.3 Sum Rule

Another assumption which can be made regarding the *a posteriori* probabilities of each class $P(\omega_k|\mathbf{x}_i)$ is that they do not differ significantly from the *a priori* probabilities $P(\omega_k)$. Although this is a strong assumption, it is justified in situations where this is a high train/test mismatch, that is, the environmental conditions such as noise during testing are significantly different to those during training. Given this assumption, the *a posteriori* probabilities can then be expressed as a factor of the priors, such that

$$P(\omega_k|\mathbf{x}_1) = (1 + \delta_{ik})P(\omega_k) \tag{6.28}$$

where $\delta_{ik} \ll 1$. Substituting this into (6.27) gives

$$\theta = \arg\max_k P(\omega_k) \prod_{i=1}^{Q} (1 + \delta_{ik}) \tag{6.29}$$

Expanding (6.29), and ignoring any terms of second order or higher, this can be approximated as

$$\theta_{SUM} = \arg\max_k (1 - Q)P(\omega_k) + \sum_{i=1}^{Q} P(\omega_k|\mathbf{x}_i) \tag{6.30}$$

This form of combination is known as the *sum rule*, and is particularly useful in mismatched train/test conditions [175].

6.6.4 Other Combination Strategies

Based on these two combination rules, a number of other strategies for combining the outputs of classifiers have been proposed [181].

The majority vote combination rule regards the output of each classifier as a binary decision, that is, 1 if a class has the highest *a posteriori* probability, and 0 otherwise. These votes are then summed over the entire suite of classifiers in order to reach a final decision. Clearly, this strategy is of little use in a situation where

only two classifiers are being used, as they will either agree, in which case the combination is trivial, or disagree, in which case another method of deciding the output must be applied. A suitable method for resolving tied votes when using any number of classifiers must also be decided upon as well. An advantage of this technique is that no actual likelihood scores from each classifier are required, meaning that it is suitable for use with the k-nn classifier.

The minimum and maximum probability rules combine the outputs of each classifier based on the minimum and maximum posterior probabilities over the entire set of classifiers respectively. These can then be expressed as

$$\theta_{MIN} = \arg\max_k \left[\min_{i=1}^{Q} P(\omega_k|\mathbf{x}_i) \right] \tag{6.31}$$

and

$$\theta_{MAX} = \arg\max_k \left[\max_{i=1}^{Q} P(\omega_k|\mathbf{x}_i) \right] \tag{6.32}$$

A similar combination strategy uses the median value of the posterior probabilities rather than the minimum or maximum. This is known as the median rule, and is expressed as

$$\theta_{MED} = \arg\max_k \left[\mathrm{med}_{i=1}^{Q} P(\omega_k|\mathbf{x}_i) \right] \tag{6.33}$$

These three combination strategies all use only a single classifier output in the final determination of the class label, meaning that the contributions of the others is completely disregarded. Additionally, the minimum rule suffers from the same drawback as the product rule - a single low probability is sufficient to exclude a particular class from selection. The median rule is clearly unsuited for a two classifier system.

6.6.5 Experimental Results

Each of the combination strategies outlined above, with the exception of the median rule and majority vote methods which are not appropriate for a two classifier system, were used to combine the outputs of two separate classifiers

Combination Strategy	Classification Error
WLC_1 Only	1.3%
WSC Only	1.1%
Pre-fusion	1.6%
Prod. Rule	1.2%
Sum Rule	0.6%
Min. Rule	1.8%
Max. Rule	0.9%

Table 6.7: Average texture classification error rates for each set individually, the combined feature using pre-fusion, and using various combination strategies.

trained using the WLC_1 and scale co-occurrence features. The results for these classification experiments are shown in table 6.7, along with the results of each classifier used separately and the results of using pre-fusion.

The results of these experiments show that the method of combination used when combining the outputs of the classifiers trained using each of the feature sets is crucial, with the overall error rate showing a significant improvement with some techniques, while actually performing worse than either classifier when using others. Pre-fusion, as expected, did not perform well, with the overall error rate being higher than either of the two sets separately. The minimum probability rule also showed extremely poor performance, again showing a higher error rate than either of the classifiers did separately. The sum rule gave the best overall performance, with the average error rate decreasing by almost 50% over each of the five trials.

6.7 Chapter Summary

In this chapter, a novel texture representation, the scale co-occurrence matrix, is presented. Such a representation has been shown to provide a robust characterisation of texture by modelling the relationships between the detail and

approximation coefficients of the wavelet transform, and addresses some of the limitations of features extracted independently from each band. A pre-processing method for normalising the grey-level histograms of these images, and a quantisation strategy for both the detail and approximation images are also presented.

Using the proposed scale co-occurrence representation, a similarity measure has been defined which gives a meaningful estimate of the similarity between two textures in either supervised or unsupervised applications. Using the recently proposed earth mover's distance when calculating such a similarity measure has been shown to provide a more accurate and robust estimate when compared to the mean-squared error and mean absolute error. Assigning weights to each of the calculated matrices based on their relative importance has been shown to further improve the resulting similarity measure, with both global and local weights defined for unsupervised and supervised problems respectively. Experimental results indicate that the proposed measure is effective for the purpose of image retrieval from a large database of textures, as well as texture classification applications.

A set of texture features extracted from the scale co-occurrence matrices has also been proposed, which have been experimentally shown to give lower error rates than methods using second order statistics of individual wavelet bands at considerably reduced computational expense. These features have also been found to perform well on textures that are misclassified by other techniques, from which it can be concluded that they contain significant new texture information. In particular, the scale co-occurrence features perform considerably better in the presence of small scale changes, a property which makes them an attractive choice in many applications where such changes are inevitable.

Finally, a number of methods of combining the wavelet log co-occurrence and scale co-occurrence features were investigated, including the product, sum, minimum, maximum and median combination rules. Experimental results show that the sum rule provides the best performance for this task, with the overall error rate

reduced by approximately 40% compared to each classifier individually. This is in contrast to using a pre-fusion technique, which resulted in a *higher* overall error rate.

Chapter 7

Script Recognition and Document Analysis

7.1 Introduction

As the world moves ever closer to the concept of the 'paperless office', more and more communication and storage of documents is performed digitally. Documents and files that were once stored physically on paper are now being converted into electronic form in order to facilitate quicker additions, searches and modifications, as well as to prolong the life of such records. A great proportion of business documentation and communication, however, still takes place in physical form, and the fax machine remains a vital tool of communication worldwide. Because of this, there is a great demand for software which automatically extracts, analyses and stores information from physical documents for later retrieval. All of these tasks fall under the general heading of *document analysis*, which has been a fast growing area of research in recent years.

A very important area in the field of document analysis is that of optical character recognition (OCR), which is broadly defined as the process of recognising either

printed or handwritten text from document images and converting it into electronic form. To date, many algorithms have been presented in the literature to perform this task, with some of these having been shown to perform to a very high degree of accuracy in most situations, with extremely low character-recognition error rates [182]. However, such algorithms rely extensively on a-priori knowledge of the script and language of the document, in order to properly segment and interpret each individual character. Whilst in the case of Latin-based languages, such as English, German and French, this problem can be overcome by simply extending the training database to include all character variations, such an approach will be unsuccessful when dealing with differing script types. Many script types and languages do no lend themselves to easy character segmentation, an essential part of the OCR process, and thus must be handled somewhat differently. Thus, the determination of the script of the document is an essential step in the overall goal of OCR.

A number of existing techniques for script recognition utilise character-based features, or connected component analysis. The paradox inherent in such an approach is that it is sometimes necessary to know the script of the document in order to extract such components. In addition to this, the presence of noise or significant image degradation can also significantly affect the location and segmentation of these characters, making them difficult or impossible to extract. Hence there exists a need for a method of script recognition which does not require such knowledge. Texture analysis techniques appear to be a logical choice for solving such a problem as they give a global measure of the properties of a region, without requiring analysis of each individual component of the script. Printed text of different scripts is also highly pre-attentively distinguishable, a property which has long been considered a sign of textural differences [44].

This chapter will give an overview of the field of document analysis and OCR, and a detailed summary of work done on the specific problem of script recognition from document images. A method for determining the script of printed

text using texture analysis features is then presented, covering the binarisation of the document, skew detection and correction, normalisation of text, extraction of texture features and final classification. A number of texture features are evaluated in this context, including the wavelet log energy, wavelet log co-occurrence and wavelet scale co-occurrence features developed in chapters 4 and 6. Recognition results using these features are reported using a database constructed with document images of 10 different scripts and the classification system described in chapter 5.

Section 7.5 outlines a technique for improving classifier performance by utilising the common properties of printed text when training the GMM. It is proposed that by taking advantage of this *a priori* information, less training data will be required to adequately describe each class density function, leading to increased over performance.

The problem of multi-font script recognition is addressed in section 7.6. This is of much importance in practical applications, since the various fonts of a single script type can differ significantly in appearance and are often not adequately characterised in feature space if modelling with a single class is attempted. A method for clustering the features and using multiple classes to describe the distribution is proposed here, with experimental results showing that this technique can achieve significant improvement when automatically dealing with many font types.

7.2 Document Processing and Analysis

Document processing is a broad field of image processing concerned with the representation, analysis, storage and transmission of document images. Primarily, this involves converting a document image acquired from a physical source into a symbolic form which is easily stored digitally, and extracting some form

of useful information from this representation. In this context, a document image is considered to be one which contains primarily symbolic objects such as words, numbers, lines and graphs, which is usually of high contrast for ease of reading [183]. Examples of document images include envelopes, letters, printed articles, newspapers, forms, sheet music, maps and technical drawings.

The field of document processing can be divided into two broad stages, *document analysis* and *document understanding* [184]. The aim of document analysis is to break down the document image into its physical regions, and convert each such region into the appropriate structural form. Document understanding is then used to deduce the overall meaning of the document, in a manner re-assembling the parts obtained from the original analysis into a logical form. A document can thus be thought of as having two distinct forms

Physical (geometric) structure, obtained by document analysis, consists of the primitive elements of a document image, such as of blocks of text, which are in turn made up of words, which are made up of individual characters.

Logical structure, obtained by document understanding, breaks the document into divisions based on its content, such as sections, subsections and paragraphs.

With these definitions, document understanding can be thought of as the process of mapping the geometric structure of a document into a logical form [184]. Such research is beyond the scope of this work, and will not be presented here.

The field of document analysis has many avenues of active research, including de-noising and binarisation of document images, determining and correcting the skew angle, segmentation into physical regions, finding text in cluttered images, extraction of data from printed forms, OCR and many others. The following sections provide a brief review of some of these topics.

7.2.1 Document Segmentation

Document segmentation is the task of dividing a document image into its physical regions, such as text areas like paragraphs, headings, images, graphs and captions. Although this process is similar in many ways to the segmentation of other images, the unique properties of documents mean that many specialised techniques have been developed to improve the segmentation performance. Since the main goal of document image segmentation is to produce a set of regions for further processing, for example OCR, the output of the segmentation process should contain regions of similar symbolic content, not necessarily having a completely uniform or homogenous appearance [185].

Document segmentation techniques can generally be classified into one of two separate approaches, *top-down* or *bottom-up*. Using the top-down approach, the entire image is analysed in order to split it into multiple regions based on the properties of these regions. This splitting operation is recursively repeated until the final segments are obtained.The two most common top down document segmentation techniques use run-length smoothing and projection profiles to determine the appropriate splitting points at each step [186, 187, 188, 189]. Top-down methods of document segmentation are often layout dependent, and attempt to segment a page based on knowledge of areas such as titles, column positions and paragraph breaks. Non-rectangular regions are also often troublesome when using this approach, as it makes region splitting difficult to correctly detect and describe. The primary advantage of the top-down methods for document segmentation is their speed, which is usually considerably faster than other techniques, since they generally do not require the extraction of individual components.

A bottom-up approach proceeds in the opposite manner, first detecting each individual element of the document image, then using grouping algorithms to recursively merge such components into progressively bigger structures until eventually the final segmentation structure is determined. O'Gorman has proposed such a

technique which defines the relationships between components in terms of polar coordinates, ie distance and direction. Segmentation is then achieved by finding the k nearest neighbour pair amongst the entire set of components, and using this information to estimate text orientation and spacing parameters [190]. Tsujimoto and Asada use a run-length representation of the document image to extract connected components which are then merged to form image segments [191]. Such a representation allows for a more efficient extraction of these components resulting in considerable computational savings. Simon *et. al.* have proposed a unique bottom-up approach to document segmentation which attempts to incorporate knowledge of the document layout when grouping components. In this way, individual components (characters) are grouped to form words, which are then grouped into lines, paragraphs, columns and pages [192]. Using this knowledge of document structure allows for a more efficient implementation of the grouping algorithms, as the components for which distances must be compared is limited to those in a specific region. Bottom-up methods are, in general, more robust in terms of document structure, allowing for the segmentation of non-rectangular objects as well as documents which do not have a traditional layout. However, compared to top-down techniques, the computational time is usually significantly greater. In addition, as the performance of these algorithms is largely dependent upon the correct identification of individual components, the segmentation of degraded and/or noisy documents with textured backgrounds is often not possible.

A number of authors have also proposed document segmentation techniques which incorporate aspects of both the top-down and bottom-up approaches to document segmentation. Kruatrachue and Suthaphan have proposed such a method which uses edge following and a sliding window to detect the initial regions of the image, which is similar to many top-down algorithms [185]. Individual elements of each block are then extracted to further segment multiple columns, images and other objects, in a manner similar to that used in bottom-up approaches.

More recently, document image segmentation techniques which are neither top-

down nor bottom-up have emerged, typically using the properties of regions of the document to perform segmentation into categories such as text, images, lines and others. Such a technique is presented by Etemad *et. al.* [193]. Using textural features of the document image, it is segmented using fuzzy membership assignments into three classes, text, images and graphics. These segmented regions can then be verified and further processed using more traditional document segmentation techniques depending on their type. The primary advantage of this approach is that it is invariant to the layout of the document, and can successfully segment areas which overlap in some manner, a case which can often cause failure using both top-down and bottom-up techniques. By using textural features, information regarding the nature of each component rather than only its size and position can be obtained.

7.2.2 Text Localisation

In a number of applications, full segmentation of an image is neither possible nor desirable. In these cases, it is necessary only to extract small amounts of text from the image, in order to obtain pertinent information. Examples of such applications include database retrieval of arbitrary images which may contain text, automatic extraction of text information, such as captions, from video streams, and finding small amounts of text in complicated documents.

One approach to detecting text in an image is to regard it as a two class segmentation problem. Strouthopoulos and Papamarkos have used the statistical properties of 3×3 blocks of a binary image to perform such a task, with PCA and a neural network used to give the final detected text regions [194]. A similar techniques for determining likely regions of text in an image have been proposed by Wu *et. al.,* who use texture features extracted from the outputs of a bank of nine Gabor-like filters at three different resolutions [195]. On all but the most simple documents, using such features will result in many false positives,

as regions of images which contain many edges are also likely to be detected. To provide greater accuracy, further stages of processing are necessary. In these stages, *strokes* are extracted from those regions classified as likely to contain text, and connected to form *chips*, which correspond to strings of text in the original image. Finally, the chips detected at each of the three resolutions are merged and refined to give the final results. Other examples of using texture features to find text in images can be found in [196, 197].

Clark and Mirmehdi have extended this form of text detection in order to find regions which may not be oriented correctly, or be too small for accurate localisation by other means [198]. Four separate statistical features are calculated over varying sized circular regions of the image, with the results used for the classification of each pixel as text or non-text via a artificial neural network.

7.2.3 Form Analysis

The extraction and processing of information contained in printed forms is another task of great importance in the field of document processing. Until recently, the vast majority of form processing has been done manually, with human operators performing all of the associated tasks up to and including data entry. To reduce the time and effort required for this task, a large amount of research has been undertaken in the fields of form identification, field location and data extraction.

Given the location of fields within a document image, the extraction of the raw data contained within those fields is a relatively trivial task, although interpretation of this data may be more difficult in many cases. The biggest problem, therefore, is the accurate localisation of each field in the form. Even when dealing with known form layouts, this can often be a difficult problem since the same form may appear very different depending on how it was printed and acquired, may be at a different scale than expected, and may not be oriented correctly [199].

A number of form analysis algorithms have been proposed in the literature to perform the following tasks [199, 200, 201, 202]:

- Determine the exact locations of fields for extraction of data.
- Automatic creation of a form layout model for a previously unknown form.
- Verification or classification of form images.

In order to perform each of these tasks, it is necessary to locate the fields within a given image. In general, such fields fall into one of four categories: rectangular, underline, "tooth" or character cell structure, and checkbox [201]. Rectangular fields, the most common of the four, consist of a fully enclosed rectangular area bordered on all sides by straight lines, either solid, dashed or dotted. Underlines are simply a straight line, again either solid or otherwise, upon which data is entered. The "tooth" structure is another common and unique form field, consisting of a line or rectangular field with part lines dividing the region into character cells into which a single character is usually entered. Finally, checkboxes are small enclosed areas, usually either square or circular in shape, which are usually ticked, crossed or otherwise marked to indicate a boolean response. Although other methods of entering data on a form exist, such as the circling of one or more options from a list, the design of these is so arbitrary as to make their detection almost impossible without some form of prior knowledge.

Apart from the lone case of circular checkboxes, each of these fields is comprised of straight lines. The focus of much form analysis research, therefore, is concerned with the detection of these lines, and determining whether those found are likely to form part of one of the above structures [199]. Approaches used to detect lines in document images include the Hough transform and projection profile based techniques [203, 204]. Although effective at detecting solid lines, such techniques have difficulties when faced with broken lines, which occur frequently in forms due to acquisition noise and the actual writing on the forms. Lines which are not

horizontal or vertical are also not correctly identified by such techniques in many cases. Zheng *et. al.* have proposed a novel technique for detecting such lines, using a directional single-connected chain and an algorithm for merging broken lines [205]. An alternative approach is suggested by Hori and Doermann, who use an approach known as *box driven reasoning* to detect rectangular regions in form images [200]. Another approach which is highly robust which respect to partially incomplete lines attempts to form rectangular regions from all possible sets of line segments, based on a number of constraints between rectangles and edges [199].

Form detection is another application which uses form analysis techniques. By determining the presence of the above-mentioned form structures, as well as the amount and locations of printed text, a likelihood of the document being a form can be calculated [206].

7.2.4 OCR

Optical character recognition, or OCR, is perhaps the largest field of document analysis, and has received a significant amount of attention over the last three decades. Primarily, OCR can be defined as the task of converting an image of text, either printed or handwritten, into its equivalent electronic representation, usually represented by some form of character set such as ASCII. More complex systems are also able to successfully extract information such as the font, size and correct positioning of the text from printed documents.

In order to accomplish the recognition of individual characters, a number of different techniques have been proposed in the literature. The earliest work in the field is generally attributed to Tauschek, who first obtained a patent for his 'reading machine' in 1929 [207]. The premise behind this simple machine was an optical mask, whereby light was passed through a series of mechanical masks, with exact matches failing to transmit any light at all. Similar methods are

still in existence today, and are widely known as 'template matching' methods. Variations on this approach include taking one dimensional projections of the character image, and the so-called 'peephole' methods [208, 209].

Since these early methods were introduced, the literature has shown many hundreds of different techniques for performing OCR on both printed and handwritten characters. A few of these general approaches include using the autocorrelation function of the image [210], the Karhunen-Loeve expansion [211], the Fourier series expansion [212], chain coding [213] and numerous approaches which attempt to represent the structural properties of the characters [214, 215].

A detailed explanation of these and other algorithms is beyond the cope of this thesis. For more information, the reader is referred to [181, 182, 216, 217].

7.3 Script and Language Recognition

Although a large number of OCR techniques have been developed in recent years, almost all existing work in this field assumes that the script and/or language of the document to be processed is known. Although it would be certainly possible to train and OCR system with characters from many languages to obtain a form of language independence, the performance of such a classifier would naturally be lower than one trained solely with the script and/or language of interest. Using specialised classifiers for each language is also advantageous in that it allows for the introduction of language and script specific knowledge when performing other required tasks such as document segmentation character separation. Using such specialised classifiers in a multi-lingual environment requires an automated method of determining the script and language of a document.

A number of different techniques for determining the script of a document have been proposed in the literature. Spitz has proposed a system which relies on specific, well-defined pixel structures for script identification [218]. Such features

include locations and numbers of upward concavities in the script image, optical density of text sections, and the frequency and combination of relative character heights. This approach has been shown to be successful at distinguishing between a small number of broad script types (Latin-based, Korean, Chinese and Japanese) and moderately effective at determining the individual language in which the text is written. Results when using a wider variety of script types (Cyrillic, Greek, Hebrew, etc) are not presented, nor is any attempt made to define the conditions for an unknown script type.

Pre-processing of document images is required before this technique is applied. The purpose of this is to form a binary image, that is, an image composed entirely of white and black pixels only. By convention, white pixels are considered background and black pixels are considered text. Following this, connected components [219] are extracted from the binary representation. For each component extracted in this way, information such as the position, bounding box, and lists of pixels runs is stored. Representing the image in this way provides an efficient data structure on which to perform image processing operations, such as the calculation of upward concavities and optical density required in the latter stages of the process. For many script types, calculating connected components will also separate individual characters, although this is not always true in the general case. Further work has attempted to address this problem by segmenting individual connected components at this stage [220]. This is required primarily for languages that possess a high degree of connectivity between characters, such as types of Indian script, and certain fonts such as italics which enhance connectivity of consecutive characters.

Suen *et. al.* also apply two extra segmentation algorithms at this point, in order to remove extremely large connected components and noise [221]. Large components are considered to be those with bounding boxes more than five times the average size. It is thought that these correspond to non-textual regions of the input image and can safely be discarded. Any bounding boxes with dimen-

sions smaller than the text font stroke are considered noise, and also removed. Practical experiments have shown some problems when using these techniques, as important image features such as the dots on 'i' characters and accent marks are erroneously removed. If character segmentation is not accurate entire lines of text may also be removed, as they are considered to be a single large connected component.

Before feature extraction, it is necessary to define various regions within a line of text. Four horizontal lines define the boundaries of three significant zones on each text line. These are

topline: The absolute highest point of the text region. Typically corresponds to the height of the largest characters in the script, for example 'A'.

x-height: The height of the smaller characters of the script, such as 'a' and 'e'. This line is not defined for a number of scripts such as Chinese.

baseline: The lowest point of the majority of the characters within a script, excluding those which are classified as *descenders*.

bottomline: The lowest point of any character within the script. Examples from Latin-based languages are the 'g' and 'q' characters.

These four boundary lines effectively divide a line of text into three separate zones. The area between the bottom and the baseline is known as the descender zone, between the top and x-height the ascender zone, and the region within the baseline and x-height is the x-zone. Figure 7.1 shows the locations of each of these lines and regions for an Latin text sample.

Calculating the positions of these lines is performed using vertical projection profiles of the connected components. By projecting the lowest position, top position, and pixels for each connected component, the positions of each line are determined as follows. The peak in the lowest position profile is taken as

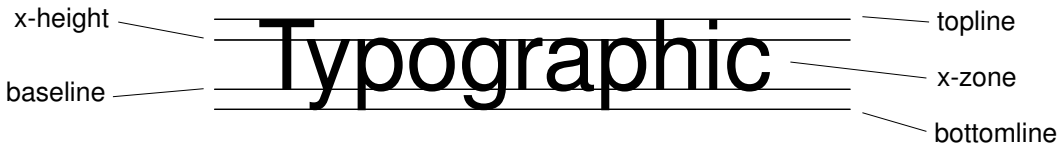


Figure 7.1: Commonly labelled text regions for an Latin script sample.

the baseline position, since the majority of characters in any known language will have their lowest point here. Searching upwards from this point, the peak in the top position profile is then labelled as the x-height, although this may lead to inaccurate x-line positioning when lines of text with a large number of capital letters and/or punctuation symbols are present [218]. The positions of the top and bottom lines are found simply by searching for the highest and lowest projected positions respectively. Having determined the positions of these lines, they are then used as a reference point for all location information for individual connected components.

The primary feature used in the script recognition algorithm of Spitz is the location and frequency of upward concavities in the text image. An upward concavity is present at a particular location if two runs of black pixels appear on a single scan line of the image, and there exists a run on the line below which spans the distance between these two runs. Once found, the position of each upward concavity in relation to the baseline of the character is noted, and a histogram of these positions for the entire document constructed. Analysis of such histograms for Latin-based languages shows a distinctly bi-model distribution, with the majority of upward concavities occurring either slightly above the baseline or slightly below the x-height line. In contrast to this, Han-based scripts exhibit a much more uniform distribution, with the modal value typically evenly spaced between the baseline and x-height. Using this information, it is possible to accurately distinguish Latin-based and Han-base scripts using a simple measure of variance. No histograms were provided for other script types such as Greek, Hebrew or Cyrillic, so it is unknown how such a method will perform for these script types.

In constant use this method has never been observed to incorrectly classify Latin or Han-based scripts [222].

Once a determination on the broad class of script has been made, identification of individual languages is performed using different techniques. For the Han-based languages, pixel density for each character is the sole feature used for discrimination. Spitz demonstrates that Japanese text has, on average, a much lower modal value of pixel density than do the other languages tested (Chinese and Korean). Chinese language documents were found to show a much higher modal value, while Korean exhibited a unique bi-modal distribution. Using this information, it is possible to classify an unknown sample into one of these three classes with excellent accuracy. Using only six lines of input text, the algorithm was shown to perform with 100% accuracy over a wide range of document images.

Language classification for Latin-based languages is performed in an entirely different manner. Since most characters are shared by all of these languages, simple metrics such as pixel density will not be able to segment them effectively. Some languages do contain unique characters (umlauts in German, accents and diacritics in French), however do to the variation in typesetting styles as well as the possibility of alternative representations, these are also inappropriate for accurate language classification.

A common approach for separation of Latin-based languages is to classify each character into a broad character class represented by a single character code [218, 221]. One commonly used mapping of characters to character codes based upon the number of connected components, and whether or not the character rises above the x-height line or below the baseline is shown in table 7.1. Characters which do not fall into any of these classes are considered unknown.

Statistical analysis shows that certain shape code combination are much more frequent in certain languages than others. For example, the shape code AAx is extremely common in the English language, as it corresponds to both ‘The’ and

Character shape code	Characters
A	A-Z, bdfhklt, 0-9, #$&()/[]@{}
x	acemnorsuvwxz
i	iàáâèéêîòóôùúûñ
g	gpqyç
j	j
U	äëïöüÄËÏÖÜ

Table 7.1: Character shape codes and the characters they represent

'the'. Other languages also possess shape code chains that exhibit an unusually high frequency of occurrence, such as 'die', 'der' and 'das' in German, and 'la' and 'le' in French. By choosing the top five Word shape tokens (WST's) for each language to be discriminated, a collection of the most common word shapes can be created. With the expected overlap between languages, this leads to a set of only 24 unique WSTs which require detection. By using the relative frequency of each of these WSTs as features for classification and training the classifier with a large selection of sample documents, this technique has been shown to provide good results. Classification of a large selection of printed text documents in 23 languages shows a correct classification rate of over 90% overall, with recognition rates for individual languages ranging from 75% to 100%.

7.4 Texture Analysis for Script Recognition

The work presented in the previous section has shown excellent results in identifying a limited number of script types in ideal conditions. In practice, however, such techniques have a number of disadvantages which in many cases make the identification of the script difficult. The detection of upward concavities in an image is highly susceptible to noise and image quality, with poor quality and noisy images having high variances in these attributes. Experiments conducted on noisy, low-resolution or degraded document images has shown that classification performance drops to below 70% for only two script classes of Latin-based

and Han. The second disadvantage of the technique proposed by Spitz is that it cannot effectively discriminate between scripts with similar character shapes, such as Greek, Cyrillic, and Latin-based scripts, even though such scripts are easily visually distinguished by untrained observers.

Determination of script type from individual characters is also possible using OCR technology. This approach, as well as others which rely on the extraction of connected components, requires accurate segmentation of characters before their application, a task which becomes difficult for noisy, low resolution or degraded images. Additionally, certain script types, for example Devanagari, do not lend themselves well to character segmentation, and require special processing. This presents a paradox in that to extract the characters for script identification, the script in some cases must already be known.

Using global image characteristics to recognise the script type of a document image overcomes many of these limitations. Because it is not necessary to extract individual characters, no script-dependent processing is required [223]. The effects of noise, image quality and resolution are also limited to the extent that it impairs the visual appearance of a sample. In most cases the script of the document can still be readily pre-attentively determined by a human observer regardless of such factors, indicating that the overall texture of the image is maintained. For these reasons, texture analysis appears to be a good choice for the problem of script identification from document images.

Previous work in the use of texture analysis has been limited to the use of Gabor filter features [77, 224]. While this work has shown that texture can provide a good indication of the script type of a document image, other texture features may perform better in this context, and the aim of this section is to investigate this theory.

7.4.1 Pre-processing of Images

After segmentation, blocks of text from typical document images are not good candidates for the extraction of texture features. Varying degrees of contrast in greyscale images, the presence of skew and noise could all potentially affect such features, leading to misclassification in many cases. Additionally, the large areas of whitespace, unequal character, word and line spacings and line heights can also have a significant effect on these features. In order to reduce the impact of these factors, the text blocks from which texture features are to be extracted must undergo a significant amount of pre-processing. The individual steps which are performed in this stage are:

- Binarisation
- Deskewing
- Block normalisation

The following sections outline each of these tasks in detail, showing examples of the results after each stage.

7.4.2 Binarisation of Document Images

Many of the algorithms used in document analysis rely on identifying connected components, that is, groups of pixels which are connected to form a single entity. In order to accomplish this it is necessary for an image to first be binarised. Although document images are typically produced with a high level of contrast for ease of reading, scanning artifacts, noise, paper defects, coloured regions and other image characteristics can sometimes make this a non-trivial task, with many possible solutions presented in the literature.

In general, binarisation techniques rely on a threshold value to segment the image into two classes, known as the background (usually white) and foreground (black). Such a threshold can be either fixed or adaptive, although fixed thresholds are of limited use in most practical applications. Additionally, thresholding can by applied globally, whereby a single value is used for the entire image, or locally, where a separate threshold is calculated for different regions in the image. While global thresholding has the advantage of simplicity, and works well for a large number of image types, it will often fail on images with varying or textured backgrounds, or documents with low frequency components which are commonly introduced by some document acquisition devices. Calculating local thresholds can overcome many of these problems, but is significantly more complex and is computationally many times more costly.

Otsu has proposed a method of calculating a global threshold which is based on optimising the inter-class separation of the resulting two class image [225]. Given a threshold value τ, a separability measure $\sigma_B^2(\tau)$ can be calculated which represents this separation factor. By choosing the value of τ which maximises this function, an optimal threshold for the given image can be found. While this method of binarisation achieves excellent results on images with a constant background, being global it often fails when faced with documents containing distinct regions or slowly varying backgrounds. Kapur [226] and Kittler [227] have also proposed algorithms for calculating global thresholds based on different separability measures.

Local adaptive binarisation methods compute a threshold for each pixel in the image based on the properties of its local neighbourhood. Lui *et. al.* have proposed such a method using grey scale and run length histogram analysis in a method they call *object attribute thresholding* [228]. Global techniques are first used to create a set of approximate thresholds, which are further refined for each local neighbourhood. Yang uses a statistical measurement called the *largest static state difference* to define a local threshold, which tracks changes in the statistical

signal pattern [229]. Another adaptive method proposed by Sauvola *et. al.* calculates a local threshold based on the observed properties of local regions [230]. Using this technique, regions containing text, graphics and textures are treated in different manners.

A common failure many binarisation techniques is poor performance in the presence of textured background, especially those with high contrast. Such situations occur frequently when colour documents are converted to greyscale images, and the distinction between the colours of similar intensity is lost. This is also often a problem when attempting to analyse watermarked documents such as cheques and forms, which can have a noticeable underlying texture. Liu and Srihari present a technique to overcome this problem based on textural features a multiple thresholds [231]. Otsu's method is iteratively used to produce a set of candidate thresholds for each region, followed by the extraction of texture features from the images generated by these threshold values. Based on these features, the optimal threshold is chosen and the final binarised image obtained. Impressive results using this technique have been obtained on a wide range of document images with varying and textured backgrounds.

For the purposes of this evaluation, all of the images used are of high contrast with no background shading effects. Because of this, a global thresholding approach provides an adequate means of binarisation, and the method proposed by Otsu is used for this purpose.

7.4.3 Skew Detection

A significant portion of the work presented in this section was compiled in conjunction with Scott Lowther, and his assistance and contributions are gratefully acknowledged.

Many techniques for document analysis require that the image be correctly

aligned such that the lines of text are horizontal. Calculating projection profiles, for example, requires knowledge of the skew angle of the image to a high precision in order to obtain an accurate result. In practical situations this is often not the case, as scanning errors, different page layouts or even deliberate skewing of text can result in misalignment. In order to correct this, it is often necessary to accurately determine the skew angle of a document image or of a specific region of the image, and for this purpose a number of techniques have been presented in the literature.

Postl [232] found that the maximum valued position in the Fourier spectrum of a document image corresponds to the angle of skew. However, this finding was limited to those documents that contained only a single line spacing, thus the peak was strongly localised around a single point. When variant line spacings are introduced, a series of Fourier spectrum maxima are created in a line that extends from the origin. Also evident is a sub-dominant line that lies at 90° to the dominant line. This is due to character and word spacings, and the strength of such a line varies with changes in language and script type. Peake and Tan expand on this method, breaking the document image into a number of small blocks, and calculating the dominant direction of each such block by finding the Fourier spectrum maxima [233]. These maximum values are then combined over all such blocks, and a histogram formed. After smoothing, the maximum value of this histogram is chosen as the approximate skew angle. The exact skew angle is then calculated by taking the average of all values within a specified range of this approximate. The authors claim that this technique is invariant to document layout, and will still function even in the presence of images and other noise.

Other techniques for the estimation of the skew angle have been proposed. Chaudhuri proposes an accurate method based on the cross-correlation of vertical image slices [234]. Whilst this method provides excellent accuracy, it has not proven robust to the presence of large images or non-text regions in the document. Numerous other techniques for automatic skew detection also exist [235, 236].

An evaluation of many such techniques was carried out, and the method proposed by Peake and Tan [233] found to provide superior results for a large variety of document images. However, a number of modifications have been made to this technique, in order to provide increased accuracy of the final detected angle, and further remove the effects of noise and/or non-text regions of the document.

The Fourier spectrum of a text-based document will contain a dominant line in the direction of the skew, due to the effects of line spacings. Whilst Peake and Tan simply take maximum values of the spectrum and use these as an estimation of the skew angle, experimental evidence has suggested that better accuracy can be found by attempting to determine the angle of this dominant line. Since by definition the line must pass through the origin, this task is somewhat simplified. By using the Radon transform, the orientation can be accurately determined by simply choosing the maximum value [237]. The algorithm used for skew determination is presented below.

1. Image Subdivision: the image is divided into blocks of size $N \times N$
2. Fourier Analysis: the Fourier spectrum is computed for each block (using FFT) and represented with the origin at the centre.
3. Mean Fourier Calculation: The Fourier transform for all blocks is averaged together to form a single Fourier block for applying the Radon transform. An example of such a block is shown in figure 7.2.
4. Apply a Mask: A doughnut shaped mask is applied to the mean Fourier block to remove DC components and to reduce extra weighting of values on the diagonals in the Fourier block.
5. Approximate Angle Calculation: Apply a Radon transform to the mean Fourier block in the range $[-90, 90]$ in angle increments of 1°. The peak of the Radon transformation is taken as the approximate angle of skew.

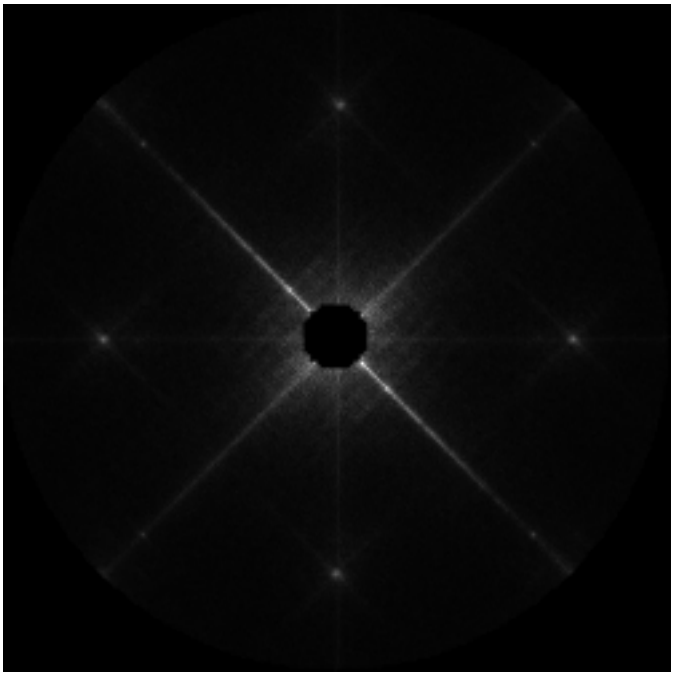


Figure 7.2: Example of the Fourier block representation of a typical document image, showing a dominant line at approximately 45° with a subdominant line approximately perpendicular to this.

6. Refined Angle Calculation: Apply a second Radon transform to the original mean Fourier block in the range $\pm t°$ of the approximate skew angle, in smaller angle increments. A suitable value for t is 1° based on the resolution of the approximate angle calculation above. The peak of this Radon transformation is taken as the final value of skew angle.

A limitation of the Fourier block in the proposed technique is its angular resolution. The angular resolution governs the accuracy of the detected skew angle. A higher angular resolution will allow the technique to detect a skew angle nearer to the true skew angle. Considering document images scanned at 150dpi, written in 12pt font with single line spacing, the length between text lines is approximately 30 pixels. Given a Fourier block size of 256x256 with a resolution of 1/256 cycles/pixel and a peak line spacing frequency of 1/30 cycles/pixel, then the peak value in each Fourier block due to the line spacing will lie approximately $256/30 \approx 9$ pixels from the origin. The angular resolution of the peak value of

the Fourier block is approximated by

$$|d\theta| = \left| \frac{\partial \theta}{\partial u} du + \frac{\partial \theta}{\partial v} dv \right| \tag{7.1}$$

where θ is the skew angle, and u and v represent the coordinate system in the Fourier domain. Using $\theta = \tan^{-1}\left(\frac{v}{u}\right)$, this can be re-written as

$$|d\theta| = \left| -\frac{\sin(\theta)}{r} du + \frac{\cos(\theta)}{r} dv \right| \tag{7.2}$$

where $r = \sqrt{u^2 + v^2}$.

Therefore, at 9 pixels from the origin of the Fourier block along the vertical axis (r=9/256, $\theta = 0°$, $dx = 0$, $dy = 1/256$), we have $d\theta \approx 6.4°$. Similarly, on the 45° diagonal to the vertical axis ($r = 30/256$, $\theta = 45°$, $dx = 1/512$, $dy = 1/512$) a value of $d\theta \approx 4.5°$ is obtained. These values give angular resolution in the range $[4.5° \ldots 6.4°]$ for finding the peak value of the Fourier blocks corresponding to the contribution of the line spacing in the original document.

From visual inspection of the Fourier block of figure 7.2 it can be seen that the dominant line extends much further than the peak value caused by particular line spacings. This dominant line is predominantly caused by harmonics of the inter line spacings and is dependant on the line spacings and the number of lines of text in the document. The existence of the dominant line aids in achieving a finer angular resolution since points further from the origin than the peak Fourier value will contribute to the detection of the skew angle. If the dominant line contributes up to four times the length of the peak value (36 pixels from the origin), the angular resolution is calculated to be in the range $[1.1°, 1.6°]$. The results shown in Table 1 indicate that the practical angular resolution obtained by the use of our method is less than 0.25. This higher angular resolution can be attributed to a higher contribution of the dominant line and also the bilinear interpolation involved with the use of the Radon transform.

Tests have been conducted using both our technique, the Averaged Block Directional Spectrum (ABDS) technique, and the technique in [233] across a range

Figure 7.3: Results of testing the two skew determination techniques on binary document images. The graph shows the percentage of images for which error in skew determination was found within the given skew error thresholds.

of documents and skew angles. The test set consisted of 94 document images randomly rotated 20 times each in the range $[-45°, 45°]$ for a total of 1880 individual tests. The documents used consisted of various invoices, letters and billing statements written predominantly in Latin script and containing varying levels of graphical content and line spacing. Each of the 95 documents was scanned at 150dpi and skew corrected manually before tests were conducted. Tests were conducted on the original document images and binary copies with a suitable binary threshold selected using Otsu's binarisation technique [225]. The results from testing on binary and greyscale documents are shown in figure 7.3 and figure respectively. For clarification, accuracies on binary images for key skew error thresholds are also presented in table 7.2. The actual errors in detected skew angle for images which were not detected correctly are shown in figures 7.5 and 7.6 for binary and greyscale images respectively.

Tests were also conducted to evaluate the robustness of the ABDS technique to varying levels of graphical content. Test images were separated into classes

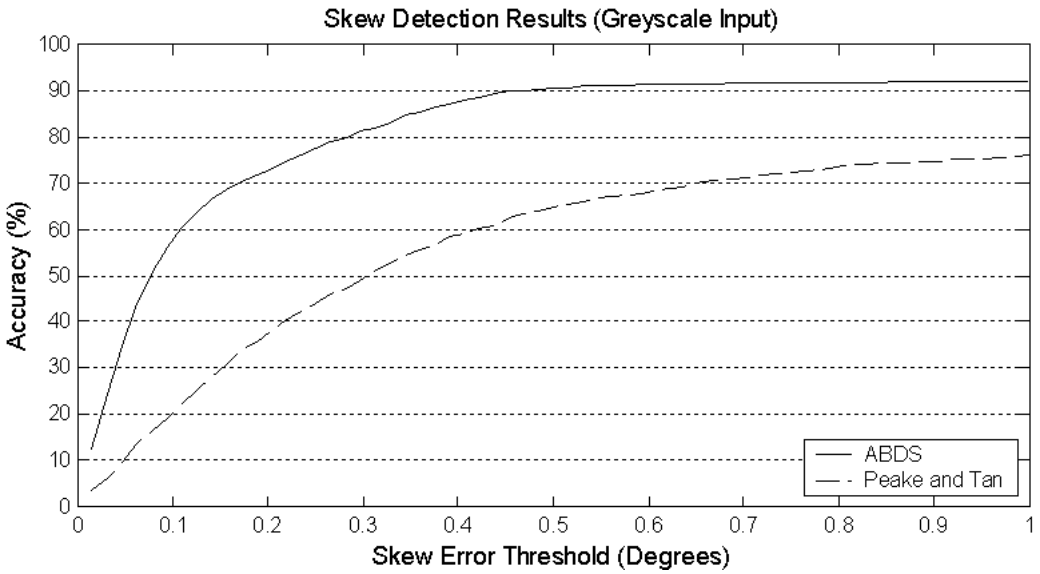


Figure 7.4: Results of testing the two skew determination techniques on greyscale document images. The graph shows the percentage of images for which error in skew determination was found.

Skew Detection Technique	**Error Threshold**			
	$\leq 1°$	$\leq 0.5°$	$\leq 0.25°$	$\leq 0.125°$
ABDS	96.6%	96.5%	96.4%	93.45%
Peake and Tan	74.15%	65.45%	48.5%	29.85%

Table 7.2: Results of testing the two skew determination techniques on binary document images against different angle accuracy thresholds. The percentages shown in the table correspond to the percentage of documents whose skew angle was correctly determined within the given error threshold.

according to the proportion of text pixels compared to non-text pixels in the binary copies of the original document images. Graphical content in the test images consisted mainly of logos, tables, graphics and borders oriented in the same direction as document text. Text pixels were found automatically using the technique presented in Section 1.2.3. Once the test images were separated, each binary image was randomly rotated 10 times in the range $[-45°, 45°]$ and the corresponding skew determined using the ABDS technique. Accuracies were obtained from the proportion of test images for which skew determination error was within 0.25. The results from testing can be seen in table 7.3.

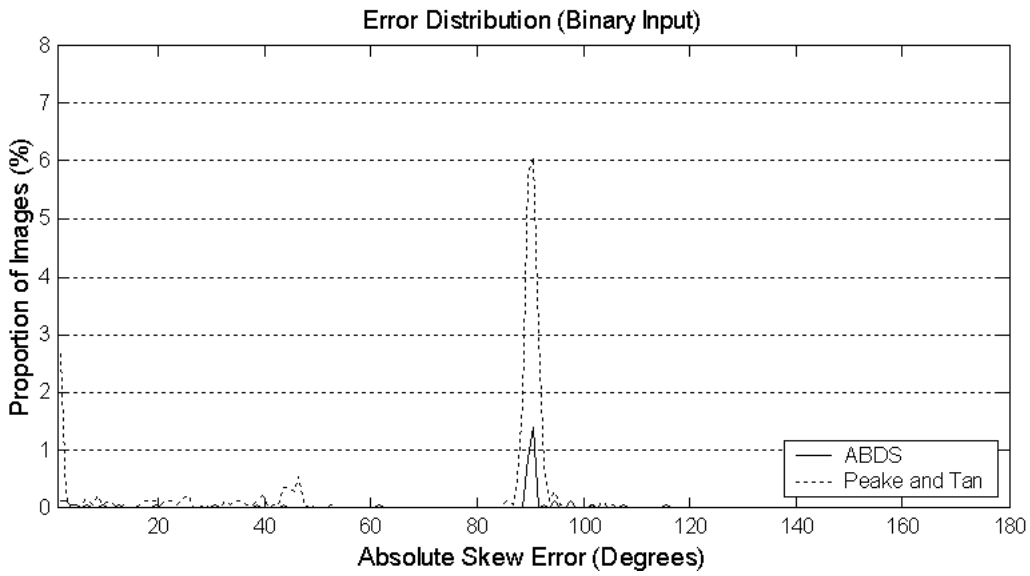


Figure 7.5: Skew error distribution for absolute skew error greater than 1° on binary images.

Text Level (%)	0-20	20-40	40-60	60-80	80-100
Accuracy (%)	94.29	93.08	99.33	100	100

Table 7.3: Accuracies obtained from testing on images with varying levels of graphical content using the ABDS technique. Percentages are given for skew determination error within 0.25°.

As can be seen, the proposed technique has performed very well, attaining an accuracy of 96.4% when determining skew within 0.25 of the true skew angle. The majority of errors occurred from the detection of the line at 90° to the true skew angle caused from character spacings, word spacings and graphical elements. Simple techniques can be employed to determine the orientation of a document in these instances.

Table 7.3 shows that the ABDS technique is significantly robust to the presence of graphics and other non-textual elements in document images. While the ABDS technique produces more accurate results when there are a high proportion of textual regions in a document image, tests indicate that accuracies well above 90% for skew error within 0.25° can be obtained even if there is very little text

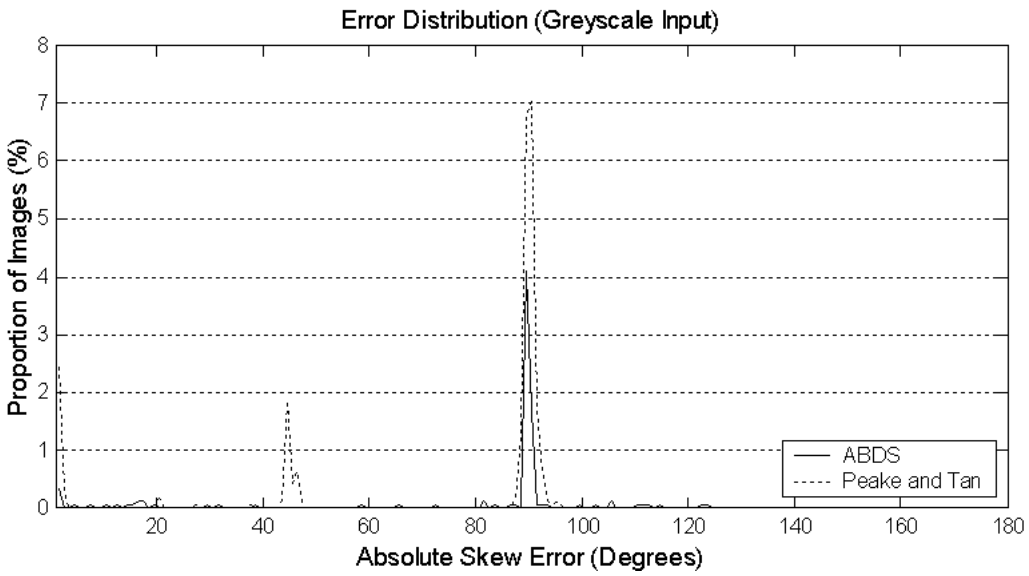


Figure 7.6: Skew error distribution for absolute skew error greater than 1° on greyscale images.

present in the images. High accuracies in low text situations can be attributed to non-textual components in document images having similar orientation to that of textual components, therefore contributing also to the dominant line in the Fourier spectrum.

As can be seen from the presented results, this technique is effective at accurately detecting the skew angle of document images with varying amounts of text, with a typical error of less than 1°. Once this angle is known, it is possible to deskew the image via simple image manipulation such that it is correctly aligned. This enables segmentation of the image to be performed accurately and text regions extracted. For the purposes of this work such segmentation is performed manually, although any number of algorithms could be used for this purpose [185, 238, 239, 240, 241].

Segments extracted in this way from a document image may still require further processing to align each line of text correctly. It is not uncommon for regions to still exhibit a small amount of skew, or to contain artifacts introduced in the

image acquisition process which cause a small degree of curvature to appear in some lines of text, particularly close to the binding edge of scanned pages. Such distortions will cause later stages of the script detection process to fail, and must be addressed.

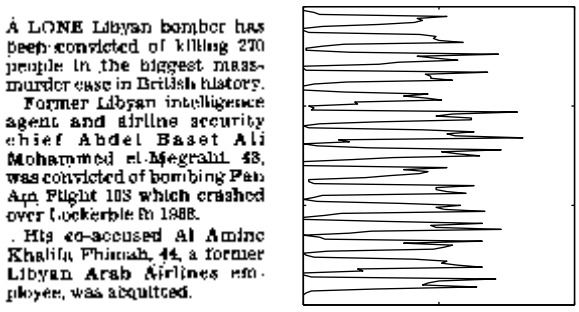
In order to correct this problem, as well as eliminating any small amounts of remaining skew, the technique proposed by Tsuruoka *et. al.* is used [242]. Although this technique was originally developed for the analysis of handwritten script, it can be equally well applied to the somewhat simpler case of printed document By traversing along each line of the text and detecting the lower convex hulls and upper and lower outer-lines, an approximation of the baseline is created. By calculating the derivative of this a approximation and assuming that the regions with high values are indicative of characters which extend below the baseline, a smoothed estimate is generated. Experiments using this technique have shown excellent results, with visually perfect images obtained in all cases.

7.4.4 Normalisation of Text Blocks

Extraction of texture features from a document image requires that the input images exhibit particular properties. The images must be of the same size, resolution, orientation and scale. Line and word spacing, character sizes and heights, and the amount of whitespace surrounding the text, if any, can also affect texture features. In order to minimise the effects of such variations to provide a robust texture estimate, our system attempts to normalise each text region before extracting texture features. This process will also remove text regions that are too small to be characterised adequately by texture features.

An effective algorithm for overcoming these problems and normalising each region of text has been developed, based on the work done by Peake and Tan [224]. After binarisation, deskewing, and segmentation of the document image, a number of operations are performed on each region in order to give it a uniform appearance.

Firstly, horizontal projection profiles are taken for each segment. By detecting valleys in these profiles, the positions of line breaks, as well as the height of each line and line space is calculated, assuming that the text is correctly aligned following deskewing. An example of a typical projection profile obtained in this manner is shown in figure 7.7.



(a) (b)

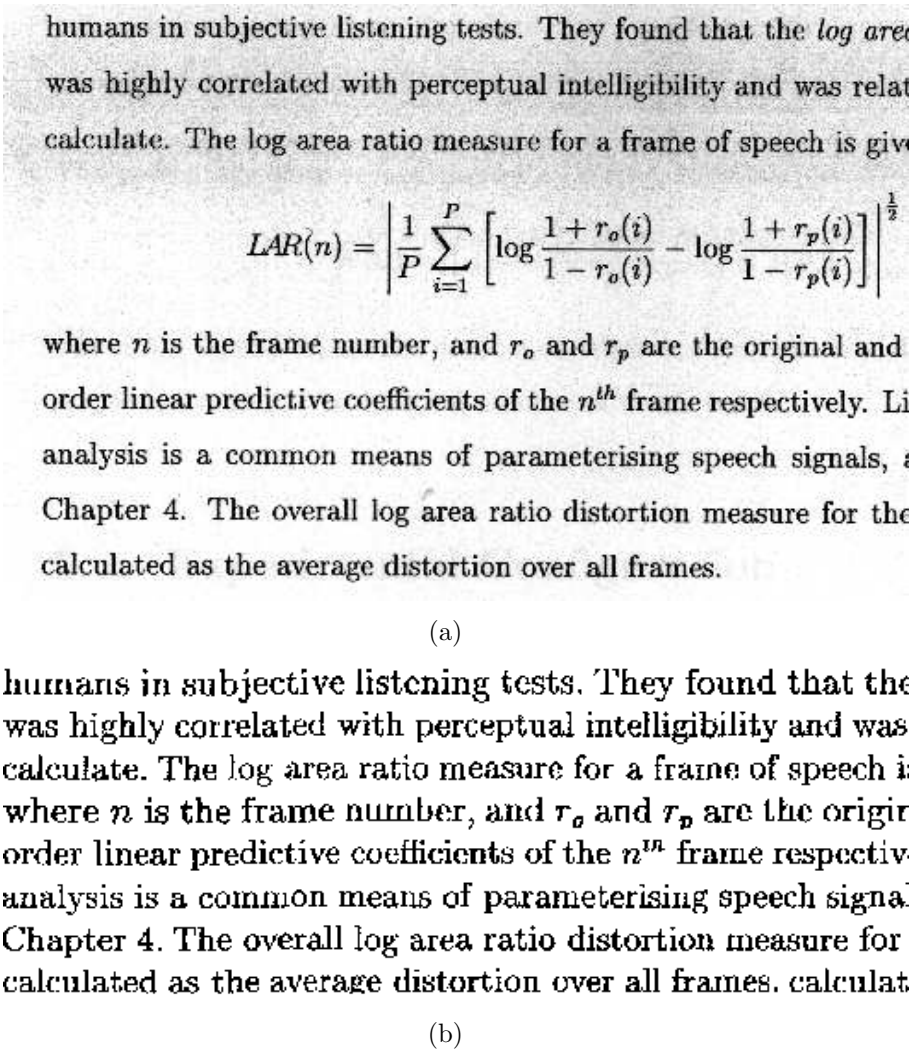
Figure 7.7: Example of projection profile of text segment. (a) Original text, and (b) projection profile.

Having detected the lines of text, the average height of the lines in the region is then calculated, and those that are either significantly larger or smaller than this average are discarded. Investigation of many regions has found that such lines often represent headings, captions, footnotes, or other non-standard text, and as such may skew the resulting texture features if they are retained. The remaining lines are then normalised by the following steps:

Scaling: Each line is scaled to convert it to a standard height.

Normalisation of character and word spacings: Often, modern word processing software expands spaces between words and even characters to completely fill a line of text on a page, leading to irregular and sometimes large areas of whitespace. By traversing the line and ensuring that each space does not exceed a specified distance, this whitespace can be removed.

Padding: After performing the above operations on each line, the length of the longest line is determined, and each of the others padded to extend them


humans in subjective listening tests. They found that the *log are*
was highly correlated with perceptual intelligibility and was relat
calculate. The log area ratio measure for a frame of speech is giv

$$LAR(n) = \left| \frac{1}{P} \sum_{i=1}^{P} \left[\log \frac{1 + r_o(i)}{1 - r_o(i)} - \log \frac{1 + r_p(i)}{1 - r_p(i)} \right] \right|^{\frac{1}{2}}$$

where n is the frame number, and r_o and r_p are the original and
order linear predictive coefficients of the n^{th} frame respectively. Li
analysis is a common means of parameterising speech signals,
Chapter 4. The overall log area ratio distortion measure for the
calculated as the average distortion over all frames.

(a)

humans in subjective listening tests. They found that the
was highly correlated with perceptual intelligibility and was
calculate. The log area ratio measure for a frame of speech i
where n is the frame number, and r_o and r_p are the origir
order linear predictive coefficients of the n^{th} frame respectiv
analysis is a common means of parameterising speech signa
Chapter 4. The overall log area ratio distortion measure for
calculated as the average distortion over all frames. calculat

(b)

Figure 7.8: Example of text normalisation process on an Latin document image. (a) Original image, and (b) normalised block.

to this length to avoid large areas of whitespace at the ends of lines. To accomplish this, the line is repeated until the desired length is achieved. Clearly, for lines which are very short, such repetition may lead to peaks in the resulting spatial frequency spectrum of the final image, and hence lines which do not satisfy a minimum length, expressed as a percentage of the longest line, are simply removed.

Following normalisation, lines must be recombined to construct the final block of text. When performing this stage of processing, it is important that the line

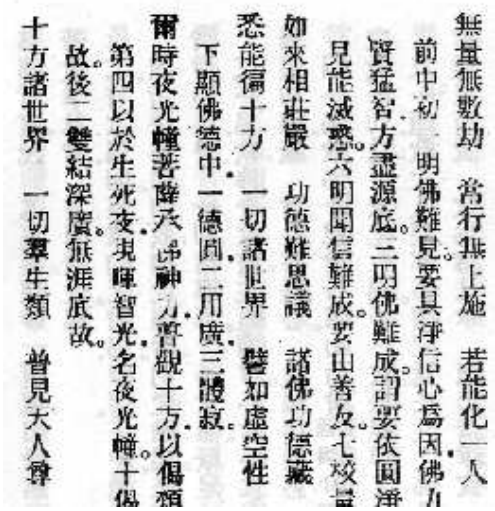
spacings are constant to avoid significant whitespace between lines. Due to differences in the nature of various scripts, analysis of the line is required in order to determine the limits and relative frequencies of character heights. For example, Latin-based scripts comprise mostly of characters of a small height, such as 'a', 'r', and 's', with a small but significant number of characters which protrude above and/or below these heights, for example 'A', 'h', 'p' and 'j'. In contrast to this, almost all characters in other script, for example Chinese, have identical heights. Determination of which class a sample of text belongs to can be made by examining the project profiles of each line. In order to obtain a uniform appearance over all script types, this information is taken into account when normalising the line spacings. To allow for a more uniform appearance, samples of text with uniform or near-uniform character heights are combined using a larger line spacing.

An example of the entire normalisation process applied to a typical text segment is shown in figure 7.8[1]. From this example it can be seen that the original image, which is somewhat noisy and contains many large regions of whitespace, highly variable line spacing and non-standard text in form of equations, is transformed into a block of relatively uniform appearance. Closer inspection reveals the existence of repeated sections, however pre-attentively this is not apparent. The algorithm described above works equally well on all tested scripts and languages, which is clearly and important property for this application. Figure 7.9 shows the results obtained after processing a Chinese document image. Note in this example the increased line spacings due to the equal height of characters in the Chinese script.

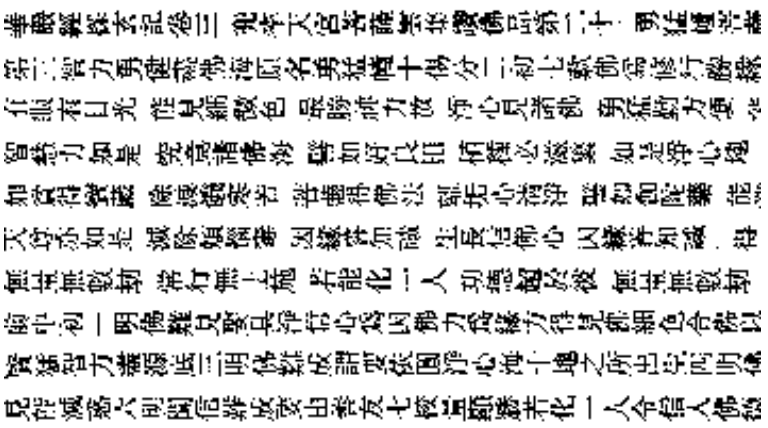
7.4.5 Texture Feature Extraction

From each block of normalised text, the following texture features are evaluated for the purpose of script identification:

[1]The example images shown here have been manually cropped after processing in order to retain a high resolution



(a)



(b)

Figure 7.9: Example of text normalisation process on a Chinese document image. (a) Original image, and (b) normalised block.

Grey-Level Co-occurrence Matrix Features

Grey-level co-occurrence matrices have been used for many years as a means of characterising texture, and although it has been proven that they are insufficient to completely describe such an image, the remain a popular choice for many applications. The construction of the GLCMs for an image, and a number of features commonly extracted from them, is described in section 3.4.3.

Due to the binary nature of the document images from which the features are

extracted, it is possible to optimise the performance of features extracted from the grey-level matrices. Since there are only two grey levels, the matrices will be of size 2×2, meaning that it is possible to fully describe each matrix with only 3 unique parameters due to the diagonal symmetry property. Using these values directly rather than attempting to extract other features more provides more information about the texture, and has experimentally shown to give better results. Using values of $d = \{1, 2\}$ and $\theta = \{0°, 45°, 90°, 135°\}$ leads to a total of 24 features.

Gabor Energy Features

The energy of the output of a bank of Gabor filters has been previously used as features for identifying the script of a document image, with good results shown for a small set of test images [77, 224]. In this work, both even and odd symmetric filters are used, given by

$$g_e(x, y) = \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\} \cos(2\pi u_0(x\cos\theta + y\sin\theta)) \quad (7.3)$$

$$g_o(x, y) = \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\} \sin(2\pi u_0(x\cos\theta + y\sin\theta)) \quad (7.4)$$

where x and y are the spatial coordinates, μ_0 the frequency of the sinusoidal component of the gabor filter, and σ_x and σ_y the frequencies of the Gaussian envelope along the principal axes, with typically $\sigma_x = \sigma_y$. In the experimental results presented in [77], a single value of $\mu_0 = 16$ was used, with 16 orientation values spaced equidistantly between 0 and 2π, giving a total of 16 filters. By combining the energies of the outputs of the even and odd symmetric filters for each such orientation, a feature vector of same dimensionality is created. To obtain rotation invariance, this vector is transformed via the Fourier transform, and the first 4 resulting coefficients used for classification. Since skew detection and correction has been performed on the test images to be used in these experiments, such a transformation is not required, and the features will be used unmodified. By combining the energies of the odd and even symmetric filters

for each resolution and orientation, a total of 16 features are obtained using this method. While these features have shown good performance on a small number of script types [77], using only a single frequency does not provide the necessary discrimination when a large set of scripts and fonts are used. To overcome this, an additional 16 filters with a frequency of $\mu_0 = 8$ are employed, giving a final dimensionality of 32.

Wavelet Energy Features

From chapter 2, the wavelet decomposition of an image can be represented as

$$A_j = [H_x * [H_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2} \tag{7.5}$$

$$D_{j1} = [G_x * [H_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2} \tag{7.6}$$

$$D_{j2} = [H_x * [G_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2} \tag{7.7}$$

$$D_{j3} = [G_x * [G_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2} \tag{7.8}$$

where A_j and D_{jk} are the approximation and detail coefficients at each resolution level j and direction k. Many authors have used the energy of each detail band, calculated by

$$E_{jk} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} D_{jk}(m,n)}{MN} \tag{7.9}$$

as a set of texture descriptors, where M and N represent the size of each detail image.

These features can be directly extracted from a region of normalised text, giving a total of $3J$ features, where J is the number of decomposition levels used in the transform. In the experiments conducted below, a value of $J = 4$ is used, leading to a feature dimensionality of 12.

Wavelet Log Mean Deviation Features

The results presented in chapter 4 show that by applying a nonlinear function to the coefficients of the wavelet transform, a better representation of naturally textured images can be obtained. A number of such functions were examined, with the magnitude of the logarithm showing the best performance on a set of naturally textured images, described by

$$LMD_{jk} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\log\left(\frac{|D_{jk}(n,m)|}{S_j\delta}+1\right)}{MN} \tag{7.10}$$

where δ and S_j are constants specifying the degree of nonlinearity in the transform. A value of $\delta = 0.001$ was found to perform well in many experiments, while S_j is dependent upon the dynamic range of the input image. The number of features obtained in this manner is equal to the wavelet energy features, thus when using 4 levels of decomposition, a dimensionality of 12 is again obtained.

Wavelet Log Co-occurrence Signatures

A second set of texture features presented in chapter 4 are the wavelet log co-occurrence signatures, which capture second-order statistics of wavelet coefficients quantised on a logarithmic scale. Experimentally, these features were shown to significantly outperform similar statistics calculated using linear quantisation. When extracting these features, a undecimated form of the FWT is used in order to retain more spatial resolution, to provide a less sparse co-occurrence matrix, and to ensure translation invariance. Extracting these features over 4 resolution levels of such a transform gives a total of 96 features. To show the improvement obtained by using a logarithmic quantisation algorithm, the classification results with features extracted using linear quantisation (the wavelet cooccurrence signatures) are also presented.

Wavelet Scale Co-occurrence Signatures

The wavelet scale co-occurrence signatures developed in chapter 6 have been shown to perform well on a variety of naturally textured images, with lower overall classification errors than any of feature sets described above. Due to the binary nature of the input images, however, the extra information retained by these features in the form of intensity information may be of little use in distinguishing between various script types. In addition, because of the extensive pre-processing used to normalise the script images, the tolerance of these features to small changes in scale is not exploited to any advantage in this application. A detailed description of the extraction of these features can be found in section 6.5.

7.4.6 Classification Results

The proposed algorithm for automatic script identification from document images was tested on a database containing 8 different script types (Latin, Chinese, Japanese, Greek, Cyrillic, Hebrew, Devanagari, Arabic), with 200 samples from each script giving at total of 1600 individual images. Examples of these images are shown in figure 7.10. Each such image was binarised, deskewed and normalised using the algorithms described above. The resulting images were divided into two equal groups to create the training and testing sets.

The texture feature sets described above were then extracted separately from each image, and used to train a GMM classifier of the design described in section 5.4. The results of these experiments are shown in table 7.4. To illustrate the effectiveness of the use of LDA for feature sets of high dimensionality, classification results when this step was omitted are also shown.

From these results, it can be seen that the wavelet log co-occurrence significantly outperform any of the other features for script classification, with an overall error rate of only 1%. As was expected, the scale co-occurrence features did not perform

Texture Features	Classification Error	
	no discriminate analysis	discriminate analysis
GLCM Features	11.9%	9.1%
Gabor Energy	7.4%	4.9%
Wavelet Energy	7.9%	4.6%
Wavelet Log MD	8.3%	5.2%
Wavelet Cooc.	5.0%	2.0%
Wavelet Log Cooc.	4.9%	1.0%
Wavelet Scale Cooc.	9.3%	3.2%

Table 7.4: Script recognition results for each of the feature sets with and without feature reduction.

as well on the binary script images, with only a slightly reduced error rate when compared to the wavelet energy features. The GLCM features showed the worst overall performance, from which it can be concluded that pixel relationships at small distances are insufficient to characterise the script of a document image.

The distribution of the errors amongst the various script classes was approximately even, with the Chinese script having the lowest overall error rate for all of the features, and the largest errors arising from misclassifications between the Cyrillic and Greek scripts.

7.5 Adaptive GMM's for Improved Classifier Performance

Printed text, regardless of the script, has a distinct visual texture and is easily recognised as such by a casual observer. Because of this, it is possible to use this *a priori* knowledge to improve the modelling of the texture features when training the GMM for each class. By training a global model using all available testing model, then adapting this model for each individual class, a more robust representation can be obtained, somewhat limiting the blind nature of the learning algorithm. Using such a technique, it is also possible to train a class using

less training observations, since an initial starting point for the model is already available. This technique has been used with great success in applications where the general form of a model can be estimated using prior information, such as the modelling of speech and speakers [243].

7.5.1 MAP Adaptation

In section 5.3.3, the maximum likelihood estimate was defined as the parameter set $\hat{\boldsymbol{\lambda}}$ such that

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} l(\boldsymbol{\lambda}) \tag{7.11}$$

where $l(\boldsymbol{\lambda})$ is the likelihood of the training observations for that parametric form $\boldsymbol{\lambda}$ defined as

$$l(\boldsymbol{\lambda}) = p(\mathbf{o}|\boldsymbol{\lambda}) \tag{7.12}$$

Given these definitions, the ML framework can be thought of as finding a *fixed* but *unknown* set of parameters $\hat{\boldsymbol{\lambda}}$. In contrast to this, the MAP approach assumes $\boldsymbol{\lambda}$ to be a *random* vector with a known distribution, with an assumed correlation between the training observations and the parameters $\boldsymbol{\lambda}$ [244]. From this assumption, it becomes possible to make a statistical inference of $\boldsymbol{\lambda}$ using only a small set of adaption data $\mathbf{o}$, and prior knowledge of the parameter density $g(\boldsymbol{\lambda})$. The MAP estimate therefore maximises the posterior density such that

$$\begin{aligned} \boldsymbol{\lambda}_{MAP} &= \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}|\mathbf{o}) && (7.13) \\ &= \arg\max_{\boldsymbol{\lambda}} l(\mathbf{o}|\boldsymbol{\lambda}) g(\boldsymbol{\lambda}) && (7.14) \end{aligned}$$

Since the parameters of a prior density can also be estimated from an existing set of parameters $\boldsymbol{\lambda}_0$, the MAP framework also provides an optimal method of combining $\boldsymbol{\lambda}_0$ with a new set of observations $\mathbf{o}$.

In the case of a Gaussian distribution, the MAP estimations o fthe mean $\tilde{m}$ and variance $\tilde{\sigma}^2$ can be obtained using the framwork presented above, given prior distributions of $g(m)$ and $g(\sigma^2)$ respectively. If the mean alone is to be estimated,

this can be shown to be given by [245]

$$\tilde{m} = \frac{T\kappa^2}{\sigma^2 + T\kappa^2}\bar{x} + \frac{\sigma^2}{\sigma^2 + T\kappa^2}\mu \tag{7.15}$$

where T is the total number of training observations, $\bar{x}$ is the mean of those observations, and μ and κ^2 are the mean and variance respectively of the conjugate prior of m. From (7.15), it can be seen that the MAP estimate of the mean is a weighted average of the conjugate prior mean μ and the mean of the training observations. As $T \to 0$, this estimate will approach the prior μ, and as $T \to \infty$, it will approach $\bar{x}$, which is the ML estimate.

Using MAP to estimate the variance parameter, with a fixed mean, is accomplished in a somewhat simpler manner. Typically, a fixed prior density is used, such that

$$g(\sigma^2) = \begin{cases} \text{constant} & \sigma^2 \geq \sigma^2_{min} \\ 0 & \text{otherwise} \end{cases} \tag{7.16}$$

where σ^2_{min} is estimated from a large number of observations, in the case of our application the entire database of script images. Given this simplified density function, the MAP estimate of the variance is then given by

$$\tilde{\sigma}^2 = \begin{cases} S_x & S_x \geq \sigma^2_{min} \\ 0 & \text{otherwise} \end{cases} \tag{7.17}$$

where S_x is the variance of the training observations. This procedure is often known as *variance clipping*, and is effective in situations where limited training data does not allow for an adequate estimate of the variance parameter.

Using MAP to estimate both the mean and variance of a distribution is also possible, with the exact methodology given in [245].

7.5.2 Classification Results

Using the same training and testing data, the features extracted from section 7.4.6 were used to create a global script model. The optimal number of mixtures for

Texture Features	Classification Error
GLCM Features	8.9%
Gabor Energy	4.5%
Wavelet Energy	4.2%
Wavelet Log MD	4.1%
Wavelet Cooc.	1.4%
Wavelet Log Cooc.	0.8%
Wavelet Scale Cooc.	3.0%

Table 7.5: Script recognition results for various feature sets using MAP adaptation with large training sets.

this model was determined using the Bayes information criterion (BIC) described in section 5.4.1. The global model was then separately adapted for each class using the technique outlined above, and the test samples classified. The overall classification results from this experiment are shown in table 7.5. These results show a small improvement in overall classifier error when compared to those of table 7.4, due to the more accurate model obtained by utilising prior information.

The benefits of using MAP adaptation can be more clearly illustrated in situations where limited training data is available. To simulate such an environment, the amount of training observations for each class was limited to 25, compared to 200 in the previous experiments. The results of classifying the same 200 test observations using classifiers trained with and without MAP adaptation using the reduced training set are shown in table 7.6.

It is important to note that in these experiments a large amount of training data (100 samples per class) is used, resulting in well-defined models. In situations where less training data is available, it is expected that results will be somewhat poorer, and the benefit of using MAP adaptation more clearly illustrated. To test this hypothesis, the amount of training data was reduced to only 25 samples per class, and the experiment above repeated, with the classification errors shown in table 7.6. This test more clearly shows the benefits of the MAP adaptation

Texture Features	Classification Error	
	no MAP adaptation	MAP adaptation
GLCM Features	12.1%	9.9%
Gabor Energy	7.8%	5.3%
Wavelet Energy	7.2%	5.8%
Wavelet Log MD	8.2%	6.1%
Wavelet Cooc.	3.3%	2.7%
Wavelet Log Cooc.	2.4%	1.3%
Wavelet Scale Cooc.	5.6%	4.0%

Table 7.6: Script recognition results with and without MAP adaptation for various texture features for small training sets.

process, with error rates significantly reduced for each of the feature sets when compared to using models trained independently using the ML algorithm. The full set of results for both full and reduced training set with and without MAP adaptation is shown in figure 7.11.

7.6 Multi-Font Script Recognition

Within a given script there typically exists a large number of fonts, often of widely varying appearance. Because of such variations, it is unlikely that a model trained on one set of fonts will consistently correctly identify an image of a previously unseen font of the same script. To overcome this limitation, it is necessary to ensure that an adequate amount of training observations from each font to be recognised are provided in order that a sufficiently complex model is developed.

In addition to requiring large amounts of training data, creating a model for each font type necessitates a high degree of user interaction, with a correspondingly higher chance of human error. In order to reduce this level of supervision, an ideal system would automatically identify the presence of multiple fonts in the training data and process this information as required.

7.6.1 Clustered LDA

The discriminate function described in section 5.2.3 attempts to transform the feature space such that the inter-class separation is maximised, whilst minimising the intra-class separation, by finding the maximum of the cost function $tr(\mathbf{C}\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{C}')$. Whilst this function is optimal in this sense, it does make a number of strong assumptions regarding the nature of the distributions of each class in feature space. All classes are assumed to have equal covariance matrices, meaning that the resulting transform will be optimal only in the sense of separation of the class means. Additionally, since the function is linear, multi-modal distributions cannot be adequately partitioned in some circumstances. Figure 7.12 shows a synthetic example of this situation, where the two classes are clearly well separated in feature space, however have the same mean and therefore an effective linear discriminate function cannot be created. When analysing scripts containing multiple fonts, it is common to encounter such multi-modal distributions in feature space within a particular script, as the texture features extracted from different fonts can vary considerably.

To overcome this limitation of LDA, it is possible to perform automatic clustering on the data prior to determining the discriminate function, and assign a separate class label to each individual cluster. Training and classification is then performed on this extended set of classes, and the final decision mapped back to the original smaller set of classes. Although this leads to less training data for each individual subclass, using the adaptation technique presented in the previous section can somewhat overcome this problem. Taking the example shown of figure 7.12, each of the clusters in both classes would be represented separately, creating a decision rule which is easily determined using the LDA approach previously proposed.

Determining the optimal number of clusters is a problem which has been previously addressed in the literature [246, 247, 248]. However, for the purposes of multi-font script recognition, using a fixed number of clusters has shown to

provide adequate results at significantly reduced computational cost. In the experiments in the following section, 10 clusters are used in all cases, as this number was found to be generally sufficient to describe the font variations present within all of the tested scripts. Although the majority of classes can in fact be represented adequately using fewer than this number of clusters, using more clusters does not significantly degrade performance.

7.6.2 Classification Results

To illustrate the limitations of using a single model in a multi-font environment, experiments using a number of fonts from each script class were conducted. A total of 30 fonts were present in the database, with 10 from Latin script, 4 each from Chinese, Japanese and Arabic, and 3 each from Devanagari, Hebrew, Greek, and Cyrillic. 100 training and testing samples were extracted from each font type.

To illustrate the limitations of using a single model for multiple fonts, each of the scripts was trained as a single class using the MAP classification system proposed above. From the results shown in table 7.7, it can be seen that large errors are introduced, with the most common misclassification occurring between fonts of the Latin and Greek. Interestingly, these results show that the simpler texture features do not suffer the same performance degradation as the more complicated features, with the wavelet energy signatures showing the lowest overall classification error of 12.3%.

The proposed clustering algorithm is implemented by using k-means clustering to partition each class into 10 regions. Each subclass is then assigned an individual label, and LDA and classification performed as normal. The results of this experiment are shown in table 7.8, with the wavelet log co-occurrence features again providing the lowest overall error rate of 2.1%. Although the error rates for each of the feature sets is slightly higher than the single font results of table 7.6, a vast improvement is achieved when compared to the results obtained using a

Texture Features	Classification Error
GLCM Features	15.9%
Gabor Energy	13.1%
Wavelet Energy	12.3%
Wavelet Log MD	12.6%
Wavelet Cooc.	14.9%
Wavelet Log Cooc.	13.2%
Wavelet Scale Cooc.	15.0%

Table 7.7: Script recognition error rates for scripts containing multiple fonts when trained with a single model.

Texture Features	Classification Error
GLCM Features	12.5%
Gabor Energy	7.7%
Wavelet Energy	6.9%
Wavelet Log MD	7.0%
Wavelet Cooc.	3.2%
Wavelet Log Cooc.	2.1%
Wavelet Scale Cooc.	5.5%

Table 7.8: Script recognition error rates for scripts containing multiple fonts when clustering is used to create multiple models.

single model only.

7.7 Chapter Summary

This chapter has shown the effectiveness of texture analysis techniques in the field of document processing, and more specifically to the problem of automatic script identification. A review of the field of document analysis is presented, covering topics such as document segmentation, binarisation, removal of skew, form processing and document understanding. Many of these applications require that the script of the document is known, making automatic script identification

of document images a required pre-processing task. A review of current methods of determining the script and/or language were presented

A number of texture features were evaluated for the purpose of script recognition, including those developed in previous chapters of this thesis. In terms of the individual performance of these features, results showed that several approaches give excellent results, with the GLCM features, wavelet log co-occurrence and scale co-occurrence features giving the lowest classification error rates. Combining these sets using a variety of classifier combination techniques was also evaluated, showing a small increase in overall performance.

In order to provide more stable models of each script class, as well as reducing the need for excessive training data, using MAP adaptation when training the classifier was evaluated. Because of the strong inter-class correlations which exists between the extracted features of script textures, this approach was found to be well suited to the application of automatic script identification. Experimental results showed a small increase in overall classification performance when using large training sets, and significant improvement when limited training data is available.

Using a single model to characterise multiple fonts within a script class has been shown to be inadequate, as the fonts within a script class can vary considerably in appearance, often resulting in a multi-modal distribution in feature space. To overcome this problem, a technique whereby each class is automatically segmented using the kmeans clustering algorithm before performing LDA is presented. By doing this, a number of subclasses are automatically generated and trained, without the need for any user intervention. Experiments performed on a multi-font script database have shown that this technique can successfully identify many different fonts of the same language.

Figure 7.10: Examples of document images used for training and testing. (a) Latin, (b) Chinese, (c) Greek, (d) Cyrillic, (e) Hebrew, (f) Devanagari, (g) Japanese and (h) Arabic.

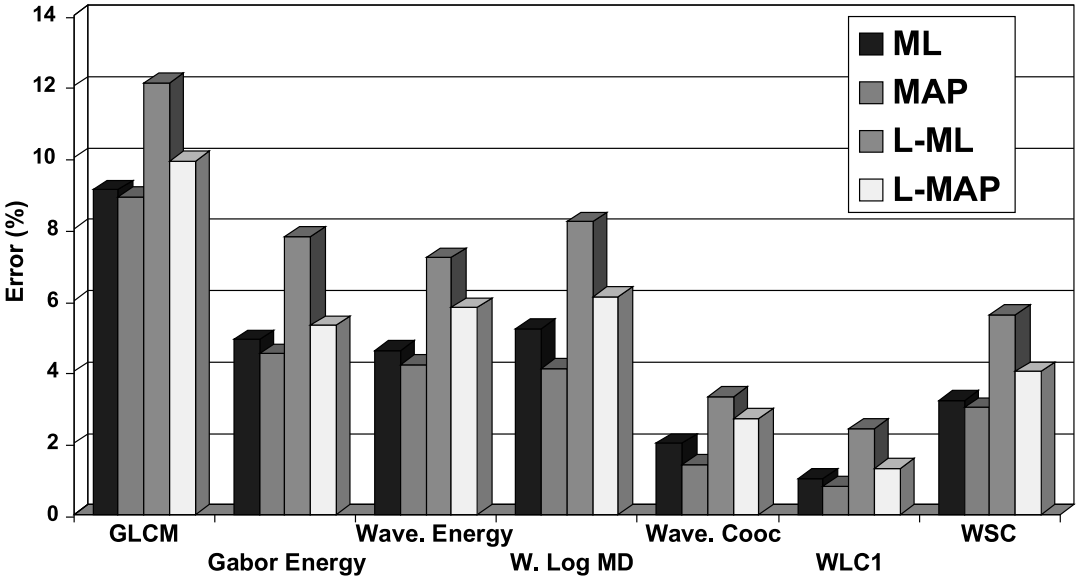


Figure 7.11: Classification errors for each of the tested texture feature sets using both ML and MAP training methods. Results for both full training sets and reduced training sets (L-ML and L-MAP) are shown.

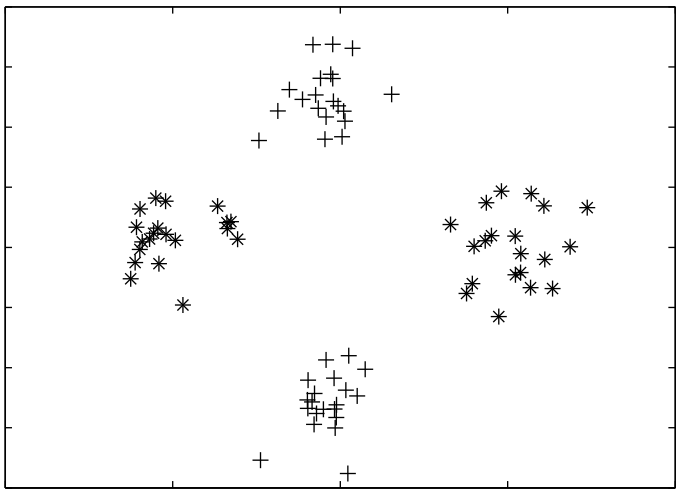


Figure 7.12: Synthetic example of the limitations of LDA. The two multi-modal distributions, although well separated in feature space, have identical means and hence an effective linear discriminate function cannot be determined.

Chapter 8

Conclusions and Future Work

This chapter concludes the thesis with a summary of the work presented. The major conclusions and original contributions made in each chapter are summarised, and a number of possible avenues for future research are identified.

8.1 Conclusions

This thesis has presented a thorough review of the topic of texture analysis, in particular those techniques which make use of the wavelet transform. While a large amount of work has been conducted in this field, it is still an extremely active topic of research, with many problems still unsolved. Texture analysis has also shown to be of great use in many practical applications, ensuring that it will remain of significant importance in the future.

The original work in this thesis is primarily targeted at three particular areas of texture analysis:

1. Improving existing features, and developing new features for the characterisation of texture, in order to increase the overall classification accuracy. To-

wards this goal, investigations into the non-linear transformation of wavelet coefficients and the relationships between bands of the wavelet transform were undertaken, with a number of new texture features resulting from these.

2. An investigation into the effects of classifier choice on overall performance, and the design of a new classification system for texture.

3. The applications of the texture features developed to the problem of automated script identification from document images, an important task in the field of document analysis.

A summary of the conclusions of each chapter are as follows:

Chapter 2 provided a thorough theoretical description of the wavelet transform, from its roots in Fourier theory and the short-time Fourier transform to the efficient algorithms used to calculate the wavelet transforms of discrete signals. Topics covered in this chapter include the continuous wavelet transform, wavelet frames, the dyadic wavelet transform, the discrete-time wavelet transform and its implementation, the fast wavelet transform, wavelet packets, and multi-dimensional wavelet transforms. A number of current applications of wavelets were discussed in the fields of signal and image processing, compression, and numerical and statistical analysis problems.

Chapter 3 presented an overview of the field of texture analysis. The difficulty of forming a concise definition of what constitutes 'texture' was investigated, with a number of different viewpoints from the literature examined. The various sub-fields of texture analysis were then explained, covering texture classification, segmentation, compression and synthesis. In all of these tasks, a variety of methodologies have been proposed, ranging from statistical measures of the grey-level values of pixels, to the properties of

individual texture elements or 'textons', and the more recent multiresolution approaches which are common today. A brief description of many of these methods has been presented, with the advantages and disadvantages of each explained. To conclude the chapter, a number of applications of texture analysis were identified, showing the diversity of fields which have been influenced by this area of research.

Chapter 4 presented the results of an investigation into the effect of quantisation strategies on the overall classification performance of wavelet texture features. By applying a non-linearity to the wavelet coefficients, it was shown that a better characterisation of many natural textures is possible. Using this information, two new texture feature sets extracted from first and second order statistics of these transformed coefficients are proposed, with experimental results performed on a number of sets of text images showing considerably improvements compared to existing features.

Chapter 5 presented a thorough review of classifier theory, giving an overview of a number of commonly used designs, with particular attention given to their use in the field of texture classification. An extensive investigation into the use of such classifiers in the field of texture analysis found that non-parametric classifiers are the most common choice for this task due to the widely varying nature of textures, often poorly modelled by many parametric models. While such classifiers successfully overcome this difficulty, their overall performance is not optimal, with error rates approaching twice the theoretical minimum. By using an optimal linear discriminate function coupled with a Gaussian mixture model, it was experimentally shown that it is possible to improve on the performance of these systems, with the proposed design providing reduced overall classification error rates. By dynamically adapting the topology of the classifier to suit the training data of each individual texture, such a parametric approach can provide a near-optimal representation of the feature distribution without overtraining the model. The computational complexity and storage requirements of the

proposed system are also considerably lower than the more commonly used nonparametric techniques.

Chapter 6 has shown that the relationships between neighbouring bands of the wavelet transform contain information which is crucial to the adequate characterisation of texture. By modelling such relationships using *scale co-occurrence matrices*, a novel and powerful texture descriptor is developed. Using the earth mover's distance, it was shown that such a descriptor can be used to search and retrieve images from a database in an efficient manner. A new set of texture features extracted from these matrices was also proposed, and experimentally shown to outperform features which are extracted independently from each band. Furthermore, by combining these features in an optimal manner with those developed in chapter 4, further improvements in overall accuracy are obtained.

Chapter 7 investigated the application of the texture classification methods previously developed in this thesis to the problem of automated script recognition from document images, an important application in the field of document processing. In order to facilitate the extraction of such features, a number of pre-processing stages were also developed, including a fast and accurate method of skew detection and line distortion removal, and an algorithm for normalising text block to give a uniform appearance. Script classifications experiments using a variety of feature sets have shown that second order statistics of wavelet coefficients provide the best results, significantly outperforming the Gabor energies which have been previously proposed in the literature. By utilising prior information regarding the appearance of printed text and a MAP framwork, classification results are further improved and the training requirements for each class reduced. The problem of detecting multiple fonts within each script was also investigated, with a clustered LDA technique proposed for this purpose.

8.2 Future Work

Continuing on from the research presented in this thesis, a number of possible avenues for future research have been identified, including:

(i) Chapter 4 presented a logarithmic quantisation technique which, when applied to the coefficients of the wavelet transform, was experimentally shown to improve the characterisation of many natural textures. By using other nonlinear transforms, it may be possible to further improve this representation, leading to lower overall classification error rates. Another possible avenue for further research is the use of multiple histogram models for each coefficient distribution, using the degree of matching for each as a feature vector.

(ii) Chapter 6 has proposed a novel texture representation which models the relationships between bands of the wavelet decomposition, and uses this representation for both image retrieval and texture classification tasks. Further investigation into the extraction of features from these matrices may lead to improved performance, and is another avenue for possible future research. More research is also required in order to link such features with observable properties of textured images, in order that a more complete understanding of the model is obtained.

(iii) The experimental results of chapter 6 have shown that these features provide a better characterisation of certain textures when compared to second-order statistics of the wavelet coefficients, while performing more poorly on other images. Further study must be undertaken in this area to identify properties of textures which make them more suited to each technique. This work could be further expanded to identify texture types which are more accurately modelled by each of the numerous competing texture analysis methodologies. Such an analysis would be greatly beneficial when determining the optimal analysis method for a given application.

(iv) The field of document processing is an active and challenging research field, and texture analysis methodologies have recently been used to good effect in many such applications of the field. Chapter 7 showed that the automatic determination of the script of a document image is one such application, with results showing a high degree of accuracy regardless of the form of the text or the font used. In providing these results a number of texture analysis algorithms were investigated, however a great number of other techniques remain untested. There exists much potential to improve on the results presented in this thesis by testing other texture-based approaches, and by adaptation of the features shown in this work. The combination of these features in an intelligent manner is also a topic worth of future consideration.

(v) The automatic detection of text from both document images and other media, such as video footage, is another application for which texture analysis has been shown to be effective. Based on the work presented in chapter 7, a number of promising avenues for further research in this area have been identified, whereby regions of an image likely to contain text or text-like features are quickly identified using textural properties.

Bibliography

[1] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, pp. 961–1005, 1990.

[2] S. G. Mallat, "A theory for multiresolution signal decomposition : the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[3] S. G. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $l^2(r)$," *Transactions of the American Mathematical Society*, vol. 315, pp. 69–87, 1989.

[4] D. Gabor, "Theory of communication," *Journal of the Institute of Electrical Engineers*, vol. 93, no. 3, pp. 429–457, 1946.

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

[6] O. Rioul and M. Vetterli, "Wavelet and signal processing," *IEEE Signal Processing Magazine*, pp. 14–38, 1992.

[7] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: Society for Industrial and Applied Mathematics, 1992.

[8] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[9] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme," *Math. Annal.*, vol. 69, pp. 331–371, 1910.

[10] S. G. Mallat and S. Zhong, "Wavelet transform maxima and multiscale edges," *Wavelets and Their Applications*, pp. 67–104, 1992.

[11] S. G. Mallat, "Zero-crossings of a wavelet transform," *IEEE Transactions on Information Theory*, vol. 37, pp. 1019–1033, 1991.

[12] M. J. Shensa, "The discrete wavelet transform: wedding the a trous and Mallat algorithms," *IEEE Transactions on Signal Processing*, vol. 40, pp. 2464–2482, 1992.

[13] S. Jaggi, A. S. Willsky, W. C. Karl, and S. Mallat, "Multiscale geometrical feature extraction and object recognition with wavelets and morphology," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 372–375, 1995.

[14] M. I. Khalil and M. M. Bayoumi, "Affine invariant object recognition using dyadic wavelet transform," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 421–425, 2000.

[15] Q. M. Tieng and W. W. Boles, "Object recognition using an affine invariant wavelet representation," in *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, pp. 307–311, 1994.

[16] X. Wu and B. Bhanu, "Gabor wavelet representation for 3-D object recognition," *IEEE Transactions on Image Processing*, vol. 6, no. 1, pp. 47–64, 1997.

[17] M. Lang, H. Guo, J. E. Odegard, C. S. Burrrus, and R. O. Wells, "Noise reduction using an undecimated discrete wavelet transform," *IEEE Signal Processing Letters*, vol. 3, no. 1, pp. 10–12, 1996.

[18] D. I. Donoho, "De-noising by soft thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[19] P. L. Ainsleigh and C. K. Chui, "A B-wavelet-based noise-reduction algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1279–1284, 1996.

[20] N. A. Whitmal, J. C. Rutledge, and J. Cohen, "Reducing correlated noise in digital hearing aids," *IEEE Engineering in Medicine and Biology Magazine*, vol. 15, no. 5, pp. 88–96, 1996.

[21] C. L. Martinez, X. F. Canovas, and M. Chandra, "SAR interferometric phase noise reduction using wavelet transform," *Electronics Letters*, vol. 37, no. 10, pp. 649–651, 2001.

[22] V. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 9, pp. 9–21, 2001.

[23] B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 22–35, 2001.

[24] J. Shapiro, "Embedded image coding using zerotress of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, 1993.

[25] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, pp. 1158–1170, 2000.

[26] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning," *IEEE Transactions and Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, 1996.

[27] Z. Xiong, K. Ramchandran, and M. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Transactions on Image Processing*, vol. 6, pp. 677–693, 1997.

[28] D. Marpe, G. Blattermann, J. Ricke, and Maass, "A two-layered wavelet-based algorithm for efficient lossless and lossy image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 7, pp. 1094–1102, 2000.

[29] S. Grgic, M. Grgic, and B. Zovko-Cihlar, "Performance analysis of image compression using wavelets," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 3, pp. 682–695, 2001.

[30] E. Y. Hamid and Z.-I. Kawasaki, "Wavelet-based data compression of power system disturbances using the minimum description length criterion," *IEEE Transactions on Power Delivery*, vol. 17, no. 2, pp. 460–466, 2002.

[31] Z. Yang, M. Kallergi, R. A. DeVore, B. J. Lucier, W. Qian, R. A. Clark, and L. P. Clark, "Effect of wavelet bases on compressing digital mammograms," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 5, pp. 570–577, 1995.

[32] L. Zeng, C. P. Jansen, S. Marsch, M. Unser, and P. R. Hunziker, "Four-dimensional wavelet compression of arbitrarily sized echocardiogrphic data," *IEEE Transactions on Medical Imaging*, vol. 21, no. 9, pp. 1179–1187, 2002.

[33] D. Zhou and W. Cai, "A fast wavelet collocation method for high-speed circuit simulation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 8, pp. 920–930, 1999.

[34] S. Barmada and M. Raugi, "Transient numerical solutions of nonuniform MTL equations with nonlinear loads by wavelet expansion in time or space domain," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 8, pp. 1178–1190, 2000.

[35] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London*, vol. B 207, pp. 187–217, 1980.

[36] J. M. Coggins, *A framework for texture analysis based on spatial filtering.* Phd, Michigan State University, 1982.

[37] H. Tamura, S. Mori, and Y. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pp. 460–473, 1978.

[38] J. Sklansky, "Image segmentation and feature extraction," *IEEE Transactions on Systems Man and Cybernetics*, vol. 8, pp. 237–247, 1978.

[39] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, pp. 786–804, 1979.

[40] W. Richards and A. Polit, "Texture matching," *Kybernetic*, vol. 16, pp. 155–162, 1974.

[41] S. Zucker and K. Kant, "Multiple-level representations for texture discrimination," in *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, (Dallas, TX), pp. 609–614, 1981.

[42] J. Hawkins, "Textural properties for pattern recognition," in *Picture Processing and Psychopictorics* (B. Lipkin and A. Rosenfeld, eds.), Academic Press, 1969.

[43] J. Bergen and B. Julesz, "Rapid discrimination of visual patterns," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 5, pp. 857–863, 1983.

[44] B. Julesz, "Visual pattern discrimination," *IRE Transactions on Information Theory*, vol. 8, pp. 84–92, 1962.

[45] B. Julesz, E. N. Gilbert, L. A. Shepp, and H. L. Frisch, "Inability of humans to discriminate between visual texture that agree in second-order statistics - revisited," *Perception*, vol. 2, pp. 391–405, 1973.

[46] B. Julesz, "Experiments in the perception of visual texture," *Scientific American*, vol. 232, no. 4, pp. 34–43, 1975.

[47] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, pp. 91–97, 1981.

[48] R. L. DeValois, "Spatial-frequency selectivity of cells in macaque visual cortex," *Vision Research*, vol. 22, pp. 545–559, 1982.

[49] R. von der Heydt, E. Peterhans, and M. Dürsteler, "Periodic-pattern-selective cells in monkey visual cortex," *Journal of Neuroscience*, vol. 12, pp. 1416–1434, 1992.

[50] A. W. Busch and W. W. Boles, "Multi-resolution pre-processing technique for rotation invariant texture classification," in *Proceedings of WOSPA*, 2000.

[51] C. Becchetti and P. Campisi, "Parsimonious texture synthesis using binomial linear prediction," in *Proceedings of Sixth International Conference on Image Processing and Its Applications*, vol. 2, pp. 600–603, 1997.

[52] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–71, 2000.

[53] R. Bajesy and L. Lieberman, "Texture gradient as a depth cue," *Computer Graphics and Image Processing*, vol. 5, pp. 52–67, 1976.

[54] H. Vorhees, "Finding texture boundries in images," Master's thesis, MIT, 1987.

[55] F. Tomita and S. Tsuji, *Computer analysis of visual textures.* Boston: Kluwer Academic Publishers, 1990.

[56] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, pp. 610–621, 1973.

[57] M. Tuceryan and A. K. Jain, *The Handbook of Pattern Recognition and Computer Vision*, ch. 2.1, pp. 207–248. World Scientific Publishing Co., 2 ed., 1998.

[58] J. E. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society*, vol. 36, pp. 192–326, 1974.

[59] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[60] R. Paget and I. D. Longstaff, "Texture synthesis via a noncausal nonparametric multiscale Markov random field," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 925–931, 1998.

[61] J. Davidson, A. Talukder, and N. Cressie, "Texture analysis using partially ordered Markov models," in *Proceesings of IEEE International Conference on Image Processing*, vol. 3, pp. 402–406, 1994.

[62] X. Gong and N.-K. Huang, "Textured image recognition using hidden Markov model," in *Proceedings of 1988 International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1128–1131, 1988.

[63] B. R. Povlow and S. M. Dunn, "Texture classification using noncausal hidden Markov models," in *Proceedings of 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 642–643, 1993.

[64] A. Speis and G. Healey, "An analytical and experimental study of the performance of markov random fields applied to textured images using small samples," *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 447–458, 1996.

[65] G. C. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 25–39, 1983.

[66] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 39–55, 1987.

[67] K. I. Laws, *Textured image segmentation.* PhD thesis, University of Southern California, 1980.

[68] T. N. Tan, "Texture edge detection by modeling visual cortical channels," *Pattern Recognition*, vol. 28, no. 9, pp. 1283–1298, 1995.

[69] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[70] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," in *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*, pp. 14–19, 1990.

[71] R. Panda and B. N. Chatterji, "Unsupervised texture segmentation using tuned filters in Gaborian space," *Pattern Recognition Letters*, vol. 18, no. 5, pp. 445–53, 1997.

[72] A. C. Bovik, "Multichannel texture analysis using localised spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 55–73, 1990.

[73] P. Kruizinga and N. Petkov, "Grating cell operator features for oriented texture segmenation," in *Proceedings of Fourtheenth International Conference on Pattern Recognition*, vol. 2, pp. 1010–1014, 1998.

[74] P. Kruizinga, N. Petkov, and S. E. Grigorescu, "Comparison of texture features based on Gabor filters," in *Proceedings of the International Conference on Image Analysis and Processing, 1999*, pp. 142–147, 1999.

[75] A. Teuner, O. Pichler, and B. J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters," *IEEE Transactions on Image Processing*, vol. 4, no. 6, pp. 863–870, 1995.

[76] D. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 947–964, 1995.

[77] T. Tan, "Rotation invariant texture features and their use in automatic script identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 751–756, 1998.

[78] A. Laine and J. Fan, "Frame representations for texture segmentation," *IEEE Transactions on Image Processing*, vol. 5, pp. 771–780, 1996.

[79] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1186–1191, 1993.

[80] J. R. Smith and S. F. Chang, "Transform features for texture classification and discrimination in large image databases," in *IEEE International Conference on Image Processing*, vol. 3, pp. 407–411, 1994.

[81] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on Image Processing*, vol. 2, pp. 429–441, 1993.

[82] A. Busch and W. W. Boles, "Texture classification using multiple wavelet analysis," in *Proceedings of DICTA*, pp. 341–345, 2002.

[83] G. Van de Wouwer, P. Scheunders, and D. Van Dyck, "Statistical texture characterization from discrete wavelet representations," *IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 592–598, 1999.

[84] M.-C. Lee and C.-M. Pun, "Texture classification using dominant wavelet packet energy features," in *4th IEEE Southwest Symposium on Image Analysis and Interpretation*, vol. 1, pp. 301–304, IEEE Comput. Soc Los Alamitos CA USA, 2000. English.

[85] J.-W. Wang, "Multiwavelet packet transforms with application to texture segmentation," *Electronics Letters*, vol. 38, no. 18, pp. 1021–1023, 2002.

[86] P. Dewaele, P. V. Gool, and A. Oosterlinck, "Texture inspection with self-adaptive convolution filters," in *Proceedings of the 9th International Conference on Pattern Recognition*, (Rome, Italy), pp. 56–60, Nov. 1988.

[87] D. Chetverikov, "Detecting defects in texture," in *Proceedings of the 9th International Conference on Pattern Recognition*, (Rome, Italy), pp. 61–63, 1988.

[88] J. Chen and A. K. Jain, "A structural approach to indentify defects in textured images," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, (Beijing, China), pp. 29–32, 1988.

[89] A. Bodnarova, M. Bennamoun, and K. K. Kubik, "Defect detection in textile materials based on aspects of the HVS," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4423–4428, 1998.

[90] R. W. Conners, C. W. McMillin, K. Lin, and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: part of an automated lumber processing system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 573–583, 1983.

[91] L. H. Siew, R. M. Hodgson, and E. J. Wood, "Texture measures for carpet wear assessment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 92–105, 1988.

[92] A. K. Jain, F. Farrokhnia, and D. H. Alman, "Texture analysis of automotive finishes," in *Proceedings of SME Machine Vision Applications Conference*, (Detroit, MI), pp. 1–16, November 1990.

[93] P. A. Freeborough and N. C. Fox, "Mr image texture analysis applied to the diagnosis and tracking of Alzheimer's disease," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 475–478, Jun 1998.

[94] J. K. Kim and H. W. Park, "Statistical textural features for detection of microcalcifications in digitized mammograms," *IEEE Transactions on Medical Imaging*, vol. 18, pp. 231–238, Mar 1999.

[95] N. R. Mudigonda, R. Rangayyan, and J. E. L. Desautels, "Gradient and texture analysis for the classification of mammographic masses," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 1032–1043, Oct 2000.

[96] J. S. Bleck, U. Ranft, M. Gebel, H. Hecker, M. Westhoff-Bleck, and C. Thiesemann, "Random field models in the textural analysis of ultrasonic images of the liver," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 796–801, Dec 1996.

[97] Q. Ji, J. Engel, and E. Craine, "Texture analysis for classification of cervix lesions," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 144–1149, Nov 2000.

[98] P. Wang, S. M. Krishnan, C. Kugean, and M. P. Tjoa, "Classification of endoscopic images based on texture and neural network," in *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4, pp. 3691–3695, 2001.

[99] R. Sutton and E. L. Hall, "Texture measures for automatic classification of pulmonary disease," *IEEE Transactions on Computers*, vol. 21, pp. 667–676, 1972.

[100] U. G. H. Harms and H. M. Aus, "Combined local color and texture anlysis of stained cells," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 364–376, 1986.

[101] G. H. Landeweerd and E. S. Gelsema, "The use of nuclear texture parameters in the automatic analysis of leukocytes," *Pattern Recognition*, vol. 10, pp. 57–61, 1978.

[102] M. F. Insana, R. F. Wagner, B. S. Garra, D. G. Brown, and T. H. Shawker, "Analysis of ultrasound image texture via generalised rician stratistics," *Optical Engineering*, vol. 25, pp. 743–748, 1978.

[103] C. C. Chen, J. S. Daponte, and M. D. Fox, "Fractal features analysis and classification in medical imaging," *IEEE Transactions on Medical Imaging*, vol. 8, pp. 133–142, 1989.

[104] A. Lundervold, "Ultrasonic tissue characterization - a pattern recognition approach," tech. rep., Norwegian Computing Center, Oslo, Norway, 1992.

[105] F. Ulaby, F. Kouyate, B. Brisco, and T. L. Williams, "Textural information in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 24, pp. 235–245, 1986.

[106] M. Simard, G. DeGrandi, K. P. B. Thomson, and G. B. Benie, "Analysis of speckle noise contribution on wavelet decomposition of SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 6, pp. 1953–1962, 1998.

[107] E. Rignot and R. Kwok, "Extraction of textural features in SAR images: statistical model and sensitivity," in *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 1979–1982, 1990.

[108] L. Kurvonen and M. T. Hallikainen, "Textural information of multitemporal ERS-1 and JERS-1 SAR images with applications to land and forest type classification in boreal zone," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 680–689, 1999.

[109] 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 Transactions on Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2310–2321, 2000.

[110] S. Dellepiane, D. Giusto, S. Serpico, and G. Vernazza, "Sar image recognition by integration of intensity and textural information," *International Journal on Remote Sensing*, vol. 12, no. 9, pp. 1915–1932, 1991.

[111] L. Pierce, F. Ulaby, K. Sarabandi, and M. Dobson, "Knowledge-based classification of polarimetric SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 1081-1086, 1994.

[112] M. Dobson, F. Ulaby, and L. Pierce, "Land-cover classification and estimation of terrain attributes using synthetic aperture radar," *Remote Sensing*, vol. 51, pp. 199–214, 1995.

[113] A. H. Schistad and A. K. Jain, "Texture analysis in the presence of speckle noise," in *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, pp. 884–886, 1992.

[114] J. H. Lee and W. D. Philpot, "A spectral-textural classifier for digital imagery," in *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 2005–2008, 1990.

[115] L. J. Du, "Texture segmentation of SAR images using localized spatial filtering," in *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 1983–1986, 1990.

[116] C.-M. Pun and M.-C. Lee, "Rotation-invariant texture classification using a two-stage wavelet packet feature approach," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 148, no. 6, pp. 422–428, 2001.

[117] D. Charalampidis and T. Kasparis, "Wavelet-based rotational invariant roughness features for texture classification and segmentation," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 825–837, 2002.

[118] N. Sebe and M. S. Lew, "Wavelet based texture classification," in *Proceedings of 15th International Conference on Pattern Recognition*, vol. 3, pp. 947–950, 2000.

[119] W. Wenjian and W. G. Wee, "Texture classification using wavelet maxima representation," *Proceedings of the SPIE The International Society for Optical Engineering*, vol. 3391, pp. 378–85, 1998.

[120] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[121] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America*, vol. 4, no. 12, pp. 2379–2394, 1987.

[122] B. A. Olshausen and D. J. Field, "Natural image statistics and efficient coding," *Network: Computation in Neural Systems*, vol. 7, pp. 333–339, 1996.

[123] E. P. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," in *Proceedings of International Conference on Image Processing*, vol. 1, pp. 62–66, 1998.

[124] M. Unser and M. Eden, "Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 4, pp. 804–815, 1990.

[125] P. Brodatz, *Textures: A Photographic Album for Artists and Designers.* New York: Dover Publications Inc., 1966.

[126] "Vistex texture database." http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

[127] G. V. Trunk, "A problem of dimensionality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 3, pp. 306–307, 1979.

[128] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[129] P. M. Narandra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. 26, no. 9, pp. 917–922, 1977.

[130] H. Liu and R. Setiono, "A probabilistic approach to feature selection - a filter solution," in *Proceedings of International Conference on Machine Learning*, (Bari, Italy), pp. 319–327, 1996.

[131] K. Chen and H. Liu, "Towards an evolutionary algorithm: a comparison of two feature selection algorithms," in *Proceedings of the 1999 Congress on Evolutionary Computing*, vol. 2, pp. 1309–1313, 1999.

[132] P. L. Lanzi, "Fast feature selection with genetic algorithms: a filter approach," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 537–540, 1997.

[133] J. E. Smith, T. C. Fogarty, and I. R. Johnson, "Genetic selection of features for clustering and classification," in *Proceedings of IEE Colloquium on Genetic Algorithms in Image Processing*, pp. 4/1–4/5, 1994.

[134] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164–171, 2000.

[135] J. Kittler, "Feature set search algorithms," in *Pattern Recognition and Signal Processing* (C. H. Chen, ed.), pp. 41–60, Alphen aan den Rijn, The Netherlands: Sijthoff and Noordhoff, 1978.

[136] P. Pudil, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119–1125, 1994.

[137] H. Liu, H. Motoda, and M. Dash, "A monotonic measure for optimal feature selection," in *Proceedings of European Conference on Machine Learning*, pp. 101–106, 1998.

[138] H. Yuan, S.-S. Tseng, W. Gangshan, and Z. Fuyan, "A two-phase feature selection method using both filter and wrapper," in *1999 International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 132–136, 1999.

[139] S. Roweis, "EM algorithms for PCA and SPCA," *Neural Information Processing Systems*, vol. 10, pp. 626–632, 1997.

[140] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* San Diego: Academic Press, 2 ed., 1990.

[141] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* New York: John Wiley & Sons, Inc., 2001.

[142] H. Abdi, "A neural network primer," *Journal of Biological Systems*, vol. 2, no. 3, pp. 247–283, 1994.

[143] Y. Q. Chen, M. S. Nixon, and D. W. Thomas, "Neural networks and texture classification," in *Proceedings of IEE Colloquium on Appliations of Neural Networks to Signal Processing*, pp. 6/1–6/4, 1994.

[144] A. Branca, W. Delaney, F. P. Lovegine, and A. Distante, "Surface defect detection by a texture analysis with a neural network," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1497–1502, 1995.

[145] A. Visa, K. Valkealahti, and O. Simula, "Cloud detection based on texture segmentation by neural network methods," in *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 1001–1006, 1991.

[146] P. Schumacher and J. Zhang, "Texture classification using neural networks and discrete wavelet transform," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 903–907, 1994.

[147] P. P. Raghu, R. Poongodi, and B. Yegnanarayana, "Unsupervised texture classification using vector quantization and deterministic relaxation neural

network," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1376–1387, 1997.

[148] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[149] L. Meng and Q. H. Wu, "Error-centre-based algorithm for support vector machine training," *Electronics Letters*, vol. 38, no. 7, pp. 349–350, 2002.

[150] D. Mattera, F. Palmieri, and S. Haykin, "An explicit algorithm for training support vector machines," *Signal Processing Letters*, vol. 6, no. 9, pp. 243–245, 1999.

[151] O. L. Mangasarian and D. R. Musicant, "Successive overrelaxation for support vector machines," *IEEE Transacations on Neural Networks*, vol. 10, no. 5, pp. 1032–1037, 1999.

[152] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, "Supervised texture segmentation using support vector machines," *Electronics Letters*, vol. 35, no. 22, pp. 1935–1937, 1999.

[153] A. Kumar and H. C. Shen, "Texture inspection for defects using neural networks and support vector machines," in *Proceedings of the 2002 International Conference on Image Processing*, vol. 3, pp. 3/353–3/356, 2002.

[154] Y. Ma, T. Fang, K. Fang, D. Wang, and W. Chen, "Texture image classification based on support vector machine and distance classification," in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, vol. 1, pp. 551–554, 2002.

[155] S. Fukada and H. Hirosawa, "Support vector machine classification of land cover: application to polarmetric SAR data," in *Proceedings of 2001 Geoscience and Remote Sensing Symposium*, vol. 1, pp. 187–189, 2001.

[156] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from imcomplete data via the EM algorithm," *J. Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[157] D. A. Reynolds, "Experimental evaluation of features for robust speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 539–643, 1994.

[158] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. 3300 AH Dordrecht, The Netherlands: Kluwer Academic Publishers, 1992.

[159] J. W. Pelecanos, *Robust Automatic Speaker Recognition*. PhD thesis, Queensland University of Technology, Jan 2003.

[160] M. Yang, N. Abuja, and D. Kriegman, "Mixtures of linear subspaces for face detection," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 70–76, 2000.

[161] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[162] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773–795, 1994.

[163] J. Olivier and R. Baxter, "MML and Bayesianism: similarities and differences," Tech. Rep. 206, Monash University, Australia, 1994.

[164] G. Schwarz, "Estimating the dimensionality of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[165] T. Chang and C. C. Kuo, "Texture segmentation with tree-structured wavelet transform," in *Proceedings of IEEE International Symposium on Time-Frequency and Time-Scale Analysis*, vol. 2, p. 577, 1992.

[166] S. Fukuda and H. Hirosawa, "A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2282–2286, 1999.

[167] H. Greenspan, S. Belongie, R. Goodman, and P. Perona, "Rotation invariant texture recognition using a steerable pyramid," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 2, (Jerusalem, Israel), pp. 162–167, 1994.

[168] S.-D. Kim and S. Udpa, "Texture classification using rotated wavelet filters," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 30, no. 6, pp. 847–852, 2000.

[169] A. Tabesh, "Zero-crossing rates of wavelet frame representations for texture classification," *Electronics Letters*, vol. 38, no. 22, pp. 1340–1341, 2002.

[170] C. W. Shaffrey, N. G. Kingsbury, and I. H. Jermyn, "Unsupervised image segmentation via Markov trees and complex wavelets," in *Proceedings of 2002 International Conference on Image Processing*, vol. 3, pp. 801–804, 2002.

[171] A. Busch, W. W. Boles, and S. Sridharan, "Calculating the similarity of textures using wavelet scale relationships," in *Proceedings of Australian and New Zealand Intelligent Information Systems Conference*, pp. 507–512, 2003.

[172] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of IEEE International Conference on Computer Vision*, (Bombay, India), pp. 59–66, 1998.

[173] F. S. Hillier and G. J. Liberman, *Introduction to mathematical programming.* McGraw-Hill, 1990.

[174] A. Busch and W. W. Boles, "Texture classification using wavelet scale relationships," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 3484–3487, 2002.

[175] J. Kittler, M. Hatef, and R. P. W. Duin, "Combining classifiers," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 2, pp. 897–901, 1996.

[176] J. Cao, M. Ahmadi, and M. Sridhar, "Recognition of handwritten numerals with multiple feature and multistage classifier," *Pattern Recognition*, vol. 28, no. 2, pp. 153–160, 1995.

[177] S. B. Cho and J. H. Kim, "Multiple network fusion using fuzzy logic," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 497–501, 1995.

[178] J. Franke and E. Mandler, "A comparison of two approaches for combining the votes of cooperating classifiers," in *Proceedings of 11th IAPR International Conference on Pattern Recognition*, vol. 2, pp. 611–614, 1992.

[179] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[180] L. Xu, A. Krzyzak, and C. Y. Suen, "Mehtods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[181] M. Hatef, J. Kittler, and R. P. Dunn, "Combining multiple classifiers," tech. rep., University of Surrey, 1996.

[182] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary OCR system for English and Arabic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 495–504, 1999.

[183] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 38–62, 2000.

[184] Y. Y. Tang, C. D. Yan, and C. Y. Suen, "Document processing for automatic knowledge acquisition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 1, pp. 3–21, 1994.

[185] B. Kruatrachue and P. Suthaphan, "A fast and efficient method for document segmentation," in *IEEE Region 10 International Conference on Electrical and Electronic Technology*, vol. 1, (Singapore), pp. 381–383, 2001.

[186] 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, pp. 161–190, 1993.

[187] 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.

[188] F. Shih, S.-S. Chen, D. Hung, and P. Ng, "A document image segmentation, classification and recognition system," in *Proceedings of the International Conference on Systems Integration*, pp. 258–267, 1992.

[189] A. Busch, W. W. Boles, and S. Sridharan, "A multiresolution approach to document segmentation," in *Proceedings of WOSPA*, pp. 43–46, 2002.

[190] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1162–1173, 1993.

[191] S. Tsujimoto and H. Asada, "Major components of a complete text reading system," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1133–1149, 1992.

[192] A. Simon, J.-C. Pret, and A. P. Johnson, "A fast algorithm for bottom-up document layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 273–277, 1997.

[193] K. Etemad, D. Doermann, and R. Chellappa, "Multiscale segmentation of unstructured document pages using soft decision integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 92–96, 1997.

[194] C. Strouthopoulos and N. Papamarkos, "Text identification for document image analysis using a neural network," *Image and Vision Computing*, vol. 16, pp. 879–896, 1998.

[195] V. Wu, R. Manmatha, and E. M. Riseman, "Finding text in images," in *Second ACM International Conference on Digital Libraries*, (Philadelphia, PA), 1997.

[196] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 147–156, 2000.

[197] N. Jin and Y. Y. Tang, "Text area localization under complex-background using wavelet decomposition," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pp. 1126–1130, 2001.

[198] P. Clark and M. Mirmedhi, "Combining statistical measures to find image text regions," in *Proceedings of the 15th International Conference on Pattern Recognition*, pp. 450–453, 2000.

[199] L. Xingyuan, D. Doermann, W.-G. Oh, and W. Gao, "A robust method for unknown forms analysis," in *Fifth International Conference on Document Analysis and Recognition*, vol. 1, pp. 531–534, 1999.

[200] O. Hori and D. Doermann, "Robust table-form structure analysis based on box-driven reasoning," in *Third International Conference on Document Analysis and Recognition*, vol. 1, pp. 218–221, 1995.

[201] J. Yuan, Y. Tang, and C. Suen, "Four directional adjacency graphs (FDAG) and their application in locating fields in forms," in *Third International*

Conference on Document Analysis and Recognition, vol. 2, pp. 752–755, 1995.

[202] H. Shinjo, K. Nakashima, M. Koga, K. Marukawa, Y. Shima, and E. Hadano, "A method for connecting desappeared junction patterns on frame lines in form documents," in *Fourth International Conference on Document Analysis and Recognition*, vol. 1, pp. 667–670, 1997.

[203] J.-L. Chen and H.-J. Lee, "An efficeitn algorithm for form structure extraction using strip projection," *Pattern Recognition*, vol. 31, no. 9, pp. 1353–1368, 1998.

[204] Y. Belaid, A. Belaid, and E. Turolla, "Item searching in forms: application to French tax form," in *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 744–747, 1995.

[205] Y. Zheng, C. Liu, X. Ding, and S. Pan, "Form frame line detection with directional single-connected chain," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pp. 699–703, 2001.

[206] A. Busch, W. W. Boles, S. Sridharan, and V. Chandran, "Detection of unknown forms from document images," in *Proceedings of Workshop on Digital Image Computing*, pp. 141–144, 2003.

[207] G. Tauschek, "Reading machine." 2026329, Dec. 1935.

[208] M. H. Glauberman, "Character recognition for business machines," *Electronics*, pp. 132–136, 1956.

[209] ERA, "An electronic reading automaton," *Electronic Engineering*, pp. 189–190, 1957.

[210] L. P. Horwitz and G. L. Shelton, "Pattern recognition using autocorrelation," *Proceedings of the IRE*, vol. 49, no. 1, pp. 30–35, 1961.

[211] Y. Noguchi and T. Iijima, "Pattern classification system using equivariance characteristic parameters," *Transactions of the Information Processing of Japan*, vol. 11, pp. 107–116, 1971.

[212] M. T. Y. Lai and C. Y. Suen, "Automatic recognition of character by Fourier descriptors," *Pattern Recognition*, vol. 14, no. 1, pp. 383–393, 1981.

[213] H. Freeman, "Boundary encoding and processing," in *Picture Processing andPsycholopictorics*, pp. 241–266, New York: Academic Press, 1970.

[214] S. Kahan, T.Pavlidis, and H. S. Baird, "On the recognition of printed characters of any font and size," *IEEE Transcations on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 274–288, 1987.

[215] T. Pavlidis and F. Ali, "Computer recognition of hand written numerals by polygonal approximations," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 5, pp. 610–614, 1975.

[216] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029–1058, 1992.

[217] G. Nagy and S. Seth, "Hierarchical representation of optically scanned documents," in *Proceedings of the Seventh International Conference on Pattern Recognition*, pp. 347–349, 1984.

[218] A. L. Spitz, "Determination of the script and language content of document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 235–245, 1997.

[219] C. Ronse and P. Devijver, *Connected Components in Binary Images: The Detection Problem.* Research Studies Press, 1984.

[220] D. S. Lee, C. R. Nohl, and H. S. Baird, "Language identification in complex, unoriented, and degraded document images," in *IAPR Workshop on Document Analysis and Systems*, pp. 76–98, 1996.

[221] C. Suen, N. N. Bergler, B. Waked, C. Nadal, and A. Bloch, "Categorizing document images into script and language classes," in *International Conference on Advances in Pattern Recognition*, (Plymouth, UK), pp. 297–306, 1998.

[222] A. L. Spitz and M. Ozaki, "Palace: A multilingual document recognition system," in *International Association for Pattern Recognition Workshop on Document Analysis Systems*, (Singapore), pp. 16–37, World Scientific, 1995.

[223] A. Busch, W. W. Boles, S. Sridharan, and V. Chandran, "Texture analysis for script recognition," in *Proceedings of IVCNZ*, pp. 289–293, 2001.

[224] G. Peake and T. Tan, "Script and language identification from document images," in *Proceedings of Workshop on Document Image Analysis*, vol. 1, pp. 10–17, 1997.

[225] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[226] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 273–285, 1985.

[227] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, pp. 41–47, 1986.

[228] Y. Liu, R. Fenich, and S. N. Srihari, "An object attribute thresholding algorithm for document image binarization," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 278–281, 1993.

[229] J. Yang, Y. Chen, and W. Hsu, "Adaptive thresholding algorithm and its hardware implementation," *Pattern Recognition Letters*, vol. 15, pp. 141–150, 1994.

[230] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen, "Adaptive document binarization," in *Proceedings of ICDAR'97*, pp. 147–152, 1997.

[231] Y. Liu and S. N. Srihari, "Document image binarization based on texture features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 540–544, 1997.

[232] W. Postl, "Detection of linear oblique structures and skew scan in digitized documents," in *Proceedings of International Conference on Pattern Recognition*, pp. 687–689, 1986.

[233] G. Peake and T. Tan, "A general algorithm for document skew angle estimation," in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 230–233, 1997.

[234] B. B. Chaudhuri and U. Pal, "Skew angle detection of digitized Indian script documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 703–712, 1997.

[235] H. S. Baird, "The skew angle of printed documents," in *Document Image Analysis* (L. O'Gorman and R. Kasturi, eds.), pp. 204–208, IEEE Computer Society Press, 1995.

[236] A. Vailaya, H. J. Zhang, and A. K. Jain, "Automatic image orientation detection," in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 600–604, 1999.

[237] S. Lowther, V. Chandran, and S. Sridharan, "An accurate method for skew determination in document images," in *Digital Image Computing Techniques and Applications, 2002*, vol. 1, (Melbourne, Australia), pp. 25–29, 2002.

[238] M. Acharyya and M. K. Kundu, "Document image segmentation using wavelet scale-space features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1117–1127, 2002.

[239] J. L. Fisher, S. C. Hinds, and D. P. D'Amato, "A rule-based system for document image segmentation," in *Proceedings of the Tenth International Conference on Pattern Recognition*, pp. 567–572, 1990.

[240] T. Taxt, P. J. Flynn, and A. K. Jain, "Segmentation of document images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1322–1329, 1989.

[241] M. I. C. Murguia, "Document segmentation using texture variance and low resolution images," in *Proceedings of 1998 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 164–167, 1998.

[242] S. Tsuruoka, N. Watanaba, N. Minamide, F. Kimura, Y. Miyake, and M. Shridhar, "Base line correction for handwritten slant correction," in *Proceedings of the Third International Conference on Document Analysis and Recognition*, vol. 2, pp. 902–905, 1995.

[243] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proceedings of EUROSPEECH*, vol. 2, pp. 963–970, 1997.

[244] C. Lee and J. Gauvain, "Bayesian adaptive learning and MAP estimation of HMM," in *Automatic Speech and Speaker Recognition: Advanced Topics*, pp. 83–107, Boston, MA: Kluwer Academic Publishers, 1996.

[245] C.-H. Lee, C.-H.Lin, and B.-H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 39, no. 4, pp. 806–814, 1991.

[246] H.-S. Rhee and K.-W. Oh, "A validity measure for fuzzy clustering and its use in selecting optimal number of clusters," in *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1020–1025, 1996.

[247] K. S. Younis, M. P. DeSimio, and S. K. Rogers, "A new algorithm for detecting the optimal number of substructures in the data," in *Proceedings of the IEEE Aerospace and Electronis Conference*, vol. 1, pp. 503–507, 1997.

[248] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–780, 1989.