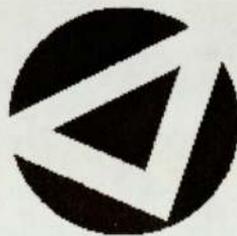


Textural Segmentation of Natural Water Scenes for Surface Pollution Detection

RAJPAL K RAI

MSc by Research in Pattern Analysis and Neural Networks

Supervisor: Professor David Lowe



ASTON UNIVERSITY

September 2000

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

ASTON UNIVERSITY

Textural Segmentation of Natural Water Scenes for Surface Pollution Detection

RAJPAL K RAI

MSc by Research in Pattern Analysis and Neural Networks, 2000

Thesis Summary

Water-borne pollutants are currently monitored through spot sampling. This gives an occasional, localised and therefore unreliable picture of the level of contamination. This research is part of a larger EU funded project, 'Blue Water' the aim of which is to develop a system capable of continuous and automatic monitoring of water-borne pollution, through the use of remotely sensed visible band camera images.

Water-borne pollution generates surface slicks which have a different texture to normal turbulent waves. In this thesis we develop a pixel by pixel segmentation algorithm which classifies the image into slick and non-slick textured regions. We test the algorithm on a set of grey scale lake images. The main stages of the algorithm are preprocessing of the images, feature extraction, classification, and finally postprocessing of the segmentation results. The segmentation process is based on a novelty detection approach. We build histogram and multivariate Gaussian density models of slick feature vectors which then represent 'normality'. Receiver operating characteristic curves are used to set the decision boundaries of normality for these models, using expertly labelled slick and non-slick data. Each unseen pixel is then classified according to this model as either normal or novel, i.e having slick or non-slick like texture. A range of feature extraction techniques have been investigated namely, statistical moments, principal components analysis and finally one and two dimensional fast Fourier transforms.

Keywords: Image Segmentation, Texture Classification, Feature Extraction, Density Modelling, Novelty Detection.

Acknowledgements

Firstly, I would like to thank Professor David Lowe for his guidance and for always finding time to discuss the work. The image set used in the project was also provided by Professor David Lowe.

I would also like to thank Dr David Barber for help with mathematical questions and for his useful suggestions.

I am grateful to the Engineering and Physical Sciences Research Council and the EU project 'Blue Water' for providing financial support.

Finally, I would like to thank my family for their constant support and my fellow MSc students for their good humour throughout the year.

Contents

1	Introduction	9
1.1	Motivation for Research	9
1.2	The Scope and Aims of this Research	11
1.3	Thesis Outline	12
2	Image Segmentation	14
2.1	Introduction	14
2.1.1	Supervised Versus Unsupervised Approach	15
2.2	Textural Segmentation of Natural Images	16
2.2.1	Textural Information	16
2.2.2	Review of Previous Work on Texture Segmentation	17
2.2.3	A General Texture Segmentation System	21
2.3	Textural Segmentation of Lake Images	23
2.3.1	Description of Image Set	23
2.3.2	Discussion of Adopted Approach	25
3	Methods, Segmentation Results and Analysis	29
3.1	Expert Labelling of Training Image	29
3.2	Segmentation by Histogram Thresholding	31
3.3	Feature Based Segmentation	34
3.3.1	Introduction	34
3.3.2	Preprocessing of Images	34
3.3.3	Statistical Moments	36

CONTENTS

3.3.3.1	Receiver Operating Characteristic (ROC) Curves . . .	41
3.3.3.2	Results	42
3.3.3.3	Analysis	44
3.3.4	Principal Component Analysis	47
3.3.4.1	Density Modelling	51
3.3.4.2	Results	51
3.3.4.3	Analysis	52
3.3.5	One Dimensional Discrete Fourier Transform	53
3.3.5.1	Results	59
3.3.5.2	Analysis	59
3.3.6	Two Dimensional Discrete Fourier Transform	62
3.3.6.1	Results	64
3.3.6.2	Analysis	64
4	Comparison of Segmentation Results	67
5	Conclusions and Future Work	72
5.1	Conclusions	72
5.2	Future Work	73
	Bibliography	77

List of Figures

- 1.1 Windwave channel experiment 11
- 2.1 (a) Texture block, (b) corresponding co-occurrence matrix 17
- 2.2 A general segmentation system 21
- 2.3 Test Images 1-3 26
- 2.4 Test Images 4 and 5 27
- 2.5 Important features of image 2 28
- 3.1 Expert labelling of image regions 30
- 3.2 (a) Histogram of training image (b) histogram of slick region only . . . 32
- 3.3 Segmentation of images 3 and 5 using histogram thresholding 33
- 3.4 Contrast enhancement 35
- 3.5 (a) Histogram of slick and (b) wave data from training image 37
- 3.6 Variance, skewness and kurtosis plotted for slick and wave data 39
- 3.7 Histograms of variance for slick and wave data 40
- 3.8 Classification matrix 40
- 3.9 Toy data example, (a) histogram of likelihood values for labelled slick
and non-slick data, (b) resulting ROC curve 41
- 3.10 ROC curves for variance values of training data 43
- 3.11 Moments: Segmentation of image 2 using different sized subimages . . . 45
- 3.12 Moments: Segmentation of image 4 using different sized subimages . . . 46
- 3.13 Eigenvalue spectra of slick data using (a) 4×4 (b) 8×8 and (c) 16×16
subimages 49
- 3.14 PCA: Slick and wave data projected onto first three principal components 50

LIST OF FIGURES

3.15	PCA: ROC curves of likelihood values of training data	52
3.16	PCA: Segmentation of image 2 using different threshold values	54
3.17	PCA: Segmentation of image 2 using different sized subimages	55
3.18	PCA: Segmentation of image 3 using different sized subimages	56
3.19	1D FFT: Slick and wave data projected onto first three principal components	58
3.20	1D FFT: Segmentation of image 2 using different sized subimages	60
3.21	1D FFT: Segmentation of image 4 using different sized subimages	61
3.22	2D FFT: Slick and wave data projected onto first three principal components	63
3.23	2D FFT: Segmentation of image 2 using different sized subimages	65
3.24	2D FFT: Segmentation of image 3 using different sized subimages	66
4.1	Segmentation results for image 2	68
4.2	Segmentation results for image 3	69
4.3	Segmentation results for image 4	70
4.4	Segmentation results for image 5	71

List of Tables

3.1	Number of subimages obtained from training image	30
3.2	Classification rates using statistical moments	42
3.3	PCA: Number of principal components retained	48
3.4	PCA: Segmentation results for different threshold values	53
3.5	Classification rates using PCA	53
3.6	1D FFT: Number of principal components retained	57
3.7	Classification rates using 1D FFT	59
3.8	2D FFT: Number of principal components retained	62
3.9	Classification rates using 2D FFT	64

Chapter 1

Introduction

1.1 Motivation for Research

Currently, sea pollution is monitored through a programme of spot sampling. The water samples taken are then sent to a laboratory for chemical analysis. It may take a few days before results are available in which time the pollution may no longer be present, making it extremely difficult to identify the source. If the spot sampling does not coincide with the discharge of pollution it is impossible to detect such occurrences, leaving the system open to abuse. This occasional, localised sampling gives an unreliable picture of the level of contamination, especially for large stretches of water. A system capable of continuous, automatic monitoring is highly desirable. The requirement for such a system provides the motivation for this research. The ideal would be to develop a system which can locate, track and possibly identify sources of surface pollutants from digital images of near-shore sea water.

This research is a small part of a much wider EU funded project, 'Blue Water', proposal number: IST-1999-10388. The project involves the following partners, Aston University (UK), British Maritime Technology (UK), ENEL (Italy), TXT Ingegneria Informatica (Italy), Yermasoyia Municipal Authority (Cyprus), University of Hamburg (Germany), University Joseph Fourier, Grenoble (France) and Flow Research Diagnostics (UK). The pattern processing algorithms will be developed exclusively within the Neural Computing Research Group at Aston University.

Matter found in sea water such as hydrocarbons, oil, algal bloom, bacteria and fish oil generate slicks which leave surface signatures by damping wave motions. These differences in the surface morphology are revealed by reflected light. The areas of damped waves have a smooth homogeneous appearance as the light has been reflected with little scatter. Normal turbulent waves on the other hand scatter the incident light giving rise to their rougher more uneven appearance. Therefore, to locate slicks within an image, techniques are required which will allow these textural differences to be identified and modelled in some appropriate feature space.

Experiments carried out under controlled conditions by Dr Philipp Lange, a member of the Blue Water consortium at the University of Hamburg, clearly show evidence of this damping effect ¹. Normal turbulent waves are generated in a windwave channel by exposing the water to a constant 5m/s wind, see Figure 1.1(a). Oleyl alcohol is then added to the channel to simulate the effect of a surface pollutant. This has an immediate damping effect on the waves as can be seen from Figure 1.1(b), eventually all waves in the channel are damped, Figure 1.1(c), as the alcohol film moves along the channel assisted by the wind. After approximately one minute has elapsed normal waves are returning once again as most of the alcohol has passed through the channel, Figure 1.1(d).

A large volume of research has been carried out in the field of image segmentation. Examples include the segmentation of satellite synthetic aperture radar images for identifying land use, segmentation for automatic detection of objects such as vehicles or people, segmentation of documents into areas of text and pictures for automatic processing [9] and the segmentation of medical images for diagnostic purposes [3]. This wide range of applications has led to the development of many different techniques, each suited to a particular type of problem. An overview of image segmentation approaches is presented in Chapter 2 with emphasis being given to discussing techniques that are relevant to the segmentation of natural images.

¹with thanks and acknowledgement to University of Hamburg for providing this data



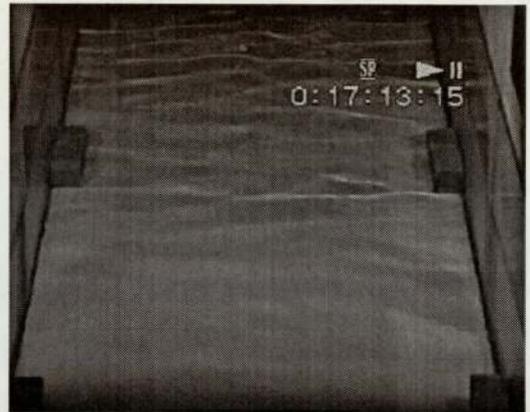
(a)



(b)



(c)



(d)

Figure 1.1: Windwave channel experiment (courtesy of University of Hamburg)

1.2 The Scope and Aims of this Research

This research aims to investigate several key areas, which will work towards finding a solution to the problem of image segmentation for water-borne pollution detection. However, before outlining the main focus of this work we must formally state the nature and scope of this project, and the assumptions which have been made.

It has not been possible to obtain near-shore sea images at this stage of the project. A set of lake images has been used instead. These images present several challenging problems representative of some of those which would be encountered in sea images. A detailed description of the test image set is provided in Section 2.3.1. Although

the images are in colour, only grey scale information has been used in this work. The images have been taken from a digital video sequence. However, no time information has been used in the segmentation work. All the test images contain some background such as rocks, trees and grass. In the final system these areas could be pre-labelled as non-slick by using geographical knowledge of the scene. In this work the segmentation algorithms have been applied to the whole image with no pre-labelling. However, the performance of the work is judged solely on the segmentation results achieved for the water regions of the image.

The primary aim of this research is to develop a segmentation process within a statistical framework, that will allow pixel by pixel segmentation of an image into slick and non-slick textured regions. A general segmentation system involves several stages including, preprocessing, feature extraction, classification and finally postprocessing. A key factor in designing a successful segmentation process is to find a feature space in which the feature vectors from different texture classes form distinct clusters. The classifier is then used to identify the decision boundaries in this feature space. The main focus of the work is to investigate and compare the performance of different feature extraction techniques. This study will also draw conclusions regarding the suitability of using visible band images for water-borne pollution detection, and whether it would be useful to include dynamic information in the algorithm.

1.3 Thesis Outline

In this chapter we have presented the background and motivation for this research. We have also stated the scope and specific aims of this MSc research project.

Chapter 2 gives an overview of image segmentation. We discuss the different nature of the image segmentation problem when applied to artificial versus natural images. This is followed by a discussion of the supervised versus unsupervised approach to image segmentation. We then concentrate on techniques relevant to the segmentation of natural images by texture. We present a review of previous work on texture feature extraction and classification. This is followed by a description of a general texture

segmentation system, based on supervised learning. Finally, we describe the image set to be used in our research. We discuss the main features of the images and how they will impact the design of the segmentation algorithm. We conclude by outlining the general approach we have adopted for implementing a successful segmentation algorithm.

A detailed description of the techniques we investigated for our segmentation algorithm can be found in Chapter 3. We also present the segmentation results achieved and provide an analysis of each method. The segmentation pipeline includes preprocessing of the images, feature extraction, statistical modelling of the features, followed by the decision stage and finally the postprocessing of segmentation results. Four feature extraction methods are evaluated namely, statistical moments, linear principal component analysis and one and two dimensional discrete Fourier transforms. Histogram thresholding is a standard non-statistical image segmentation method often used on images of man-made environments. We include a description of this method and present the segmentation results achieved for our image set. These results are later compared to those achieved using the statistical techniques detailed above.

In Chapter 4 we compare the segmentation results achieved with the range of methods we have employed.

Chapter 5 summarises the conclusions of our investigation and presents a direction for future work.

Chapter 2

Image Segmentation

2.1 Introduction

There exists a wide spectrum of methods for image analysis and segmentation. The techniques employed vary depending on the nature of the problem in hand. There is an obvious split in the methods used to segment natural images and those used for artificial or man-made environments. Natural scenes have a much higher level of structural complexity and variability than man-made environments. A typical natural scene will contain several different textural regions all of which will be subject to variation in time. The effect of environmental conditions is one major source of variability in natural scenes. For example, fog, mist, the level of daylight and the prevailing wind conditions will alter the contrast, brightness and the textural information in the image. Standard image processing techniques such as template matching, thresholding and edge detection are often used to segment images of artificial objects or man-made environments. However, this approach would fail if applied to the problem of segmenting an image containing textural regions [15]. The alternative to working in the pixel domain is to work in feature space. This usually involves representing our original data with a smaller number of variables or features which are chosen for their ability to discriminate between different textures. These feature vectors can then be used as inputs to an appropriate classifier.

One pixel domain segmentation method is to apply thresholding to the image to

select pixels with certain grey values, these will correspond to the grey values of the object to be identified. We demonstrate the serious shortcomings of this method when applied to natural scenes in Section 3.2. In the rest of this thesis we concentrate on statistical, feature based segmentation methods. A comparison of segmentation results achieved using feature vectors and image thresholding is made in Chapter 4, demonstrating the superiority of the former approach.

2.1.1 Supervised Versus Unsupervised Approach

Classification techniques fall into two main groups, supervised and unsupervised. In supervised classification the class labels are known for the training data, whereas there is no labelled data for unsupervised classification.

For supervised image segmentation an expert user must first label the data by selecting regions of the image that belong to recognised classes. The labelled data can then be used in a range of classification methods such as k-nearest neighbour, Fisher discriminant analysis, conditional density models or neural networks.

In unsupervised classification we aim to establish the number of classes present and the features which distinguish them by seeking clusters within similar data. This is done using iterative techniques such as k-means (also known as c-means), see Webb [23] or Duda and Hart [6] for a detailed description of clustering techniques. There are several difficulties with unsupervised methods such as knowing the number of centres to choose, initialisation of the cluster means and the iterative nature of the process may make it time consuming. Once clusters have been identified the user must then assign them to the classes of interest. As one might expect unsupervised classification is more difficult. However, it may be chosen in preference to supervised classification in certain cases when there is a lack of sufficient labelled training data.

For certain problems a semi-supervised approach using partly labelled data could be employed. In many practical situations we wish to classify data as either conforming to some predefined normal or abnormal state, for example condition monitoring of aircraft engines. A conditional density model of 'normality' could be constructed using

feature vectors extracted from the labelled data corresponding to normal operation. Boundaries of normality would also be set for the model. Any prior knowledge and data available for the abnormal case could also be used in setting this boundary. For a new feature vector, the probability of belonging to the normal class would be calculated, if this is within the boundaries defined it is classed as normal, else it is classed as abnormal. This approach can be extended to multiclass problems in which one class or a group of classes are defined as normal, any data which deviates from this ‘normal’ model by a predefined margin is then classed as abnormal or novel. This technique is commonly known as novelty detection and is used in a wide range of applications including fraud detection, condition monitoring and alarming, and in medical diagnostic systems. Nairac *et al* [12] used this approach to detect unusual signatures in jet engine vibration data.

In this research we work with a set of images which can be labelled into slick and non-slick regions by an expert. For this reason only supervised and semi-supervised classification techniques will be considered here on in. This approach presents an easier problem than using an unsupervised method for the reasons outlined above.

2.2 Textural Segmentation of Natural Images

2.2.1 Textural Information

Texture is a local property that depends on the tonal (grey level) variations of pixels in a given spatial region. There are no precise mathematical models for describing homogeneous image textures. However, as humans are capable of discriminating between different textures fairly easily, texture features used for classification are designed to correspond well to our human visual textural perception [20]. These features include coarseness, contrast, directionality, line-likeness, regularity and roughness [21].

2.2.2 Review of Previous Work on Texture Segmentation

A large volume of literature exists on image segmentation by texture. Many researchers have asked the question, what makes a good set of features for texture classification. There is no overall best method which has emerged, instead a range of methods exists some of which are better suited to certain problems than others. In recent years there has been significant interest in the use of multiresolution decomposition methods such as wavelets [3, 5, 11, 13, 15, 19] and the competing method of multichannel filtering such as with Gabor filters [9, 11, 15, 19, 24]. The range of texture feature extraction techniques encountered in this literature survey fall into four main groups, statistical, geometric, model based and signal processing methods. In the remainder of this section we discuss the advantages and disadvantages of these different techniques. All these approaches extract features from a windowed section of the image with the pixel of interest in the centre. Wavelet analysis and Gabor filtering can also be performed on the image as a whole.

In the statistical approach the stochastic properties of the tonal variations are characterised. One statistical method is to use co-occurrence matrices. In this method the joint or co-occurrence of grey level values between pixel pairs, along a predefined displacement vector are recorded in a $G \times G$ matrix, where G is the number of grey levels. For example, the co-occurrence matrix for the texture block shown in Figure 2.1(a) for the displacement vector $(1, 0)$, is given in Figure 2.1(b).

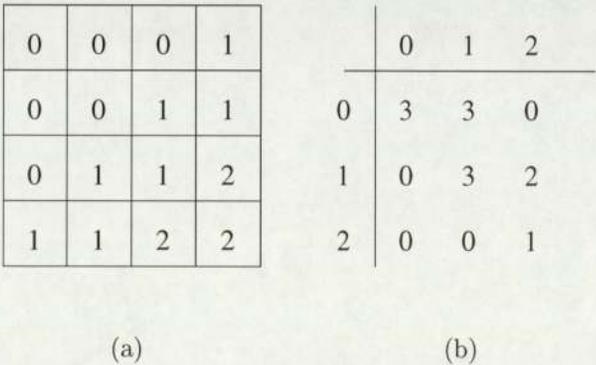


Figure 2.1: (a) Texture block, (b) corresponding co-occurrence matrix

This method aims to capture information regarding the spatial dependency of grey

level values. Smooth textures will tend to have larger values clustered around the diagonal. Second order statistical texture features are then extracted from this co-occurrence matrix [20]. There are a range of possible features which could be used, Strand and Taxt [20] selected angular second moment, contrast, correlation and entropy. Using a nearest neighbour classifier good segmentation results were achieved for two test images which contained four different natural textures each, taken from the Brodatz photographic album [2]. However, the boundaries were not clearly identified. Better segmentation results were achieved if the number of features used was increased from 4 to 8. However, this method was not able to segment an image containing textures with a stochastic structure. This highlights an important aspect of methods which extract features from a window of predefined size. Satisfactory segmentation results can only be achieved if the textural structure is small in relation to the window size. For stochastic textures, features are required which measure the average spatial texture properties, therefore larger windowed areas are needed [20]. This method is also inefficient in terms of memory requirements and computational time, this problem becomes more serious for images composed of large numbers of grey values.

Faugeras and Pratt [7] presented the view that texture regions can be thought of as two-dimensional samples from a stochastic process, described by its statistical parameters. Furthermore, this is suggested as a good model for natural textures such as sand, grass and water. A whitening operator was applied to the two-dimensional autocorrelation function of a windowed texture region. The resulting two-dimensional uncorrelated texture pattern was assumed to be a sample from the underlying stochastic generating process. A histogram of the grey level values of this uncorrelated texture pattern was formed from which features were extracted such as mean, standard deviation, skewness and kurtosis. Several natural texture patterns from the Brodatz album [2], were compared by evaluating their Bhattacharyya distance (B-distance) [8]. This is a scalar function of the conditional probability densities of feature vectors of two classes. Gaussian densities were assumed in this case. This is a computationally intensive method and it was suggested the whitening operator be replaced with a Sobel

gradient operator to speed the process up [18]. The results of this study did not provide convincing evidence that a stochastic model was suitable for natural textures. Several texture pairs such as grass and sand or wool and sand did produce large B-distances, however, other texture pairs such as grass and raffia did not.

Geometric methods are based on the view that textures can be characterised with set geometric rules. A subclass is structural methods in which textures are described as a set of basic repeating patterns with some displacement rules. In practice, it is difficult to extract such rules from real textures in which the underlying structure is too complicated to be described by a set of rigid geometric rules. Strand and Taxt [20] used a geometric method based on detecting local frequencies along a set of direction vectors to segment images containing different textural regions. The segmentation results were compared to those achieved using co-occurrence matrices, which were discussed earlier in this section. The local frequency method performed about equally as well as the co-occurrence matrix method in discriminating between the textures, with the local frequency method being slightly better at detecting the boundaries. As with the co-occurrence matrix method the local frequency method was not able to discriminate between textures with a stochastic structure for the reasons discussed above.

One popular model based approach for texture analysis is to employ Markov random fields (MRFs). A random field is a spatial function that assigns a random variable at each spatial location [14]. A MRF is a random field probability density function. Fitting a MRF to a texture block allows a method of encoding spatial information which characterises mutual influences of neighbouring pixels. For a first order MRF the probability of a given pixel is defined such that it depends on the state of its nearest neighbours. A second order MRF would take into account the first and second nearest neighbours. The texture feature vector is given by the model parameters which maximise the probability of observing a given texture block, having assumed a certain parameterised probability density function. Chen and Huang [4] fitted 2nd order MRF models to a set of four images of different natural textures, these included grass, tree bark, calf leather and woodgrain from the Brodatz album [2]. A four di-

dimensional feature vector was generated for each $N \times N$ training texture block from the test images. Principal component analysis was used to project the four dimensional feature vectors on to two dimensions. This was repeated using three different MRF models, the generalised ising model, auto-binomial model and gaussian markov random field model. Visual examination of the two dimensional projections showed none of these models were able to completely separate any of the four textures, in fact there was considerable overlap between the distributions of the classes. Classification error rates were given for nearest neighbour, quadratic and Fisher's linear classifier. These confirmed the poor performance of the MRF models, the best correct classification rate achieved was 30%. This experiment was repeated using a set of four sandpaper textures. This produced similarly poor correct classification rates. This method is also computationally intensive. Chen and Huang [4] state the computational time required to generate a single feature vector from a 64×64 texture block as ranging from 0.42 seconds for the gaussian markov random field model to 4.8 seconds for the generalised ising model, on a Sun 4/490 Sparc station.

Signal processing methods perform frequency analysis of textures. This can be done through filtering in the frequency domain. Fourier transforms, wavelet analysis and Gabor filtering are some of the most common signal processing methods. Wavelets allow multiresolution decomposition of signals representing them as coarse versions then going to finer and finer detail. Wavelets allow a joint time frequency representation of signals. However there is a trade-off between resolution in the time domain and frequency domain. As the resolution increases in the time domain it decreases in the frequency domain. Wavelets have been used by many researchers for carrying out texture segmentation [3, 15]. Physiological experiments carried out on animals suggest simple cells in the visual cortex have a receptive field whose response is dependent on the frequency and orientation of the visual stimulus. Two-dimensional Gabor filters have been suggested as a good approximation for this receptive field and have been successfully applied to texture classification problems [5, 9, 11, 15, 24].

2.2.3 A General Texture Segmentation System

A general texture segmentation system is shown in Figure 2.2. The image containing two or more different textures is digitised into a $M \times N$ matrix whose elements have possible integer values from $\{0, 1, \dots, G - 1\}$ where G is the number of grey levels.

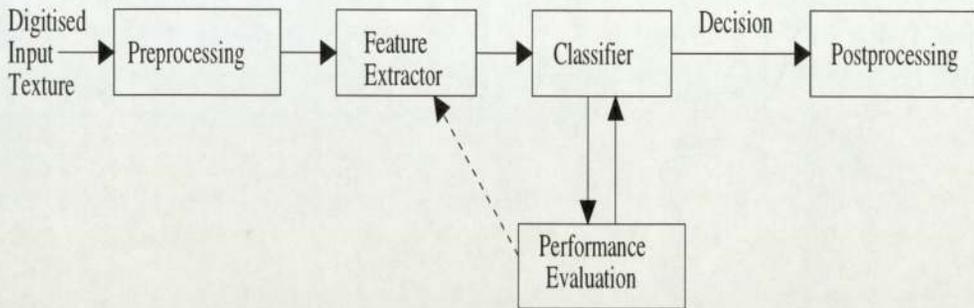


Figure 2.2: A general segmentation system

The main stages of the segmentation process are preprocessing, feature extraction, classification and finally postprocessing.

In general the preprocessing stage can involve one or more transformations being carried out on the original input variables, in this case an $M \times N$ matrix of pixel grey levels, to generate a new set of output variables, an $\tilde{M} \times \tilde{N}$ matrix. Such transformations are carried out for one of two main reasons. Firstly, they can be used to include prior knowledge about the desired form of the solution, which can lead to a significant improvement in performance [1]. Prior knowledge is often used to make the segmentation process invariant to certain effects. For example, invariance to scale, position and rotation are common in many image segmentation applications.

Another reason for employing preprocessing transformations is to enhance or restore images in such a way as to improve the discrimination ability of the extracted feature vectors. For example, the grey level histogram of an image can be manipulated in several different ways to increase its contrast. This can improve the overall resolution of the image and amplify textural differences. Noise removal is an important image enhancement technique. Images may suffer from impulse noise which is additive or multiplicative noise such as from variable illumination of the scene. These noise sources can be reduced by using different filtering techniques. Restoration involves

correcting specific damage suffered by the image [14]. For instance correcting for geometric distortions or blurring of images. A large volume of literature exists on the image processing methods mentioned above, Petrou and Bosdogianni [14] provide a good source of reference.

Section 3.3.2 details the preprocessing transformations deemed necessary for this work based on an examination of the image set to be used, see Section 2.3.1, and the objectives of the segmentation process.

In principal the aim of feature extraction is to represent the original data with as few variables as possible whilst retaining maximum discriminatory information. The most suitable texture feature extraction technique for a given problem will depend largely on the data. The aim is for all feature vectors captured from one texture to form a cluster and the clusters corresponding to different texture classes to be separated in some sense [4]. The performance of four different texture feature extraction techniques is investigated in Section 3.3.

For segmentation each pixel has to be classified, so ideally we would have a feature vector for every pixel. One method is to generate each feature vector from an $M \times M$ subimage block with the pixel of interest in the centre. This is a standard approach used by several researchers for texture feature extraction [4, 22, 7, 20]. However, one must be aware of the trade-off between discrimination ability and boundary detection. Large block sizes will tend to give better discrimination at the expense of accurate boundary detection and vice versa for small block sizes. The effect of using different block sizes on discrimination ability is investigated and discussed in Chapter 4.

In a general texture classification problem the aim is to assign a previously unseen texture pattern x to the correct class j , where $1 \leq j \leq k$ and there are a total of k distinct classes. The classifier is designed using a set of training texture patterns from each class. The performance of the classifier is evaluated by measuring a suitable error rate for classifying a set of labelled test patterns. This process will also validate the feature extraction stage.

Supervised classification techniques fall into two main categories. Methods based

on modelling the density of the data directly (parametric and non-parametric methods) and those which use discriminant functions (linear and non-linear) to minimise a predefined cost function without modelling the density of the data first. The choice of classifier will depend largely on the data. Its success will be judged on its ability to segment the feature space into disjoint areas containing the different classes. A detailed presentation of a range of classifiers is given by Webb [23].

Postprocessing involves carrying out transformations on the output variables from the classifier. This allows prior information regarding the form of output solution required to be incorporated into the segmentation process. For example in an image segmentation problem we may have prior knowledge regarding the minimum size of the object to be recognised. Any objects identified by the classifier which are smaller than this can be disregarded as misclassifications, providing a smoother segmentation result. In a more complicated system there may exist a bank of classifiers and the postprocessing stage used to carry out a rule based combination of the outputs variables into one final decision. This was the approach used by Busch [3] for segmenting multi-modal magnetic resonance images of the brain. The postprocessing stage is an important step in many practical applications which can lead to a significant improvement in the performance of the system. Section 3.3.3.2 details the postprocessing transformations implemented in this work.

2.3 Textural Segmentation of Lake Images

2.3.1 Description of Image Set

The segmentation algorithm will be tested on five lake images taken using a Sony digital video camera. Each image is 576 by 768 pixels with 8 bits/pixel. The original images were in colour, represented with three matrices which specified the intensity levels of red, green and blue for each pixel. The images were converted to grey scale by averaging the three intensity values and rounding the result to the nearest integer. This produced a grey level between 0 and 255 for each pixel. This approach was chosen for

its ease of implementation. Another method of converting a colour image to grey scale is to carry out principal components of the image using the three spectral components, red, green and blue (RGB). The RGB vector for each pixel is then projected onto the first principal component, producing a grey scale image with maximum contrast. However, this method is computationally much more intensive. The test images are shown in Figure 2.3 and Figure 2.4 along with their grey scale histograms.

As discussed in Section 2.1 natural images have a far greater degree of variability than those of man-made environments. This is certainly the case for the image set presented in Figure 2.3 and Figure 2.4. The images contain a range of textures and are taken from different angles with varying zoom and are of different locations. All of this makes the image segmentation problem more difficult. In this section we will draw attention to specific sources of variability in the image set. These can be summarised as follows;

- Different size, shape and tone of slicks
- Differences in scale, i.e zoomed in or out images
- Different wave textures due to local wind and scale effects
- The presence of shadows, reflections and glint due to the level and direction of sunlight

We have outlined the slick regions of image 2 in Figure 2.5. There are four main regions of slick, each has a different size and shape. The slicks all have the same smooth homogeneous texture. However, there are also water regions which do not contain any surface pollutants and yet have the a smooth homogeneous texture. These regions are marked as 1 and 2 in Figure 2.5. At this stage we would not expect our segmentation process to be able to distinguish these regions from the slick areas we have highlighted. The texture of normal turbulent waves is varied as can be seen by comparing the different test images in Figure 2.3 and 2.4. There is also a difference in scale from image to image. Image 4 has been taken with a greater zoom than image 2. We can

also notice a change in scale as we move from the foreground to the background in some of the images. This is certainly the case for image 1. The varied light conditions under which the images have been taken also present potential difficulties. Image 5 contains glint as it has been taken in direct sunlight.

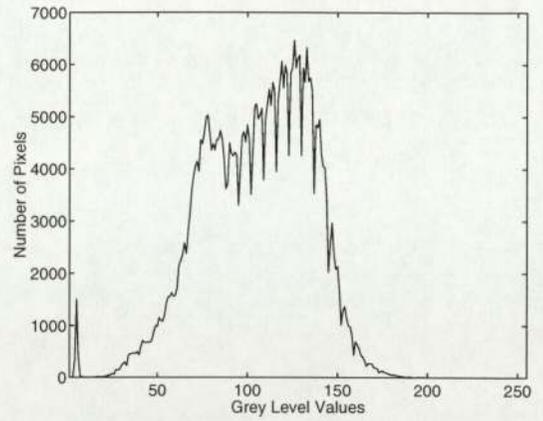
2.3.2 Discussion of Adopted Approach

As we have seen from the discussion in Section 2.3.1 the size, shape and tone of the slick varies depending on its source and the wind conditions. However, the slick's texture exhibits an invariance property. It retains a smooth homogeneous texture under a range of environmental conditions, as we have seen from the image set. The texture of normal waves does not share this invariance property, indeed the reverse has shown to be true. This relationship between the texture of slick and normal turbulent waves is an important feature which can be used for segmentation. It suggests that a classifier trained using a novelty detection approach would be well suited to this problem. Such a classifier would detect non-slick textures as outliers based on a statistical model of normal slick texture in feature space. The alternative approach would be to detect slick textures as outliers based on a statistical model of normal wave textures in feature space. This would clearly require a much more complicated model due to the range of different textures possible for normal turbulent waves.

A range of classification techniques were discussed in Section 2.2.2. In this section we have discussed image features and how they will effect the classifier design. In conclusion, a novelty detection classifier trained to recognise non-slick textures as outliers is deemed a suitable approach, and has be developed further in this work. Further evidence for the suitability of this approach is presented in Chapter 3.



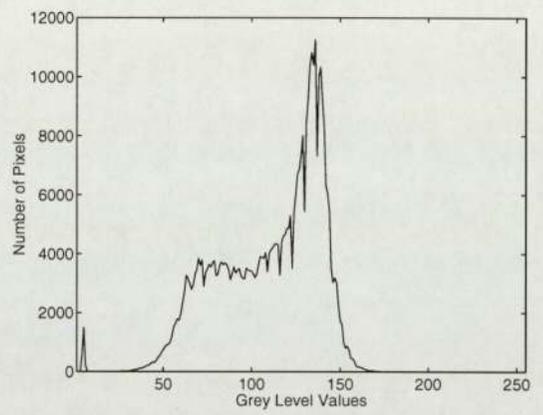
(a) Image 1



(b)



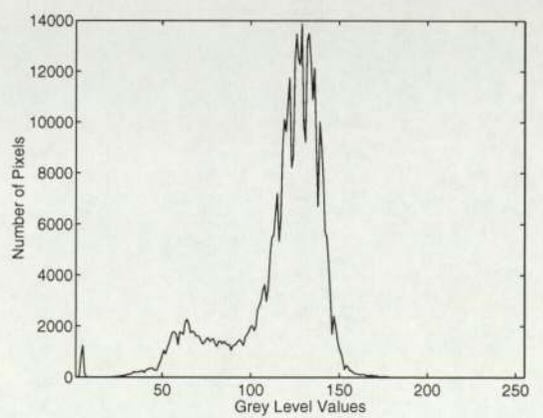
(c) Image 2



(d)

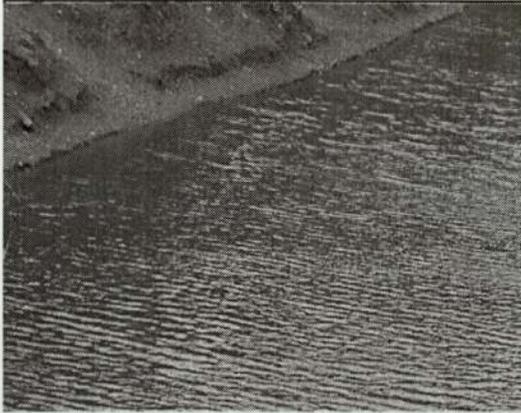


(e) Image 3

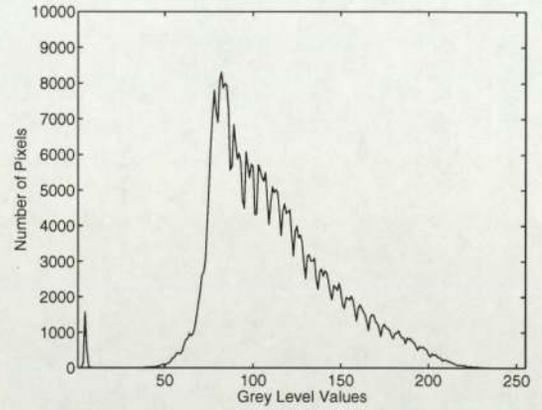


(f)

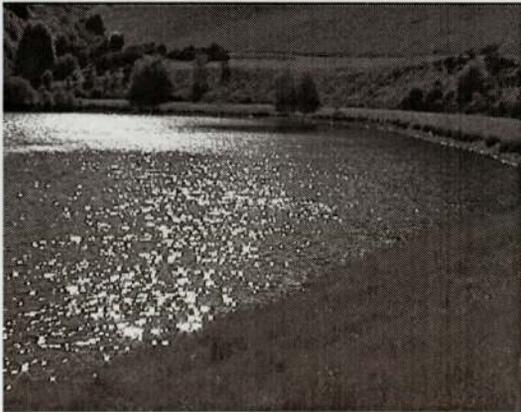
Figure 2.3: Test Images 1-3



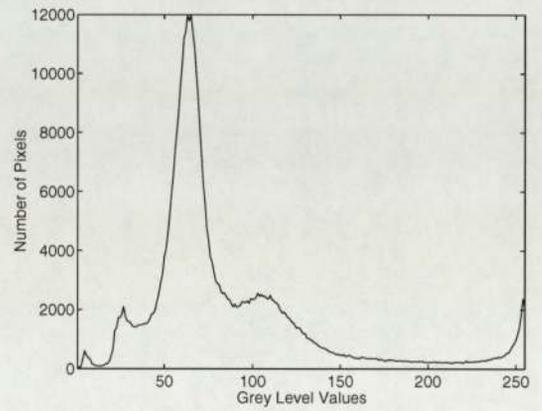
(a) Image 4



(b) Histogram of (a)



(c) Image 5



(d) Histogram of (c)

Figure 2.4: Test Images 4 and 5



Figure 2.5: Important features of image 2

Chapter 3

Methods, Segmentation Results and Analysis

3.1 Expert Labelling of Training Image

Our supervised segmentation process requires labelled slick and non-slick data. This is obtained by expertly selecting block regions from our training image which correspond to only slick or non-slick and assigning the correct label. The training image shown in Figure 3.1 was chosen as it contains a relatively large continuous region of slick which is easy to label in this block fashion. The other images in the set contain smaller, disjoint slick regions of irregular shape which would be difficult to label using this approach.

The areas highlighted in the training image have been labelled as follows. Region 1 and 2 are slick, data from region 1 will be used to train the classifier and data from region 2 will be used for testing. Data from regions 3 will be used to test the classifiers ability to correctly label non-slick textured areas. Each highlighted region is then formed into a set of non-overlapping subimages which are $n \times n$ pixels. Feature vectors are extracted from these subimages. In this work we have investigated how the segmentation performance varies for 4×4 , 8×8 and 16×16 subimages. Obviously the number of subimages obtained from a labelled region will decrease as the block size increases. Table 3.1 states the number of subimages obtained from the three regions



Figure 3.1: Expert labelling of image regions

Block Size	Region 1: Slick	Region 3: Wave
4×4	512	4096
8×8	128	1024
16×16	32	256

Table 3.1: Number of subimages obtained from training image for different block sizes

for different block sizes. Region 1 and 2 are 32 by 256 pixels and region 3 is 256 by 256 pixels.

It is clear that the number of training examples falls sharply as the block size increases. One needs to be aware that this could lead to our classifier being overfitted to the training data, especially when using 16×16 subimage blocks. The limited test data should help to provide some clues as to whether this is the case. More training examples could be obtained from the same regions by overlapping the subimage blocks. However, this would lead to correlations within the test data and was therefore not deemed to be a suitable approach.

3.2 Segmentation by Histogram Thresholding

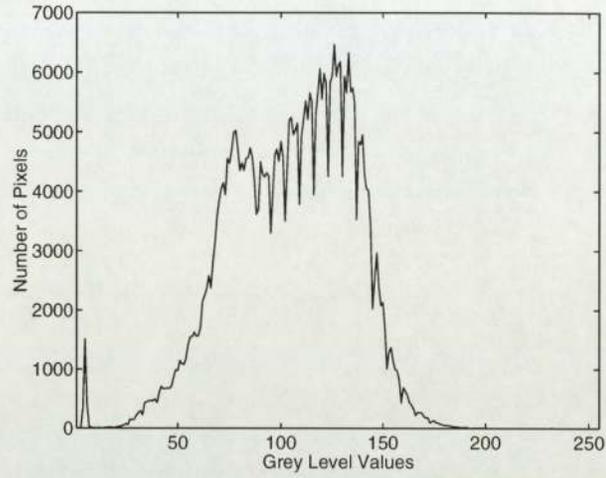
Histogram thresholding is a standard image processing technique used to identify objects in man-made scenes. This method works best for well defined images where a dark object lies on a light background or vice versa. We have employed this method on our image set to demonstrate its limitations when applied to natural scenes containing different textural regions. Figure 3.2(a) shows the grey level histograms for the complete training image and Figure 3.2(b) shows the histogram for region 1 of the training image which is labelled as slick. Segmentation using histogram thresholding relies on selecting a range of grey values which correspond to the object or region of interest. The image is then segmented by identifying all pixels with grey values within this range.

The grey levels in region 1 range from 49 to 113 which is indicated by the solid lines on Figure 3.2(b). We have chosen to set thresholds at 60 and 90, this range contains 97% of the grey values in region 1. These thresholds are used to classify any pixel with a grey value less than 60 or above 90 as non-slick. Figure 3.3(b) and 3.3(d) show the segmentation achieved for image 3 and image 5 respectively, by applying these thresholds. All pixels classified as non-slick are coloured in black all other pixels keep their original grey values.

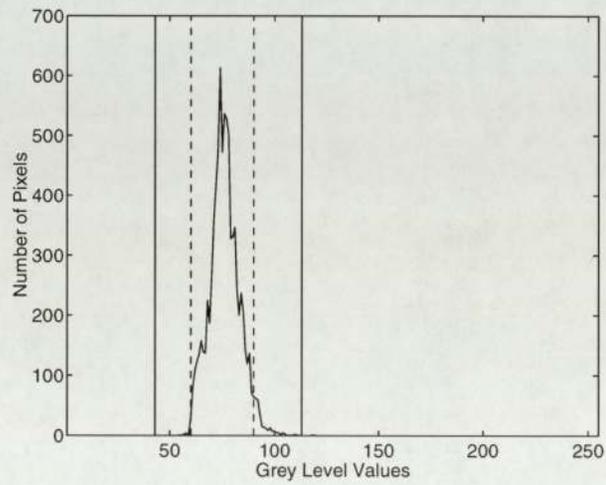
We are unable to segment the slick regions from the non-slick regions using this method. The slick regions highlighted by arrow A in Figure 3.3(b) are incorrectly classified as non-slick as their grey values fall outside the range we set earlier. A similar problem is observed in Figure 3.3(d), where water areas are incorrectly classified as slick and slick regions such as highlighted by arrow B are incorrectly classified as non-slick.

This method fails because the slick regions in the test images do not contain the same range of grey values as the slick regions in the training image, even though they have the same texture. This suggests a textural segmentation process should be independent of the average intensity of a local region if it is to successfully classify textures in natural images, which will be subject to variations in the level of illumination.

A comparison is made of the segmentation results obtained using histogram thresh-



(a)

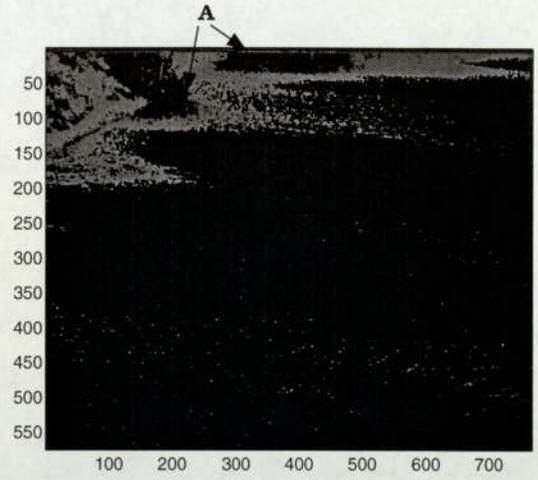


(b)

Figure 3.2: (a) Histogram of training image (b) histogram of slick region only



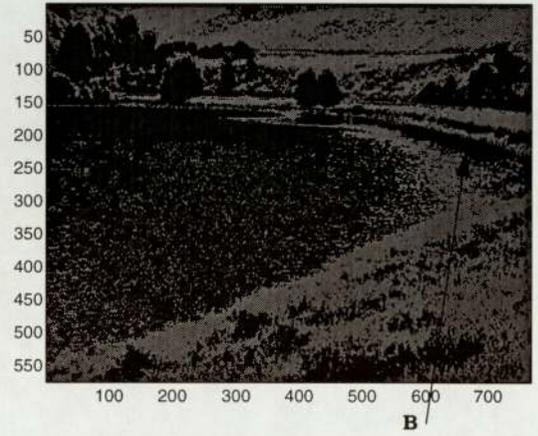
(a) Image 3



(b) Segmentation of (a)



(c) Image 5



(d) Segmentation of (c)

Figure 3.3: Segmentation of images 3 and 5 using histogram thresholding

olding and feature based methods in Chapter 4 for the test images.

3.3 Feature Based Segmentation

3.3.1 Introduction

In this section we describe the techniques employed in our feature based texture segmentation process. We have investigated four feature extraction methods namely, statistical moments, principal components analysis, one-dimensional fast Fourier transforms (1D FFT) and the two-dimensional fast Fourier transforms (2D FFT). Each method relies on extracting feature vectors from 4×4 , 8×8 or 16×16 subimages. These dimensions were deliberately chosen to be powers of two so 1D FFT and 2D FFT algorithms could be used. These are very efficient implementations of the one and two dimensional discrete Fourier transforms. For example, the one dimensional discrete Fourier transform of N points can be computed in $\mathcal{O}(N^2)$ operations, the 1D FFT is an $\mathcal{O}(N \log_2 N)$ process which is an immense difference. See Press *et al* [16] for a detailed description of how the FFT is implemented.

All these feature extraction techniques, apart from the 2D FFT, operate on vectors rather than two dimensional arrays. Vectors were obtained from the subimages by concatenating the columns. A similar approach was used by Tolba [22] who used concatenation of both rows and columns to generate vectors from subimages.

In the remainder of this thesis we refer to subimages being classified as either slick or non-slick, by this we mean classification according to whether they have slick or non-slick like texture.

3.3.2 Preprocessing of Images

As we discussed in Section 2.3.1 the tone of the slick can vary from image to image. This observation was supported by the results we obtained for segmentation based on histogram thresholding in Section 3.2. We ensure the feature vectors will be invariant to changes in brightness of the image by making the mean of each subimage equal to

zero.

The contrast will also vary between images, this is defined as the range between the maximum and minimum grey levels in the image. We chose to enhance the contrast of each image by stretching the histogram of the whole image to the full grey range of 0 to 255 as shown in Figure 3.4.

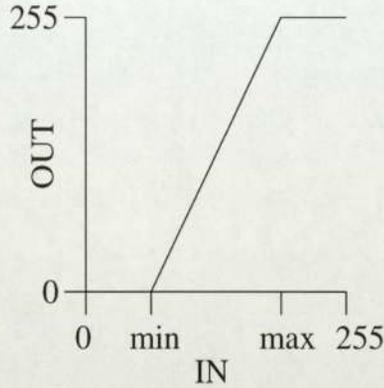


Figure 3.4: Contrast enhancement

This is achieved by applying the following simple transformation to each pixel,

$$out = round \left[\frac{(in - min) * (G - 1)}{max - min} \right] \quad (3.1)$$

where G is the number of grey levels and the result has been rounded to the nearest integer. Contrast enhancement improves the overall resolution of the image, which in turn will accentuate the difference between smooth and non-smooth textured regions. Petrou [14] discusses several contrast enhancement methods based on histogram manipulation. Such methods work best when applied to the image as a whole. Local contrast enhancement can cause distortion in smooth parts of an image which contain a narrow range of grey levels. Stretching the histogram of such local regions will amplify every small variation, making the original smooth area appear rougher.

As we saw from our discussion in Section 2.3.1 the test images are subject to variations in scale as we move from the foreground to the background. We have not implemented any preprocessing to take this effect into account for this study.

3.3.3 Statistical Moments

One way of viewing each $M \times M$ subimage from our training set, is as a set of M^2 independent random samples from a given one dimensional distribution. Intuitively we would expect the distribution of samples from a slick region to be different to that obtained from a region of turbulent waves. The most obvious difference we would expect is in the variance, expecting a lower variance for the slick data. Figure 3.5(a) and 3.5(b) show the histograms of the pixel values from region 1 and region 3 of the training image, which correspond to slick and turbulent waves respectively.

We can characterise these distributions in a compact form by their moments. The mean, variance, skewness and kurtosis being the first four moments. However, as we have chosen to zero mean each subimage we form a feature vector based on only the second, third and fourth moments. Form each $M \times M$ subimage we obtain a set of grey values x_1, \dots, x_N where $N = M^2$. The variance of these values is defined as,

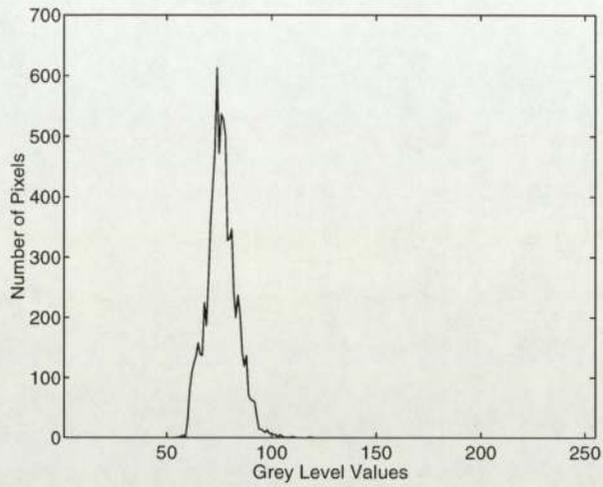
$$Var(x_1, \dots, x_N) = \frac{1}{N-1} \sum_{j=1}^N (x_j)^2. \quad (3.2)$$

The skewness characterises the degree of asymmetry of a distribution. Positive values of skewness indicate a distribution with an asymmetric tail extending out towards more positive x , and negative values indicate a distribution whose tail extends towards more negative values of x [16]. Skewness is defined as,

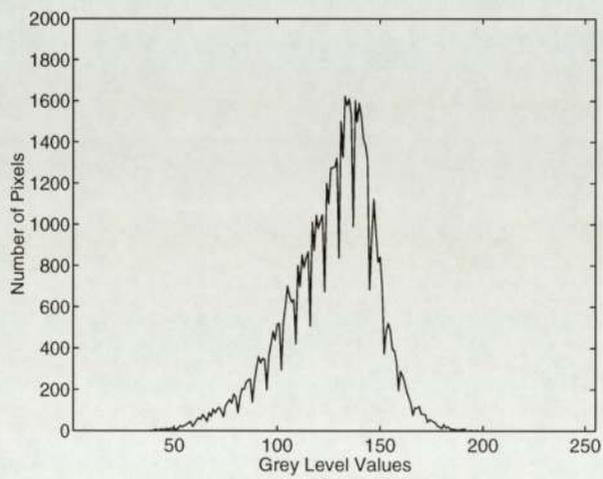
$$Skew(x_1, \dots, x_N) = \frac{1}{N} \sum_{j=1}^N \left[\frac{(x_j)}{\sigma} \right]^3 \quad (3.3)$$

where

$$\sigma(x_1, \dots, x_N) = \sqrt{Var(x_1, \dots, x_N)}. \quad (3.4)$$



(a)



(b)

Figure 3.5: (a) Histogram of slick and (b) wave data from training image

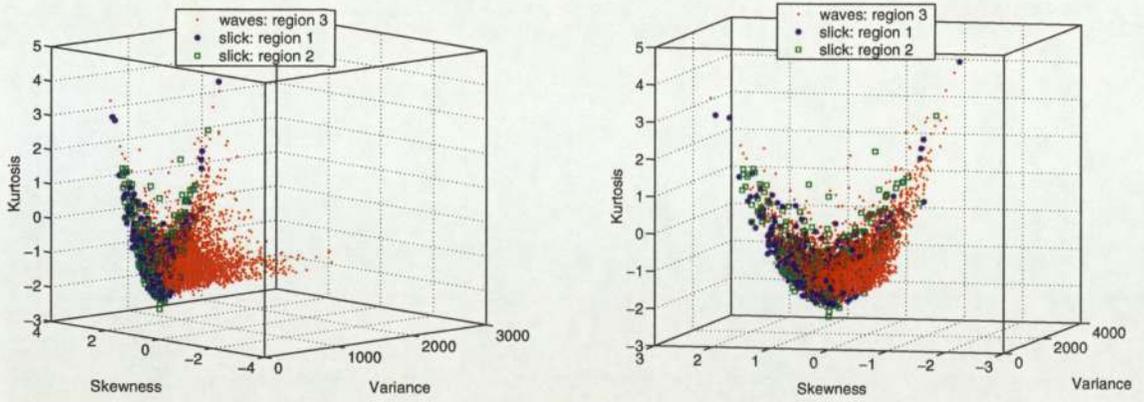
Finally, kurtosis the fourth moment measures the peakedness or flatness of a distribution relative to a Gaussian distribution. Kurtosis is defined as,

$$Kurt(x_1, \dots, x_N) = \left\{ \frac{1}{N} \sum_{j=1}^N \left[\frac{(x_j)}{\sigma} \right]^4 \right\} - 3. \quad (3.5)$$

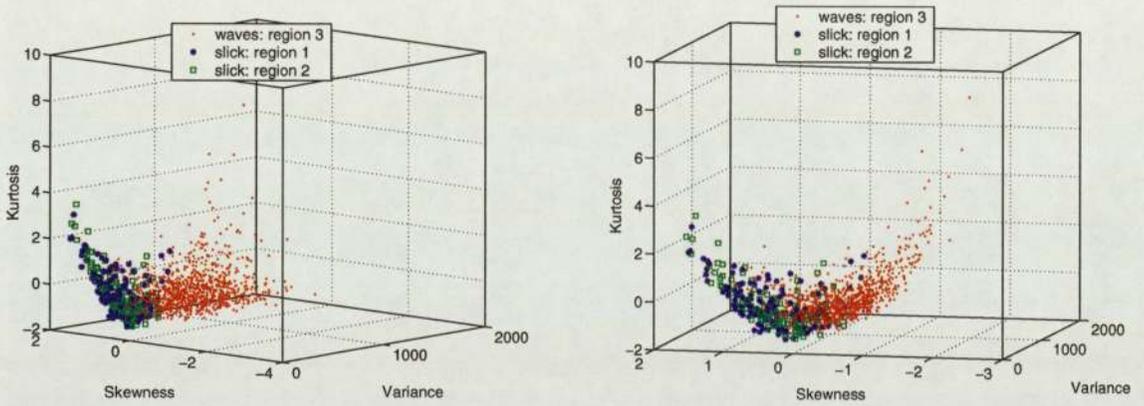
Figure 3.6 shows three dimensional plots of variance, skewness and kurtosis for the expertly labelled slick and wave data of the training image, obtained using 4×4 , 8×8 and 16×16 subimages. It is clear that we can not separate the distribution of slick and wave feature vectors exactly using the skewness and kurtosis dimensions. However, we see that there is less overlap between the distributions along the variance dimension and this reduces as we increase the size of the subimage used. We can completely separate the two distributions when using 16×16 subimages based on variance alone. This effect can clearly be seen in Figure 3.7 which shows histograms of the variance values for different sized subimages. We investigate the generalisation ability of this method in Section 3.3.3.2 by applying the technique to two test images.

We can use the histograms shown in Figure 3.7 as density models of the variance for slick and non-slick textured regions. By setting a threshold for variance these models can be used for classification. Any previously unseen subimage which has a lower variance than the threshold will be classified as having slick like texture and any subimage with a higher variance will be classified as having non-slick like texture. Changing the position of the threshold will vary the number of true positives, false positives, true negatives and false negatives. The classification matrix shown in Figure 3.8 defines these quantities. For example, true positives are the number of slick subimages correctly classified as slick as a fraction of the total number of slick subimages in the test set. Similarly, false positives are the number of non-slick subimages incorrectly classified as slick as a fraction of the total number of non-slick test cases. The true positives and false negatives sum to one, as do the false positives and true negatives.

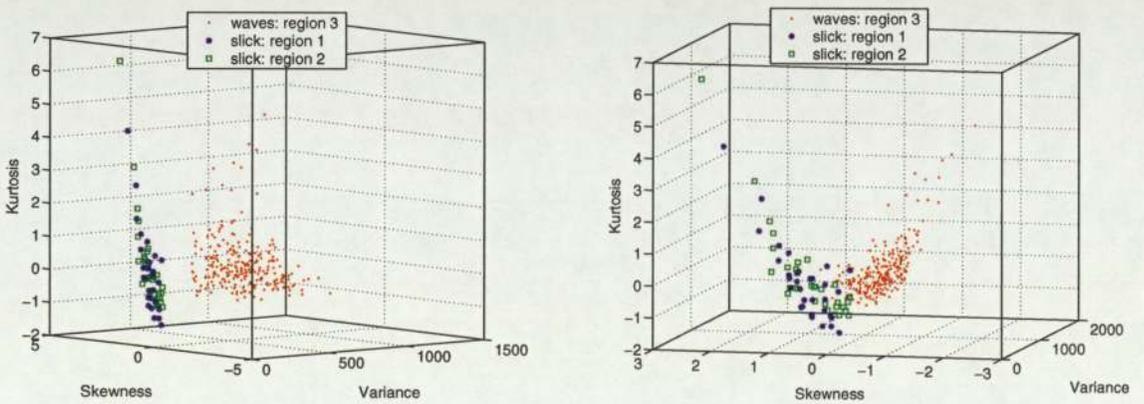
We can set the decision threshold in a principled fashion by using receiver operating characteristic (ROC) curves, as explained below.



(a) Using 4×4 subimages

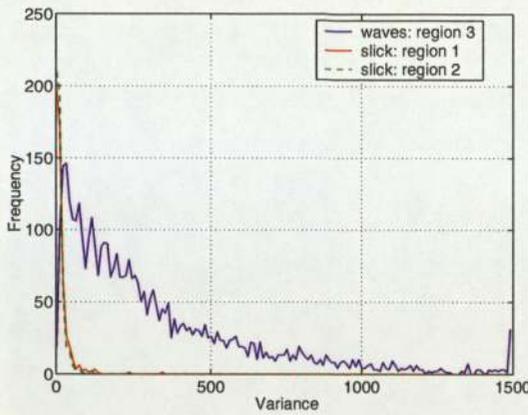


(b) Using 8×8 subimages

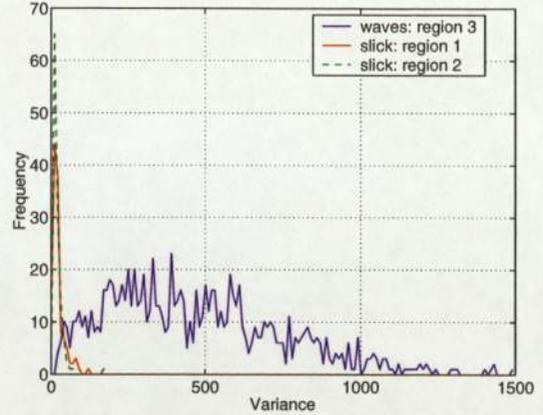


(c) Using 16×16 subimages

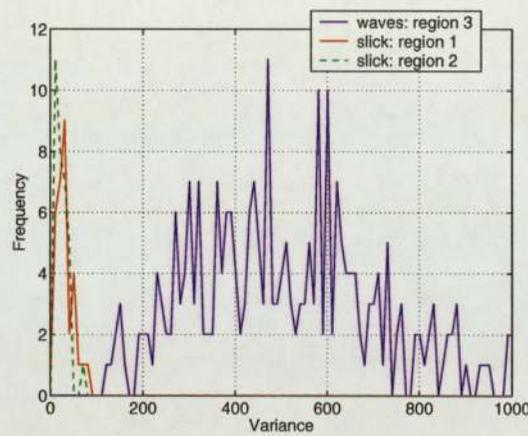
Figure 3.6: Variance, skewness and kurtosis plotted for slick and wave data



(a) Using 4×4 subimages



(b) Using 8×8 subimages



(c) Using 16×16 subimages

Figure 3.7: Histograms of variance for slick and wave data

		Predicted Class	
		C1	C2
Actual Class	C1	TP	FN
	C2	FP	TN

C1 - Class 1
 C2 - Class 2
 TP - True Positive
 FN - False Negative
 FP - False Positive
 TN - True Negative

Figure 3.8: Classification matrix

3.3.3.1 Receiver Operating Characteristic (ROC) Curves

A toy data example is given in Figure 3.9(a) which shows the distribution of likelihood values for slick and non-slick labelled subimages. By setting a threshold, we can classify any new subimage with a likelihood value greater than the threshold as slick, and those with a value lower than the threshold as non-slick. Figure 3.9(b) shows how the true positives vary with the false positives for a range of thresholds, this is called the ROC curve. The area under the curve is a measure of how well we are able to discriminate between the two classes. The closer the curve is to the left-hand and top-border, the better the discrimination ability. The 45-degree line represents the case for two completely overlapping distributions, in which case it is not possible to discriminate one class from the other.

In this work we have assumed it is equally bad to misclassify a slick subimage as non-slick, as it is to misclassify a non-slick subimage as slick. However, if required, a different penalty could be assigned to each possible misclassification by using a loss/cost matrix. This would in turn alter the decision rule used to classify each subimage.

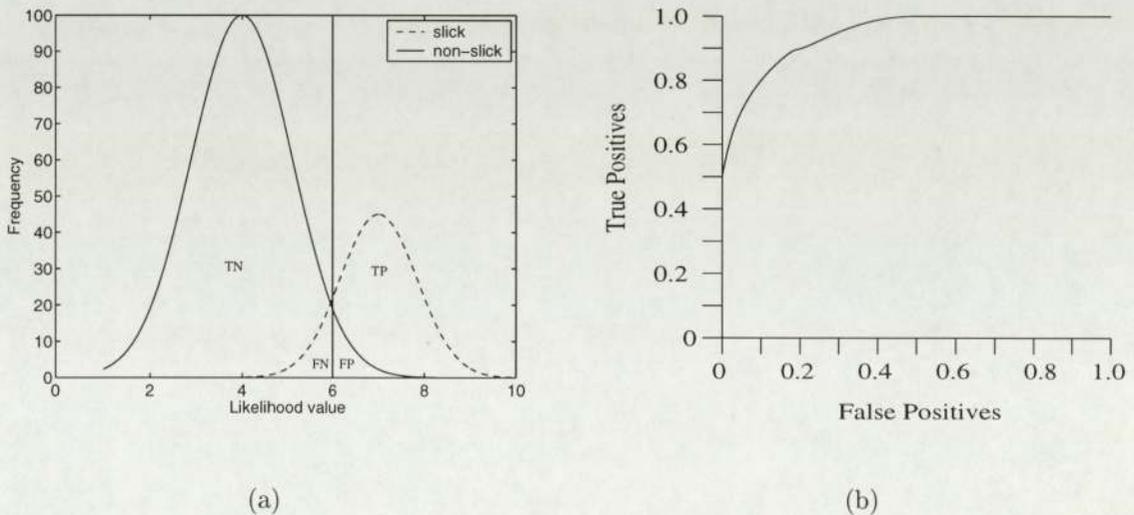


Figure 3.9: Toy data example, (a) histogram of likelihood values for labelled slick and non-slick data, (b) resulting ROC curve

3.3.3.2 Results

The ROC curves obtained for the variance values of the labelled data from the training image are shown in Figure 3.10. Data from regions 1 and 2 was used as training and test cases respectively, for slick texture, data from region 3 was used as the non-slick texture case. Section 3.1 describes the expert labelling of the training image.

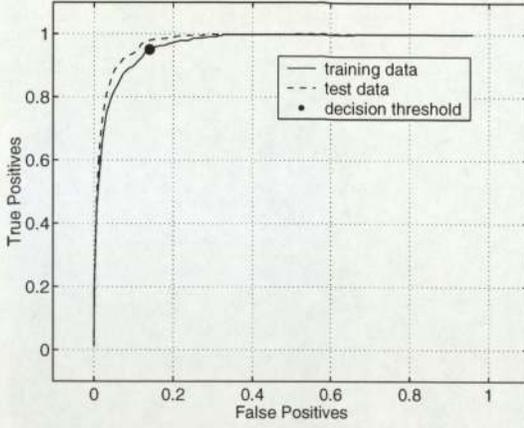
The area under the ROC curve increases as we use larger subimages, indicating better discrimination ability. We have marked the position of the chosen threshold value on each curve. In each case the threshold was chosen where the tangent to the curve is approximately 45-degrees, sometimes referred to as the ‘knee’ of the curve. Beyond this point the false positives begin to increase at a faster rate than the true positives. Table 3.2 gives the true and false positives rates achieved with the chosen threshold for different sized subimages. We will examine how the segmentation performance varies if the threshold is chosen at different points along ROC curve in Section 3.3.4.2.

Block size	Training data		Test data	
	TP	FP	TP	FP
4×4	0.95	0.14	0.98	0.14
8×8	0.99	0.05	0.99	0.05
16×16	1.0	0	1.0	0

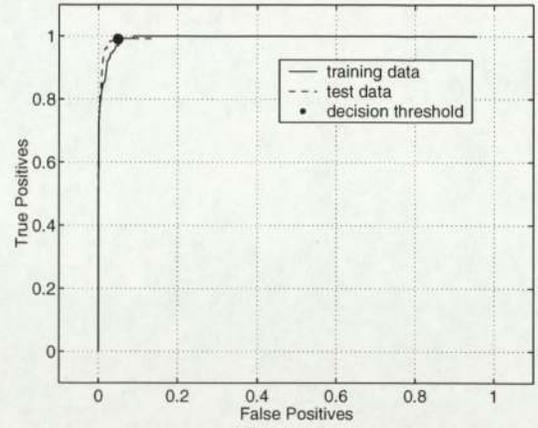
Table 3.2: Classification rates using statistical moments

We will now present and discuss the segmentation results achieved for images 2 and 4 from our test set. All results are based on the thresholds set for the training image. Ideally each pixel would be classified based on a feature vector extracted from a subimage, with the pixel of interest in the centre. However, computationally this would be very intensive. We have chosen to generate a feature vector for each 2 by 2 pixel block, this still allows fine segmentation but with a reduced computational burden. Pixels classified as belonging to non-slick textured regions are coloured in black, those classified as belonging to slick textured regions keep their original grey values.

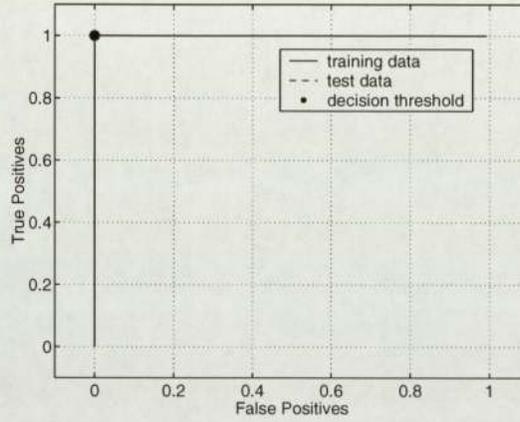
The segmentation results achieved for image 2 using 4×4 , 8×8 and 16×16 subimages are shown in Figure 3.11. The final stage in our segmentation process is



(a) Using 4×4 subimages



(b) Using 8×8 subimages



(c) Using 16×16 subimages

Figure 3.10: ROC curves for variance values of training data

postprocessing. Figure 3.11(b) and 3.11(c) show the segmentation results obtained using 4×4 subimages, before and after postprocessing respectively. We have many false positives in the lower part of the Figure 3.11(b) which contains normal waves. We discuss the reasons for these false positives in Section 3.3.3.3. However, we can observe that many of these false positives are isolated cases, or cases where only one or two neighbouring 2 by 2 pixel blocks have the same class label. Our postprocessing step involves using a 6 by 6 pixel smoothing window to remove these isolated misclassifications. Each 2 by 2 pixel block within the window will have one class label, we consider the class of the central block and its eight neighbours. If there are three or

more neighbouring blocks with the same class label as the central block it will keep its original class label. If not it will change class. The value three was chosen empirically. This produced the best balance between an effective reduction in the number of isolated misclassifications, and preserving boundaries of correctly identified regions. The smoothing window is scanned across the whole image moving across two pixels at a time, until the edge of the image is reached, then down by two pixels and so forth.

Strand [20] used a similar technique for postprocessing in which the central pixel was assigned to the most dominant class in the smoothing window.

All future segmentation results presented in this thesis have been postprocessed in this manner.

Figure 3.12 shows the segmentation results achieved for image 4 using 4×4 , 8×8 and 16×16 subimages.

3.3.3.3 Analysis

The segmentation results for image 2, shown in Figure 3.11 show an improvement as we use larger subimages. The most dramatic effect is on reducing the number of misclassifications in the lower part of the image which contains a low contrast fine wave texture. A similar effect was observed for the labelled training data where the false positive rate decreased from 0.14 to 0.05 for the wave data, as the subimage size was increased from 4×4 to 8×8 . However, the increase in the true positive rate for the slick data was less significant, changing from 0.95 to 0.99. We suggest this effect is due to the fact that a 4×4 subimage is not large enough to capture the basic structure of the wave texture. By structure we mean the characteristic local variations in grey levels.

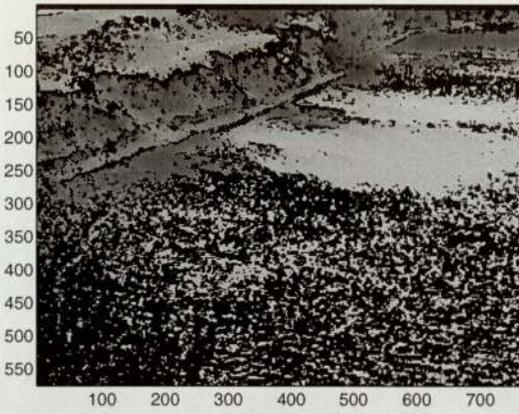
The smooth water areas in the top half of the image are classified as having slick like texture as we would expect from our earlier discussion in Section 2.3.1.

These results also display how boundary detection becomes courser as we use larger subimages.

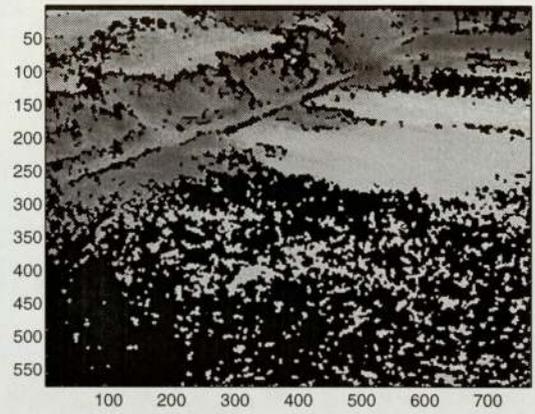
The segmentation results for image 5, shown in Figure 3.12 show a less dramatic



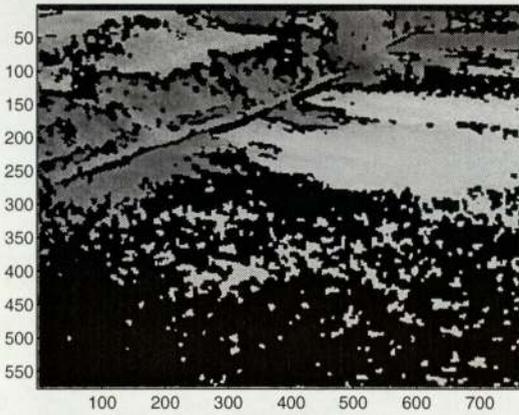
(a) Image 2



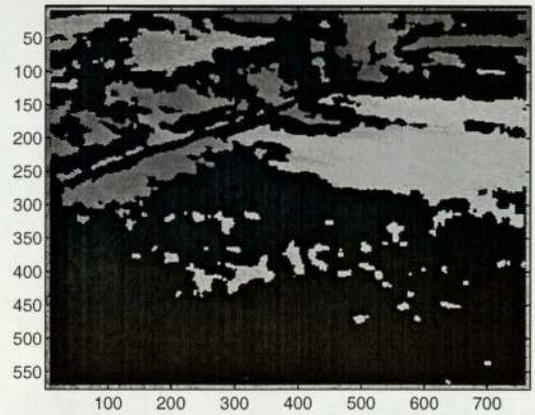
(b) Segmentation of (a) using 4×4 subimages before postprocessing



(c) Segmentation of (a) using 4×4 after postprocessing

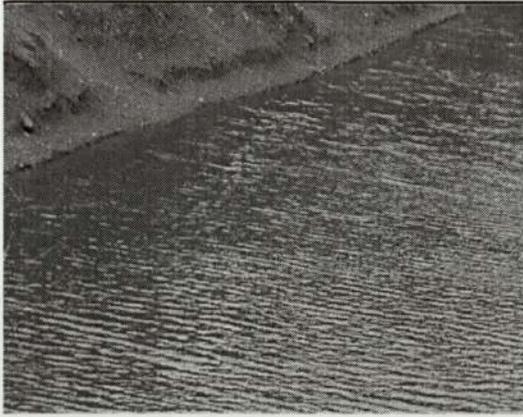


(d) Segmentation of (a) using 8×8 subimages after postprocessing

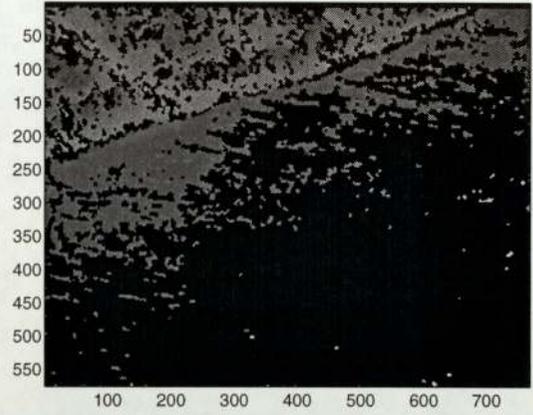


(e) Segmentation of (a) using 16×16 subimages after postprocessing

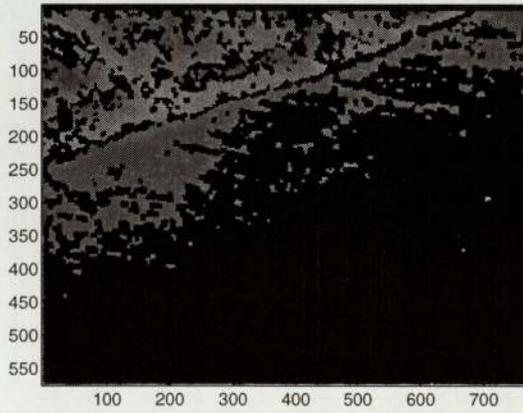
Figure 3.11: Segmentation of image 2 using statistical moments



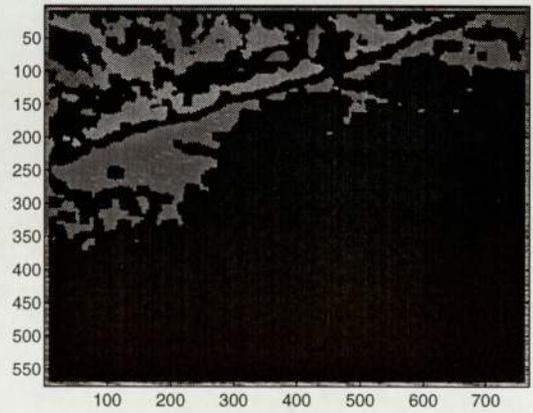
(a) Image 4



(b) Segmentation of (a) using 4×4 subimages



(c) Segmentation of (a) using 8×8 subimages



(d) Segmentation of (a) using 16×16 subimages

Figure 3.12: Segmentation of image 4 using statistical moments

improvement for larger subimages. This can be attributed to the different wave texture present. The two main differences are the increased contrast and fineness of the texture. Both these factors will lead to increased discrimination ability based on a measure of variance, and should allow the use of smaller subimages.

3.3.4 Principal Component Analysis

We consider the following scenario, we obtain a M^2 dimensional vector, x_i from each $M \times M$ subimage by concatenating its columns. We generate a population of such vectors $\mathbf{x} = (x_1, \dots, x_N)^T$ from our slick labelled data. We then consider \mathbf{x} as a random vector population and attempt to model its distribution with a parameterised multivariate probability distribution, $p(y)$. A conditional density model could be used to classify a previously unseen vector x_j based on its likelihood given this model, $p(x_j|\mathbf{x})$. For this approach to work we require the vector populations from the slick and wave data to cluster in different regions of this high dimensional space. Therefore we expect relatively large values for the likelihood of vectors which have slick like texture, compared to lower values for vectors with non-slick texture.

However, modelling the distribution of such high dimensional data requires a complicated model with many free parameters. A large amount of data is required in order to obtain robust estimates for these parameters. This approach is not feasible when dealing with a limited amount of training data, as in our case. Principal components analysis (PCA) can be employed to reduce the dimensionality of our data, enabling a simpler density model to be used. This is achieved by projecting our original m -dimensional data onto a n -dimensional subspace spanned by a set of orthogonal vectors, where n is chosen such that most of the variance of the original data is preserved. These orthogonal vectors are referred to as principal components. The first principal component is in the direction of maximum variance in the original data. The second principal component is orthogonal to the first and in the direction of maximum variance in this subspace and so on. These principal components are given by the eigenvectors of the symmetric positive definite covariance matrix Σ , and are ordered

in terms of their corresponding eigenvalues. The k^{th} principal component having the k^{th} highest eigenvalue. The covariance matrix of a sample of vectors $\mathbf{x} = (x_1, \dots, x_N)^T$ is given by,

$$\Sigma_{\mathbf{x}} = \frac{1}{N} \sum_{j=1}^N (x_j - \mu_{\mathbf{x}})(x_j - \mu_{\mathbf{x}})^T \quad (3.6)$$

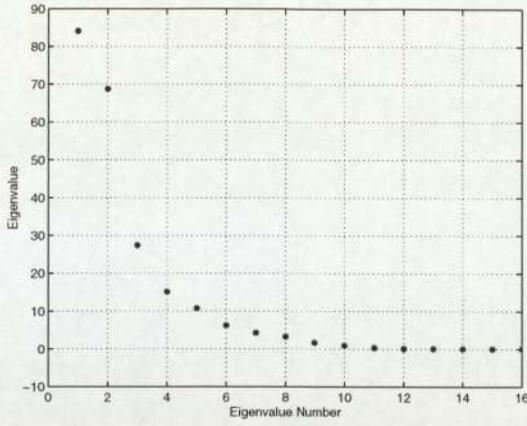
where

$$\mu_{\mathbf{x}} = \frac{1}{N} \sum_{j=1}^N x_j \quad (3.7)$$

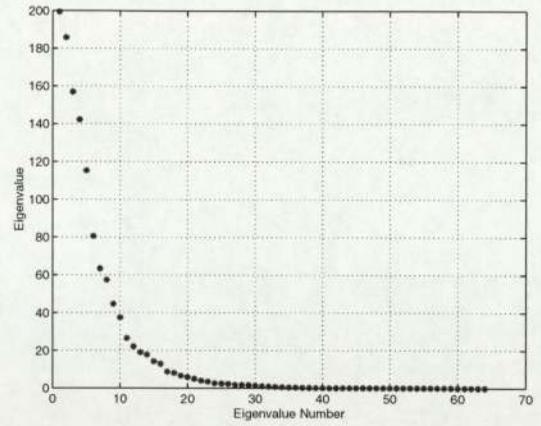
See Webb [23] for a derivation of principal components and further details on eigenvector eigenvalue decomposition of symmetric matrices. PCA would not be suitable for data which contained non-linear correlations as it may over estimate the intrinsic dimensionality of the data [1]. Another difficulty with PCA is determining the number of principal components to retain. One method is to use the eigenvalue spectrum. Figure 3.13 shows the eigenvalue spectra obtained for the slick data from region 1 of the training image, using 4×4 , 8×8 and 16×16 subimages. The eigenvalue spectrum can be viewed as separating the data into signal and noise subspaces. A cutoff point can be defined where the eigenvalues fall sharply before levelling off at small values (the ‘scree’ test) [23]. All eigenvalues below this cutoff can be viewed as representing the noise in the data and so can be neglected. This cutoff is sometimes referred to as the ‘elbow’ of the curve. This is still rather a subjective method but one which provides a good guide as to the number of principal components to retain. Table 3.3 shows the number of principal components retained by applying this test to the eigenvalue spectra in Figure 3.13.

Block Size	Number of principal components retained
4×4	6
8×8	15
16×16	25

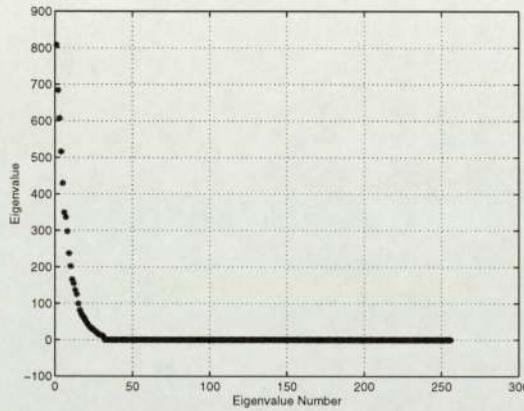
Table 3.3: PCA: Number of principal components retained for different block sizes



(a)



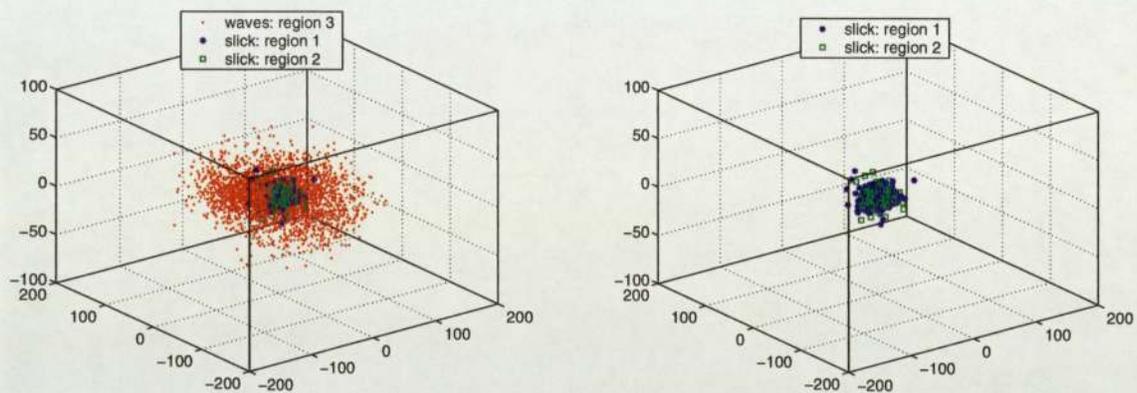
(b)



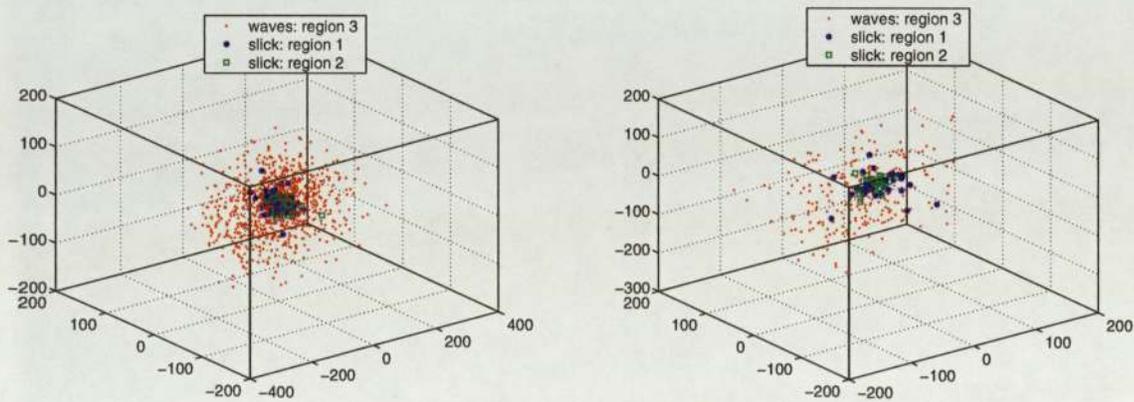
(c)

Figure 3.13: Eigenvalue spectra of slick data using (a) 4×4 (b) 8×8 and (c) 16×16 subimages

By employing PCA we have been able to significantly reduce the dimensionality of our original data. We can now form our feature vectors by projecting the original slick and wave data vectors obtained from the training image, onto the first n principal components, as given in Table 3.3. Figure 3.14 shows the distribution of these feature vectors projected onto the first three principal components, obtained using different sized subimages. The slick feature vectors are clustered together with the wave feature vectors distributed around this central cluster. The separation of the classes increases with increasing block size.



(a) Using 4×4 subimages



(b) Using 8×8 subimages

(c) Using 16×16 subimages

Figure 3.14: PCA: Slick and wave data projected onto first three principal components

3.3.4.1 Density Modelling

From the exploratory data visualisation carried in Section 3.3.4 we conclude that a single multivariate Gaussian distribution is a suitable form for modelling the distribution of the slick feature vectors. A more complicated model such as a mixture of gaussians could also be considered. However, as a first step we chose to adopt the simpler single multivariate Gaussian model. In d dimensions the multivariate Gaussian probability density function can be written as

$$p(y) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(y - \mu)^T \Sigma^{-1} (y - \mu)\right\}. \quad (3.8)$$

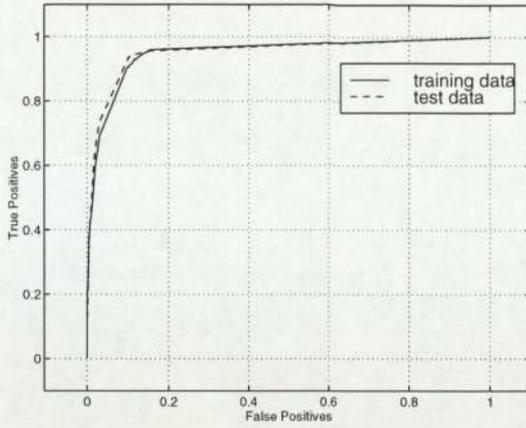
This distribution is governed by the d -dimensional mean vector μ , and the $d \times d$ covariance matrix Σ . These parameters were estimated using the slick feature vectors obtained from the training image, as defined by equation 3.7 and 3.6 respectively. In this instance the covariance matrix will be diagonal as we have removed correlations within the data.

3.3.4.2 Results

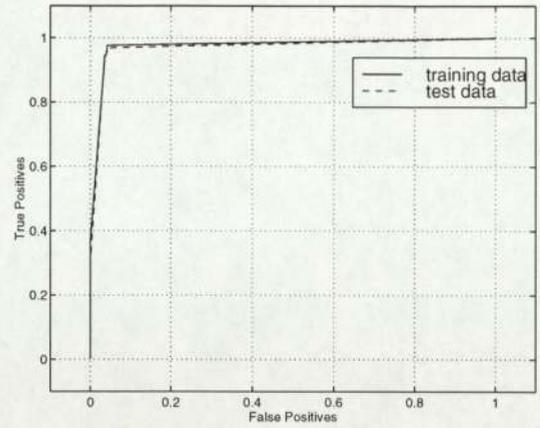
Given our conditional density model of ‘normal’ slick texture we calculated the likelihood values for all the labelled slick and wave subimages. These values were used to generate the ROC curves shown in Figure 3.15. The discrimination ability clearly increases as we use larger block sizes. The ROC curve obtained when using 16×16 subimages shows perfect segmentation was achieved of the two classes in the training set.

We experimented with choosing different threshold values and observing the effect on segmentation performance. Three different thresholds were chosen using 4×4 subimages as detailed in Table 3.4. The corresponding segmentation results are given in Figure 3.16.

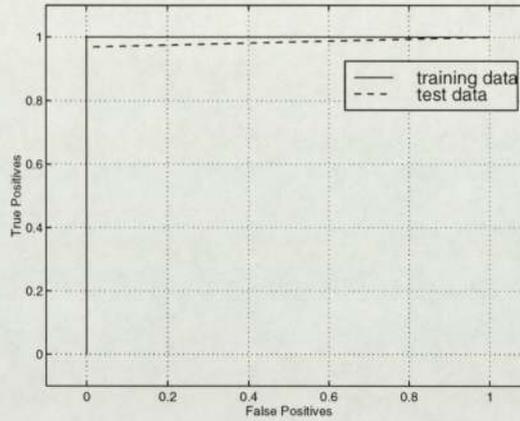
Table 3.5 gives the classification rates achieved for the training data for threshold values chosen to be at the ‘knee’ of the curve for different subimage sizes.



(a) Using 4×4 subimages



(b) Using 8×8 subimages



(c) Using 16×16 subimages

Figure 3.15: PCA: ROC curves of likelihood values of training data

Figure 3.17 and 3.18 show the segmentation results for images 2 and 3 using 4×4 , 8×8 and 16×16 subimages.

3.3.4.3 Analysis

We can see from the results in Figure 3.16 that the decision threshold chosen from the ROC curve has a significant effect on the segmentation results. By visual examination the best segmentation results would be those of Figure 3.16(b) which has the lowest false positive rate, 0.02 out of the three sets of results. However, it also has the lowest true positive rate, 0.6. This demonstrates the trade-off which can be made between the

Segmented Image	TP	FP
Figure 3.16(b)	0.60	0.02
Figure 3.16(c)	0.90	0.10
Figure 3.16(d)	0.96	0.15

Table 3.4: PCA: Segmentation results for different threshold values

Block size	Training data		Test data	
	TP	FP	TP	FP
4×4	0.96	0.15	0.95	0.15
8×8	0.98	0.05	0.98	0.05
16×16	1.0	0	0.98	0.0

Table 3.5: Classification rates using PCA

false positive and true positive rate in order to obtain the most suitable results for a given situation. For example in a segmentation system we may be prepared to accept a slightly lower true positive rate if it means that the false positive rate becomes very low.

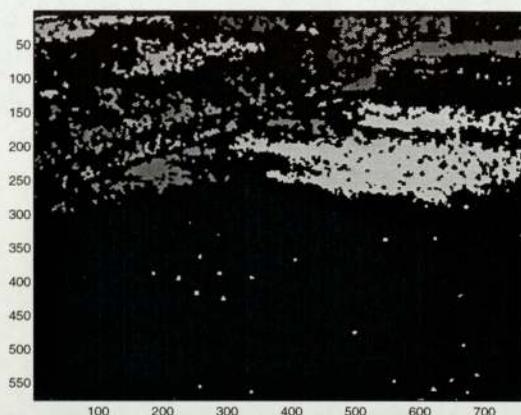
The results of Figure 3.17 and 3.18 demonstrate the effect on segmentation performance of using different block sizes. As was the case for feature vectors based on statistical moments, using PCA with small block sizes does not achieve correct segmentation of water areas which contain fine wave textures, such as in the lower part of Figure 3.17 and 3.18. The results improve with larger block size with 16×16 subimages producing the best results. However, this is at the expense of boundary detection which becomes less well defined and has led to the misclassification of small slick textured regions. These results also validate the choice of using a multivariate Gaussian density model for the slick feature vectors.

3.3.5 One Dimensional Discrete Fourier Transform

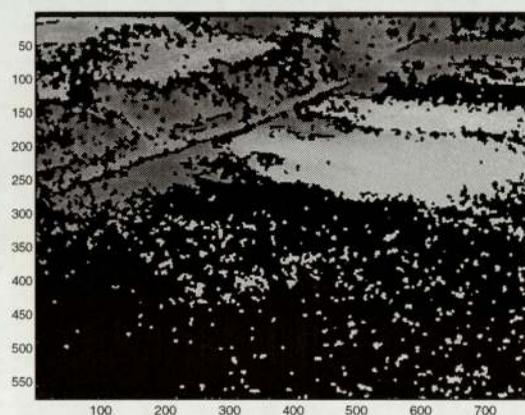
Any discrete series $\{x_n\}$, $n = 0, 1, \dots, (N - 1)$ can be decomposed exactly as a sum of discrete sinusoidal waves. The magnitude of these sinusoidal waves is given by a set of N Fourier components X_k , $k = 0, 1, \dots, (N - 1)$, which represents the discrete frequency spectrum of our original series given by,



(a) Image 2



(b)



(c)

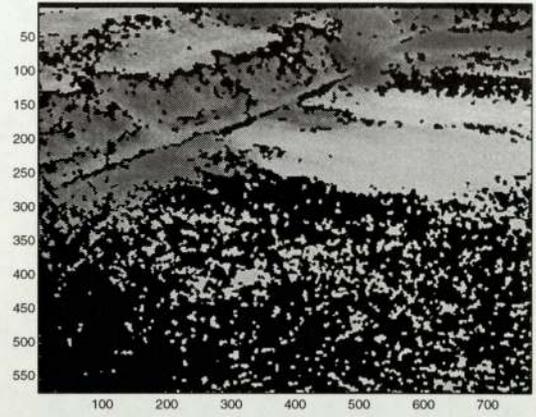


(d)

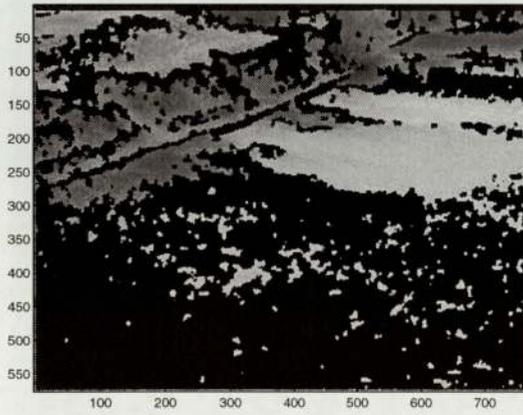
Figure 3.16: PCA: segmentation of image 2 using different threshold values



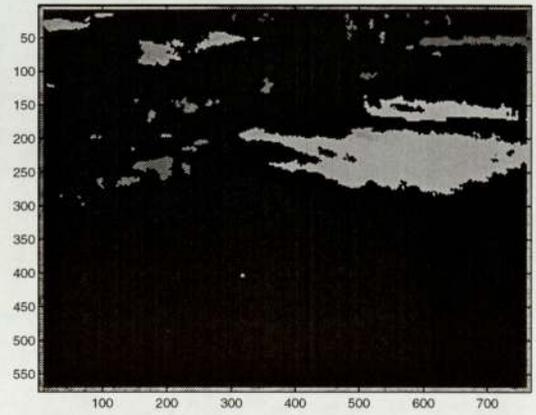
(a) Image 2



(b) Segmentation of (a) using 4×4 subimages



(c) Segmentation of (a) using 8×8 subimages

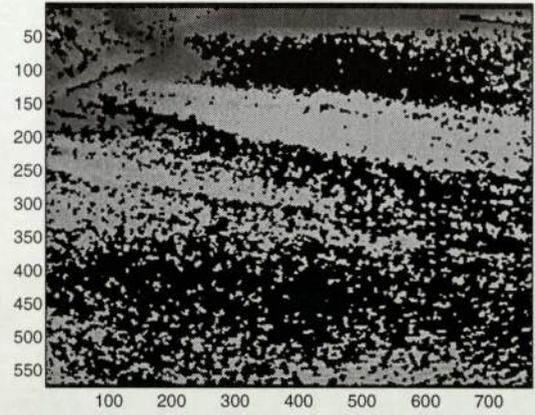


(d) Segmentation of (a) using 16×16 subimages

Figure 3.17: Segmentation of image 2 using PCA



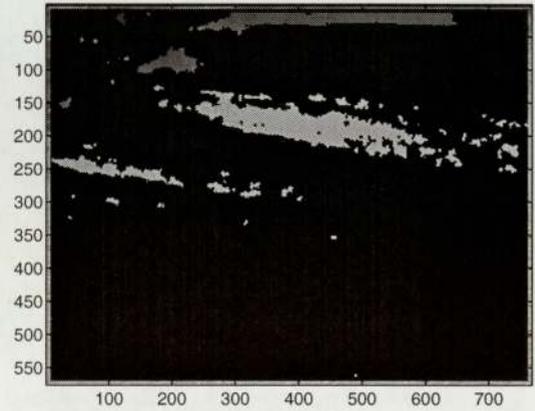
(a) Image 3



(b) Segmentation of (a) using 4×4 subimages



(c) Segmentation of (a) using 8×8 subimages



(d) Segmentation of (a) using 16×16 subimages

Figure 3.18: Segmentation of image 3 using PCA

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-j(\frac{2\pi kn}{N})}. \quad (3.9)$$

The frequency power spectrum is given by the magnitude of X_k . Essentially, we are projecting our original discrete series or N -dimensional vector $\{x_n\}$, onto an orthogonal set of basis vectors $e_k[n]$ in N -dimensional space given by,

$$e_k[n] = \exp^{-j(\frac{2\pi kn}{N})} \quad 0 < k < N. \quad (3.10)$$

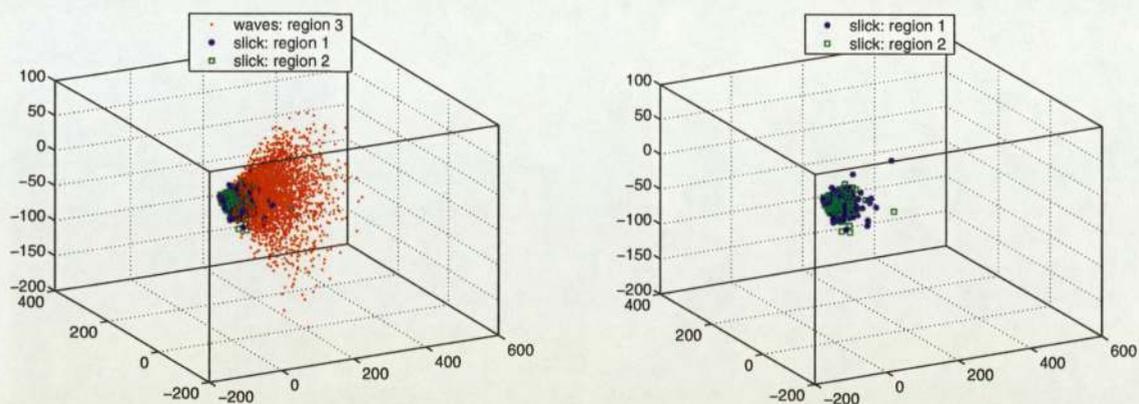
We have investigated the use of the frequency power spectrum of our subimages for texture classification. We first generated a set of vectors for our labelled slick and wave data from the training image, by concatenating the columns of each subimage. We then applied the 1D FFT to each N -dimensional vector which gave us another complex N -dimensional vector. We obtained the discrete power spectrum of each subimage by taking the magnitude of each component of this complex vector. Finally, we employed PCA to reduce the dimensionality of the discrete power spectrum, as described in Section 3.3.4. The number of principal components retained is shown in Table 3.6. It is interesting to note that for 8×8 and 16×16 subimages, the data can be summarised using fewer principal components than when PCA was applied directly to the original data.

Block Size	Number of principal components retained
4×4	4
8×8	7
16×16	12

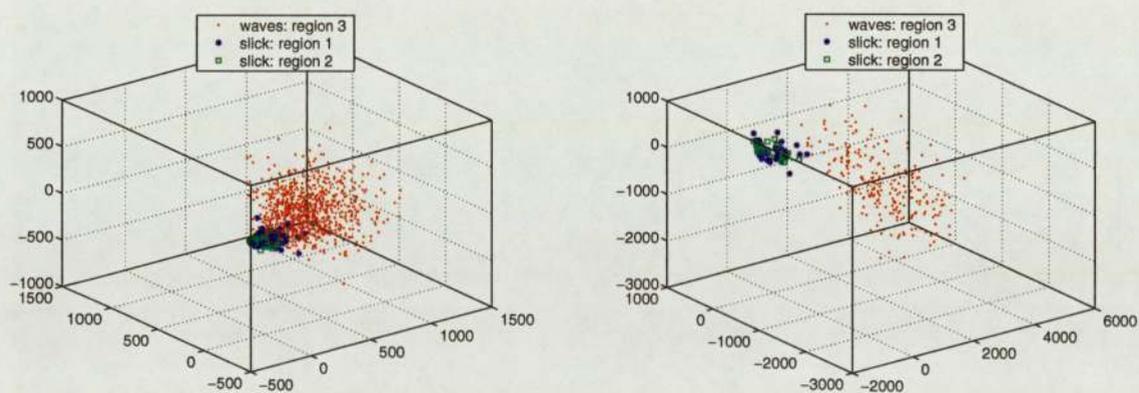
Table 3.6: 1D FFT: Number of principal components retained for different block sizes

As for the PCA method our aim is to build a conditional density model of the slick feature vectors which we can then be used for classification. However, we must first choose an appropriate parameterised form for this model. We carry out exploratory

visualisation of our slick and wave texture feature vectors by plotting their projections onto the first three principal components, see Figure 3.19.



(a) Using 4×4 subimages



(b) Using 8×8 subimages

(c) Using 16×16 subimages

Figure 3.19: 1D FFT: Slick and wave data projected onto first three principal components

These 3D plots show we are able to achieve good separation between the slick and wave classes, especially when using 8×8 or 16×16 subimages. The slick feature vectors are tightly clustered as was the case in the PCA method. Once again we conclude that a multivariate Gaussian distribution is suitable to model the slick feature vectors, see Section 3.3.4.1.

3.3.5.1 Results

As described in Section 3.3.4.2 we obtained a set of ROC curves for the likelihood values of our slick and wave training data, given by our conditional density model of ‘normal’ slick texture. In order to remain as concise as possible we do not present the ROC curves. Once again we chose the decision threshold to be at the ‘knee’ of the curve. Table 3.7 gives the classification rates achieved for the slick and wave training data using different size blocks.

Block size	Training data		Test data	
	TP	FP	TP	FP
4×4	0.95	0.12	0.93	0.12
8×8	0.93	0.04	0.98	0.04
16×16	0.88	0	1.0	0

Table 3.7: Classification rates using 1D FFT

Figure 3.20 and 3.21 show the segmentation results for images 2 and 4 using 4×4 , 8×8 and 16×16 subimages. These results are based on the threshold set for the training data.

3.3.5.2 Analysis

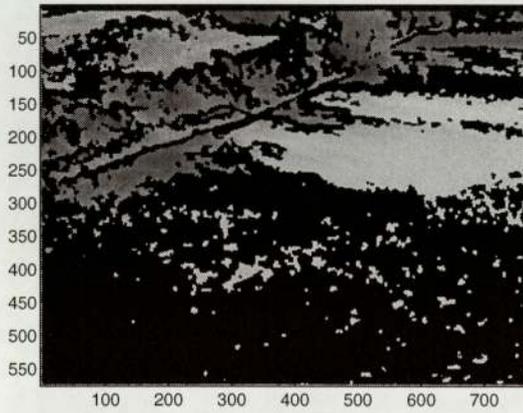
The 1D FFT results follow the same general pattern that has emerged from the results for statistical moments and PCA. The segmentation performance improves with larger block sizes. However this improvement is largely due to the reduction in the number of false positives. However, this was accompanied by worse boundary detection for the PCA method. In comparison the 1D FFT method achieves better boundary detection using 16×16 subimages. See Figure 3.20(d) and 3.17(d). Similarly good results are achieved for image 4 using 16×16 subimages, which contains a more well defined texture which presents an easier texture segmentation problem. These results are the most promising so far in terms of having low numbers of false positives whilst retaining fairly accurate boundary detection.



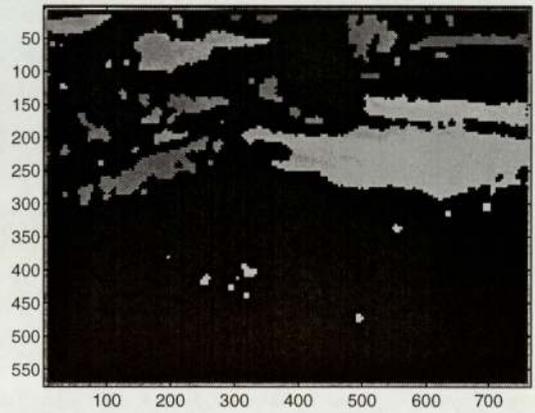
(a) Image 2



(b) Segmentation of (a) using 4×4 subimages

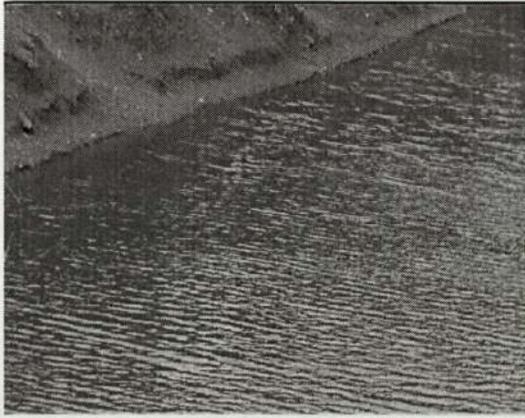


(c) Segmentation of (a) using 8×8 subimages

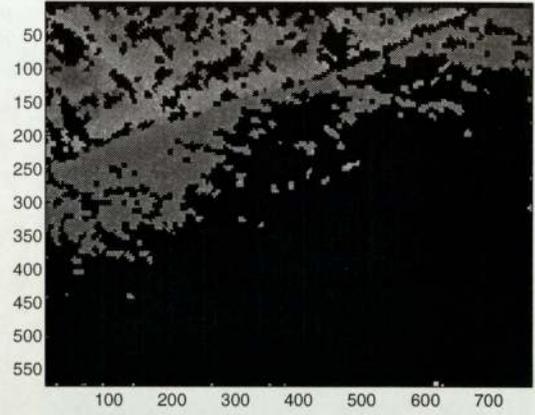


(d) Segmentation of (a) using 16×16 subimages

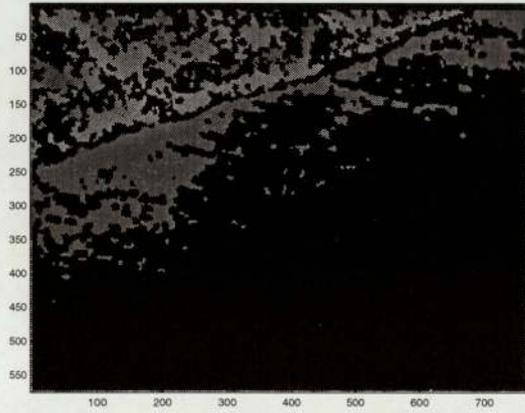
Figure 3.20: Segmentation of image 2 using 1D FFTs



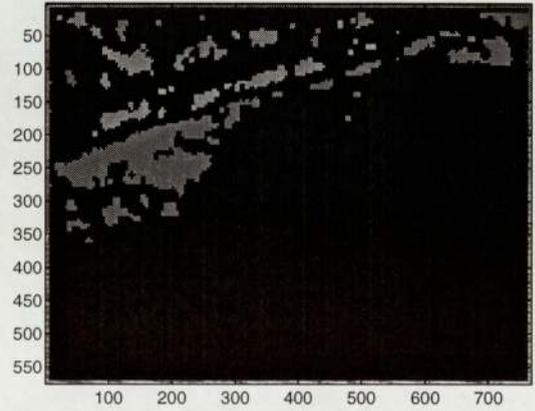
(a) Image 4



(b) Segmentation of (a) using 4×4 subimages



(c) Segmentation of (a) using 8×8 subimages



(d) Segmentation of (a) using 16×16 subimages

Figure 3.21: Segmentation of image 4 using 1D FFTs

3.3.6 Two Dimensional Discrete Fourier Transform

The 1D Fourier transform can be extended to two dimensions. This allows a 2D array of numbers to be exactly represented as a linear combination of a set of two dimensional orthogonal basis vectors, the magnitudes of which are the Fourier coefficients. Consider a $N_1 \times N_2$ array of numbers given by,

$$y(r, s) \quad \begin{array}{l} r = 0, 1, \dots, N_1 - 1 \\ s = 0, 1, \dots, N_2 - 1. \end{array} \quad (3.11)$$

The two-dimensional discrete Fourier transform of $y(r, s)$ is given by,

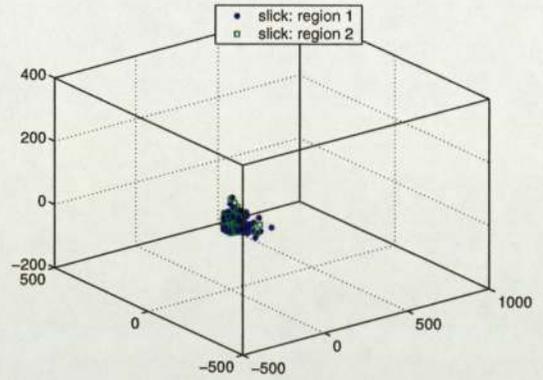
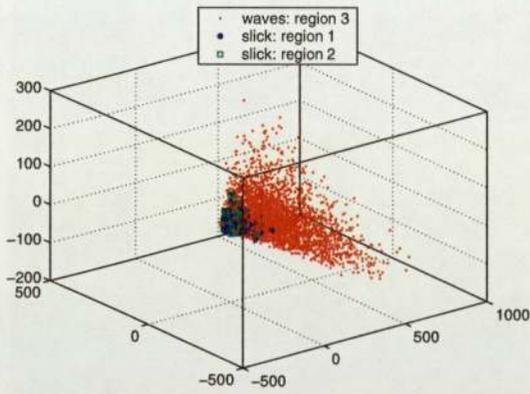
$$Y(k, m) = \frac{1}{N_1 N_2} \sum_{r=0}^{N_1-1} \sum_{s=0}^{N_2-1} y(r, s) e^{-j2\pi(kr/N_1 + ms/N_2)} \quad \begin{array}{l} k = 0, 1, \dots, N_1 - 1 \\ m = 0, 1, \dots, N_2 - 1. \end{array} \quad (3.12)$$

We applied the 2D FFT directly to each $M \times M$ subimage, then formed the result into a vector by concatenating the columns. We took the magnitude of each complex component of this vector in order to obtain the power spectrum. Finally, PCA was then employed to reduce the dimensionality of this vector as described in Section 3.3.4. The number of principal components retained is shown in Table 3.8. The data can be summarised with the same number of principal components as for the 1D FFT method for 4×4 and 8×8 subimages and fewer components for 16×16 subimages. We form the slick and wave texture feature vectors as before by projecting our original M^2 vector onto the principal components.

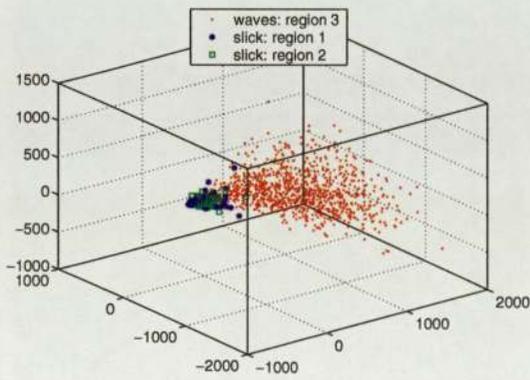
Block Size	Number of principal components retained
4×4	4
8×8	7
16×16	10

Table 3.8: 2D FFT: Number of principal components retained for different block sizes

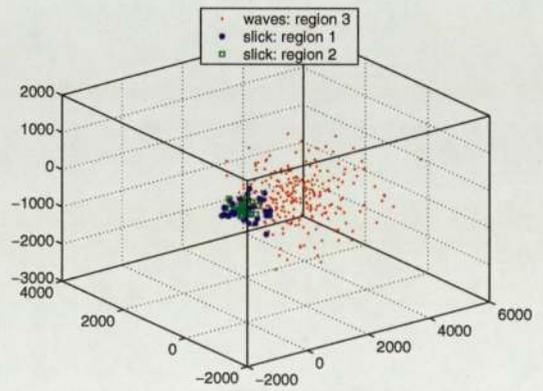
Figure 3.22 shows the slick and wave texture features projected onto their first three principal components.



(a) Using 4×4 subimages



(b) Using 8×8 subimages



(c) Using 16×16 subimages

Figure 3.22: 2D FFT: Slick and wave data projected onto first three principal components

As for the 1D FFT method these 3D plots show we are able to achieve good separation between the slick and wave classes, especially when using 8×8 or 16×16 subimages. A multivariate Gaussian distribution is used to model the slick feature vectors as described in Section 3.3.4.1.

3.3.6.1 Results

We obtain likelihood values for the slick and wave training data based on our conditional density model of ‘normal’ slick texture, these are used to generate ROC curves as described in Section 3.3.4.2.

We present the classification rates for the training data in Table 3.9, these results are based on selecting the likelihood threshold value to be at the ‘knee’ of the ROC curve, see Section 3.3.3.2 for a more detailed description.

Block size	Training data		Test data	
	TP	FP	TP	FP
4×4	0.93	0.12	0.95	0.12
8×8	0.94	0.04	0.98	0.04
16×16	1.0	0	1.0	0

Table 3.9: Classification rates using 2D FFT

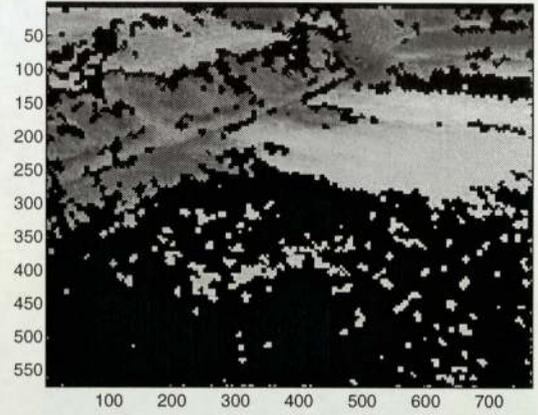
Figure 3.23 and 3.24 show the segmentation results for image 2 and image 3 respectively. These results are based on the threshold chosen for the training image.

3.3.6.2 Analysis

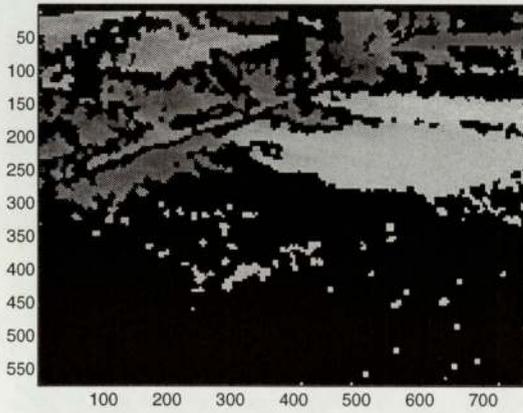
Using 2D FFTs has given an improvement in the segmentation results compared to the 1D FFT method. The 2D FFT produces better results for 8×8 and 16×16 subimages as can be seen in Figures 3.23 and 3.24. The improved correct classification of non-slick textures is accompanied by good boundary detection which also shows an improvement over the results for the 1D FFT method.



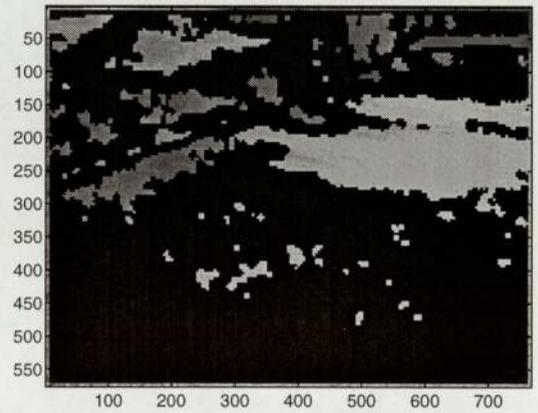
(a) Image 2



(b) Segmentation of (a) using 4×4 subimages



(c) Segmentation of (a) using 8×8 subimages

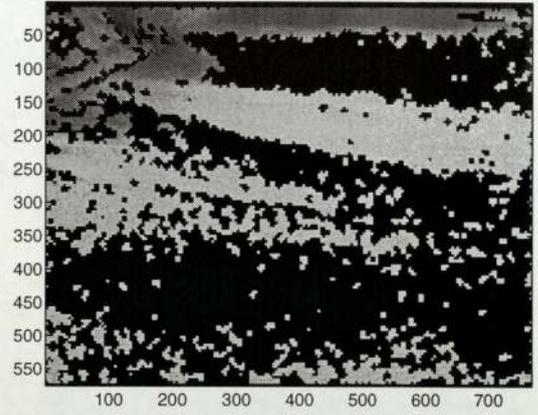


(d) Segmentation of (a) using 16×16 subimages

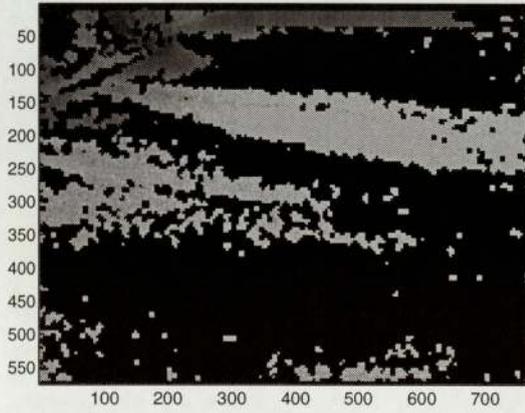
Figure 3.23: Segmentation of image 2 using 2D FFTs



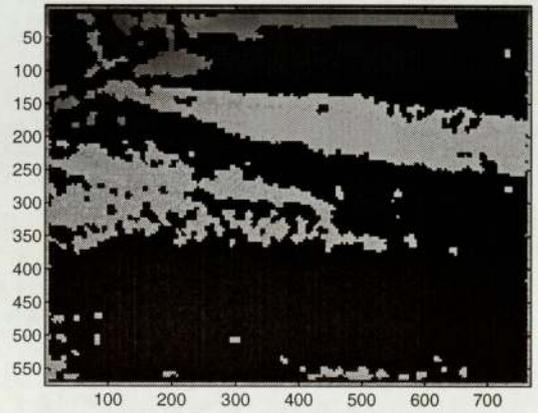
(a) Image 3



(b) Segmentation of (a) using 4×4 subimages



(c) Segmentation of (a) using 8×8 subimages



(d) Segmentation of (a) using 16×16 subimages

Figure 3.24: Segmentation of image 3 using 2D FFTs

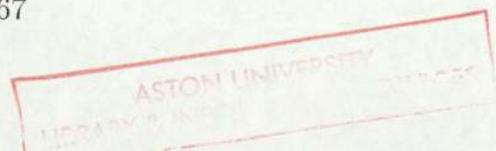
Chapter 4

Comparison of Segmentation

Results

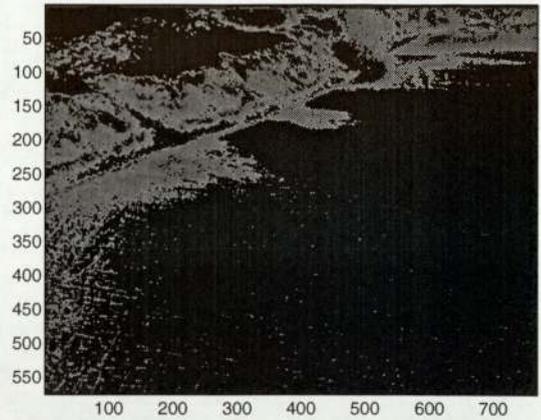
In this section we compare the segmentation results for the test images based on using 8×8 subimages for the range of methods employed in this thesis. Firstly we can see that histogram thresholding is clearly not a suitable approach. The success of this method relies solely on the slick textured regions containing the same range of grey level values from image to image. The intrinsic variability of these natural images makes this approach unfeasible.

The performance of the remaining four feature based methods, namely statistical moments, PCA, 1D FFTs and 2D FFTs is closely matched. This is especially true for those images which contain well defined textures. By well defined, we mean where there is a clear difference between the slick and non-slick textured regions, and the non-slick textured regions have high contrast such as in image 4 and 5. See Figure 4.3(a) and 4.4(a). The slightly superior performance of the FFT methods apparent from the segmentation results of image 2 and 3, see Figure 4.1 and 4.2. The FFT based methods achieved fewer false positives by classifying more of the fine textured, low contrast water regions correctly. Overall, the 2D FFT method produced the best segmentation results.

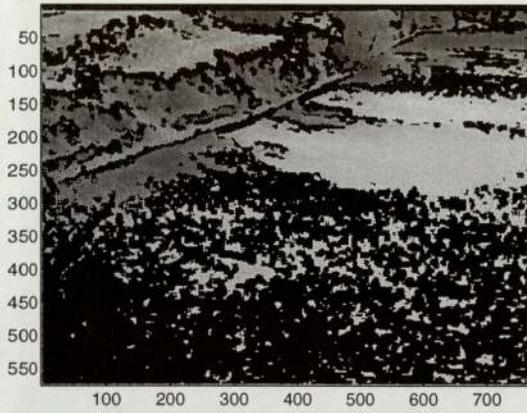




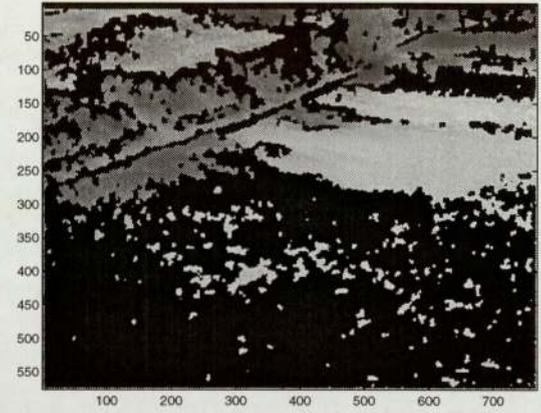
(a) Original image



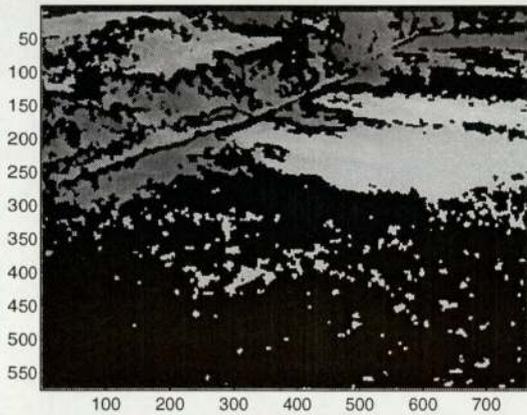
(b) Segmentation of (a) using histogram thresholding



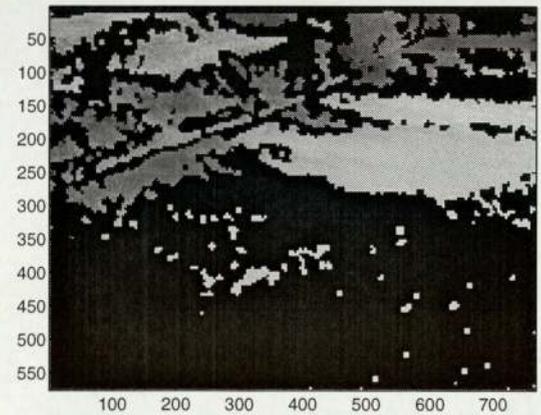
(c) Segmentation of (a) using statistical moments



(d) Segmentation of (a) using PCA



(e) Segmentation of (a) using 1D FFTs

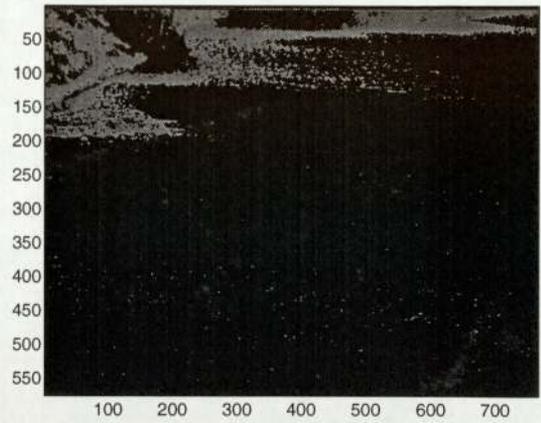


(f) Segmentation of (a) using 2D FFTs

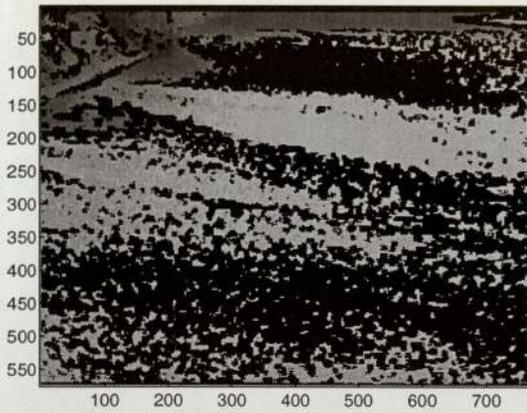
Figure 4.1: Segmentation results for image 2



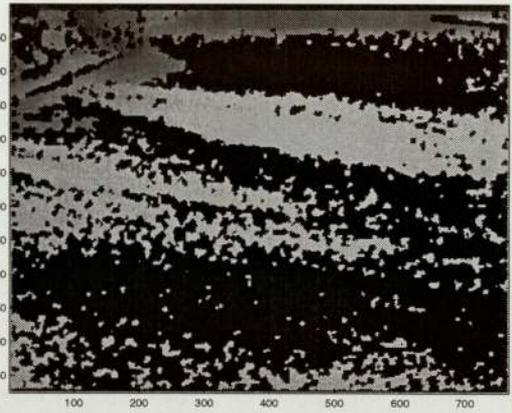
(a) Original image



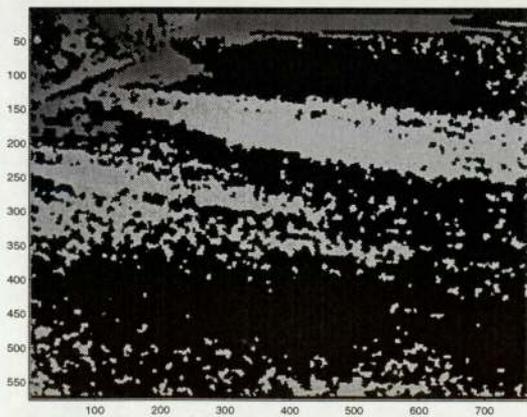
(b) Segmentation of (a) using histogram thresholding



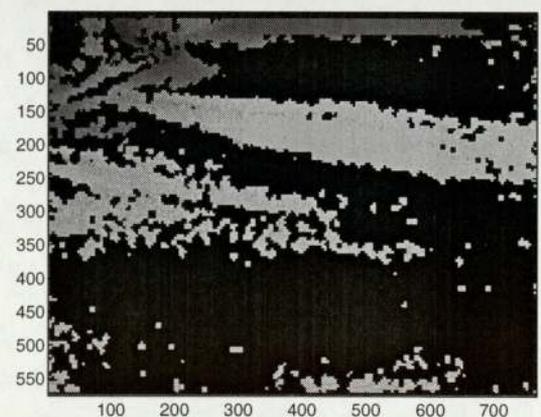
(c) Segmentation of (a) using statistical moments



(d) Segmentation of (a) using PCA



(e) Segmentation of (a) using 1D FFTs

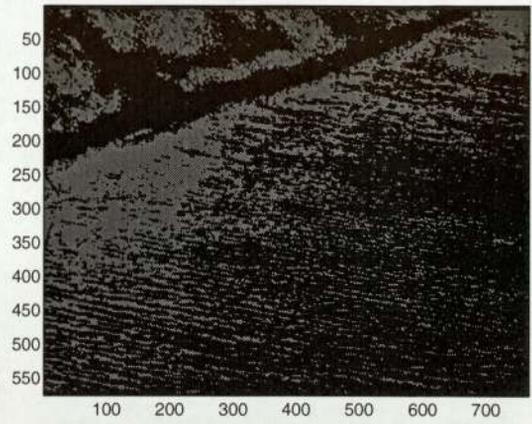


(f) Segmentation of (a) using 2D FFTs

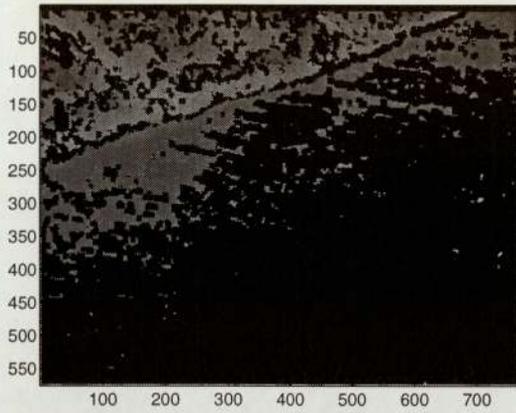
Figure 4.2: Segmentation results for image 3



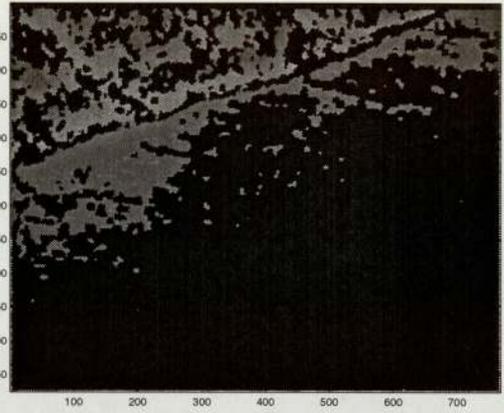
(a) Original image



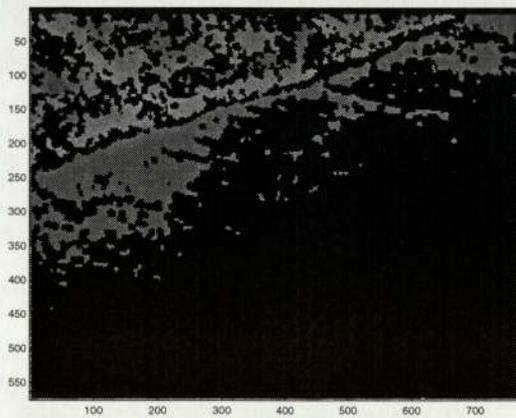
(b) Segmentation of (a) using histogram thresholding



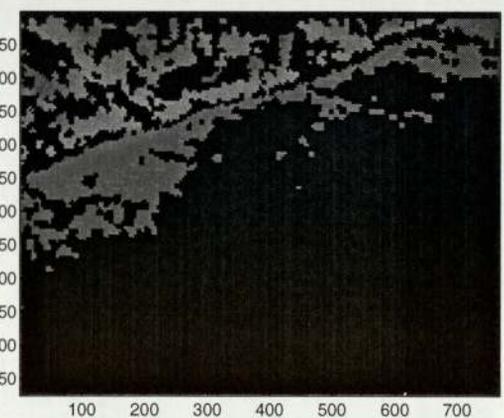
(c) Segmentation of (a) using statistical moments



(d) Segmentation of (a) using PCA

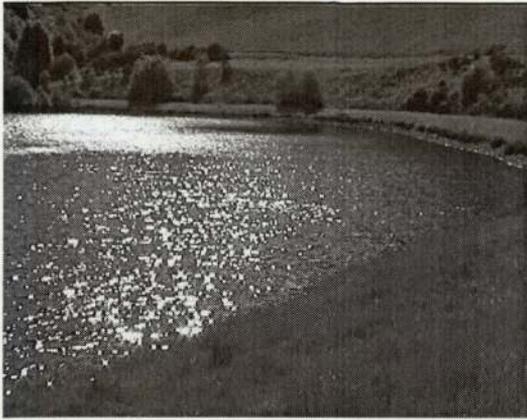


(e) Segmentation of (a) using 1D FFTs

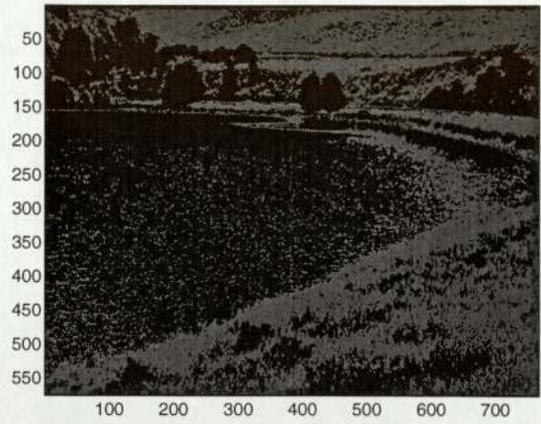


(f) Segmentation of (a) using 2D FFTs

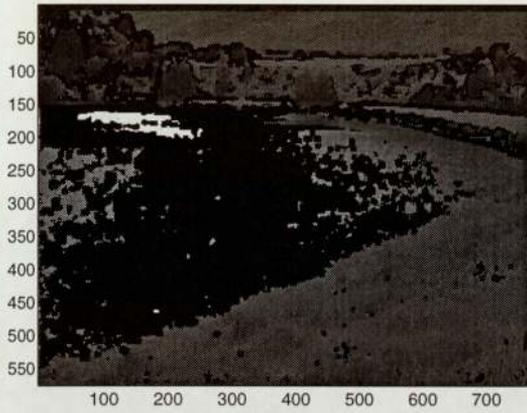
Figure 4.3: Segmentation results for image 4



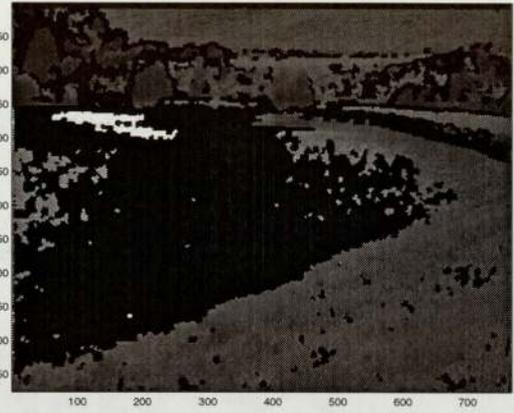
(a) Original image



(b) Segmentation of (a) using histogram thresholding



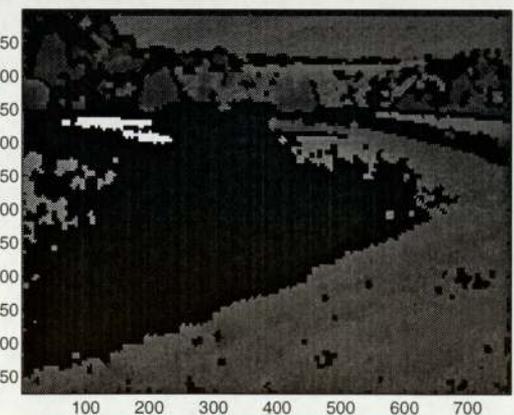
(c) Segmentation of (a) using statistical moments



(d) Segmentation of (a) using PCA



(e) Segmentation of (a) using 1D FFTs



(f) Segmentation of (a) using 2D FFTs

Figure 4.4: Segmentation results for image 5

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The primary aim of this research was to develop a segmentation process within a statistical framework, that would allow pixel by pixel segmentation of an image into slick and non-slick textured regions. We have investigated statistical moments, principal components analysis and one and two dimensional fast Fourier transforms as methods of feature extraction and presented a range of segmentation results for our lake test images. These feature based methods have been compared with histogram thresholding, a standard image processing method most often used for the segmentation of images of man-made environments. The segmentation process was based on a novelty detection approach. We built histogram and multivariate Gaussian density models of slick feature vectors, which represented ‘normality’. ROC curves were used to set the decision boundaries of normality for these models. Previously unseen feature vectors were then classified as normal or novel according to this model, i.e. having slick or non-slick texture.

We investigated how the segmentation performance varied for different threshold points chosen on the ROC curve. The feature vectors were extracted from subimages of 4×4 , 8×8 or 16×16 pixels.

We can draw the following conclusions from this study,

- The subimage size used to extract the feature vectors has a significant effect on segmentation performance. The best results were achieved using 16×16 pixel subimages for the range of images in the test set. This gave better performance than the small subimages as they were not large enough to capture the underlying textural variations of the non-slick textured regions which gave rise to high misclassification rates.
- The threshold chosen from the ROC curve also has a significant effect on segmentation performance. High true positive rates were achieved for the range of methods investigated by selecting the decision threshold for ‘normality’ at the ‘knee’ of the ROC curve. However, this produced a relatively high false positive rate for some of the methods. A significant improvement could be obtained in the segmentation results if a lower threshold point was chosen. The reduced false positive rate gave much better segmentation results overall, although we had to accept a lower true positive rate.
- We demonstrated the failure of the histogram thresholding method for this segmentation problem.
- The 2D FFT method gave the best segmentation results overall.

This research has produced some promising results and demonstrates the feasibility of extracting textural information from grey scale images for use in a segmentation process. The work has also highlighted several areas which could be investigated further to improve the segmentation process, with a view to developing a system which could carry out full slick segmentation. These points are discussed in the next section.

5.2 Future Work

The test images used contain several objects or regions which appear to have slick like texture. These include rocks, trees, and calm water regions such as those identified in

Figure 2.5. A method of identifying these regions as non-slick is required. Features based on intensity levels or shapes may be useful.

The use of dynamic information could be investigated. This could help to differentiate slicks which should be persistent over some reasonable period of time, from shorter duration effects such as the appearance of calm water regions created by local wind effects. Dynamic information could also be used to track the boundaries of a slick.

Bibliography

- [1] C.M Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [2] P Brodatz. *Textures: A Photographic Album For Artists and Designers*. New York: Dover, 1956.
- [3] C Busch. Wavelet based texture segmentation of multi-modal tomographic images. *Computer & Graphics*, 21(3):347–358, 1997.
- [4] C.C Chen and C.L Huang. Markov random fields for texture classification. *Pattern Recognition Letters*, 14:907–914, 1993.
- [5] J.G Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:1148–1161, 1993.
- [6] R.O Duda and P.E Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., 1973.
- [7] O.D Faugeras and W.K Pratt. Decorrelation methods of texture feature extraction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(4):323–332, 1980.
- [8] K Fukunaga. *Statistical Pattern recognition*. Academic Press, 2nd edition, 1990.
- [9] A.K Jain and S Bhattacharjee. Text segmentation using gabor filters for automatic document processing. *Machine Vision and Applications*, 5:169–184, 1992.

BIBLIOGRAPHY

- [10] J.F James. *A Student's Guide to Fourier Transforms with Applications in Physics and Engineering*. Cambridge University Press, 1999.
- [11] S Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2nd edition, 1999.
- [12] A Nairac, N Townsend, R Carr, S King, P Cowley, and L Tarassenko. System for analysis of jet engine vibration data. *Integrated Computer-Aided Engineering*, 6(1):53–65, 1999.
- [13] D.E Newland. *Introduction to Random Vibrations, Spectral and Wavelet Analysis*. Longman Scientific & Technical, 3rd edition, 1993.
- [14] M Petrou and P Bosdogianni. *Image Processing: The Fundamentals*. John Wiley & Sons Ltd, 1999.
- [15] O Pichler, A Teuner, and B Hosticka. A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition*, 29(5):733–742, 1996.
- [16] W.H Press, S.A Teukolsky, W.T Vetterling, and B.P Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.
- [17] R Schalkoff. *Pattern Recognition, Statistical, Structural and Neural Approaches*. John Wiley & Sons, Inc., 1992.
- [18] R.J Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley & Sons, Inc., 1989.
- [19] J.-L Starck, F Murtagh, and A Bijaoui. *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press, 1998.
- [20] J Strand and T Taxt. Local frequency features for texture classification. *Pattern Recognition*, 27(10):1397–1406, 1994.

BIBLIOGRAPHY

- [21] H Tamura, S Mori, and T Yamawaki. Textural features corresponding to visual perception. *IEEE Trans. Systems Man and Cybernetics*, 8:460–473, 1978.
- [22] A.S Tolba and A.N Abu-Rezeq. A self-organising feature map for automated visual inspection of textile products. *Computers in Industry*, 32:319–333, 1997.
- [23] A Webb. *Statistical Pattern Recognition*. Arnold, 1999.
- [24] T.P Weldon, W.E Higgins, and D.F Dunn. Efficient gabor filter design for texture segmentation. *Pattern Recognition*, 29(12):2005–2015, 1996.