Sonar Image Interpretation for Sub-Sea Operations

A thesis submitted for the degree

of

Engineering Doctorate in Optics and Photonics

Oliver James Daniell (ojd5@hw.ac.uk)

School of Engineering and Physical Sciences

Heriot-Watt University, SeeByte Ltd.

Dr. Scott Reed (scott.reed@seebyte.com)

Prof. Yvan Petillot (y.r.petillot@hw.ac.uk)

October 2015

Abstract

Mine Counter-Measure (MCM) missions are conducted to neutralise underwater explosives. Automatic Target Recognition (ATR) assists operators by increasing the speed and accuracy of data review. ATR embedded on vehicles enables adaptive missions which increase the speed of data acquisition. This thesis addresses three challenges; the speed of data processing, robustness of ATR to environmental conditions and the large quantities of data required to train an algorithm.

The main contribution of this thesis is a novel ATR algorithm. The algorithm uses features derived from the projection of 3D boxes to produce a set of 2D templates. The template responses are independent of grazing angle, range and target orientation. Integer skewed integral images, are derived to accelerate the calculation of the template responses. The algorithm is compared to the Haar cascade algorithm. For a single model of sonar and cylindrical targets the algorithm reduces the Probability of False Alarm (PFA) by 80% at a Probability of Detection (PD) of 85%. The algorithm is trained on target data from another model of sonar. The PD is only 6% lower even though no representative target data was used for training.

The second major contribution is an adaptive ATR algorithm that uses local sea-floor characteristics to address the problem of ATR robustness with respect to the local environment. A dual-tree wavelet decomposition of the sea-floor and an Markov Random Field (MRF) based graph-cut algorithm is used to segment the terrain. A Neural Network (NN) is then trained to filter ATR results based on the local sea-floor context. It is shown, for the Haar Cascade algorithm, that the PFA can be reduced by 70% at a PD of 85%.

Speed of data processing is addressed using novel pre-processing techniques. The standard three class MRF, for sonar image segmentation, is formulated using graph-cuts. Consequently, a 1.2 million pixel image is segmented in 1.2 seconds. Additionally, local estimation of class models is introduced to remove range dependent segmentation quality. Finally, an A* graph search is developed to remove the surface return, a line of saturated pixels often detected as false alarms by ATR. The A* search identifies the surface return in 199 of 220 images tested with a runtime of 2.1 seconds. The algorithm is robust to the presence of ripples and rocks.

Acknowledgements

I would like to acknowledge the guidance, direction and proof reading provided by Yvan Petillot and Scott Reed. The work of Nicolas Valeyrie, Yan Pailhas and Jamil Sawas in compiling a database of sidescan sonar ATR papers provided a very useful starting point point for this work. In addition I would like to acknowledge the informative and stimulating discussions with all of the above named and additionally, colleagues, too numerous to name, at SeeByte Ltd.

Many friends also contributed to the proof reading on this document, I am particularly grateful for James Rowell's work in proof reading the technical chapters and David Walker's work in proof reading everything else.

I would also like to thank DSTL and NATO CMRE for providing the MUSCLE SAS data and SeeByte Ltd. for providing the Sidescan Sonar data.

Finally I would like to thank my wife, Ruth Butler, for putting up with the many weekends invested in writing this thesis.

ACADEMIC REGISTRY
Research Thesis Submission

Name:	Oliver Daniell		
School/PGI:	School of Engineering and Physical Sciences / Industrial Doctorate Centre in Optics and Photonics		
Version: *(i.e. First, Resubmission, Final)*	First	Degree Sought (Award **and** Subject area)	Engineering Doctorate in Optics and Photonics

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	

Submission

Submitted By *(name in capitals):*	OLIVER DANIELL
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should bound into the submitted thesis.

Updated February 2008, November 2008, February 2009, January 2011

Contents

List of Figures

Acronyms

ATR Automatic Target Recognition

AUV Autonomous Underwater Vehicle

CADAC Computer Aided Detection and Classification

CMRE Centre for Maritime Research and Experimentation

CRF Conditional Random Field

$\mathbb{C}$**WT** Complex Wavelet Transform

DL Down Left

DoG Difference of Gaussians

DR Down Right

DSTL Defence Science and Technology Laboratory

DT$\mathbb{C}$**WT** Dual Tree Complex Wavelet Transform

DTW Dual Tree Wavelet

DWT Discrete Wavelet Transform

EOC Environmental and Operating Condition

FLS Forward Looking Sonar

GLCM Gray Level Co-occurrence Matrix

GLOH Gradient Location and Orientation Histogram

HOS Higher Order Statistics

ICE Iterative Conditional Estimation

ICM Iterated Conditional Modes

LoG Laplacian of Gaussians

MAP Maximum a Posteriori

MBES Multi-Beam Echo Sounder

MCM Mine Counter-Measure

MCMC Markov Chain Monte Carlo

MLO Mine Like Object

MLP Multi-Layer Perceptron

MLT Mine Like Texture

MMP Maximum Marginal Probability

MRF Markov Random Field

MSD Mean - Standard Deviation

MSER Maximally Stable Extremal Regions

NN Neural Network

PCA Principle Component Analysis

PCNN Pulse Coupled Neural Networks

PD Probability of Detection

PFA Probability of False Alarm

RI Re-acquire Inspect

RIFT Rotation Invariant Feature Transform

ROC Receiver Operator Characteristics

ROI Region of Interest

SAS Synthetic Aperture Sonar

SCM Search-Classify-Map

SFBF Serpentine Forward-Backward Filter

SIFT Scale Invariant Feature Transform

SLAM Simultaneous Localisation and Mapping

SNR Signal to Noise Ratio

SONAR SOund Navigation And Ranging

SSS Sidescan Sonar

SURF Speeded Up Robust Features

SVM Support Vector Machine

TVG Time Variable Gain

UL Up Left

UR Up Right

USAN Univalue Segment Assimilating Nucleus

Chapter 1

Introduction

1.1 Introduction

In this thesis we research the use of Automatic Target Recognition (ATR) algorithms in Mine Counter-Measure (MCM) missions. MCM missions are conducted by navies across the world to neutralise the threat caused by underwater explosive devices. There is a requirement for these missions to be safe, fast and accurate. To this end, human interaction is minimised and data-collection is performed by tethered vehicles or Autonomous Underwater Vehicles (AUVs). These survey vehicles mount Sidescan Sonar (SSS) sensors that rapidly image large areas of the sea-floor. However, with modern, high-resolution, sensors human operators can suffer from data overload. As such, review of the data is time-consuming and prone to human error.

ATR algorithms can assist human operators in reviewing MCM missions. These algorithms identify Mine Like Objects (MLOs) in the sonar images and flag them for review by an operator. In this scenario the ATR algorithm is used to increase the tempo (or speed) of a mission and increase the probability that a target will be detected. Using the ATR algorithms to assist operators limits the damage that can be caused by an algorithm failing to detect a target.

ATR algorithms are also used in autonomous MCM systems. SeeByte Ltd. [2] are currently actively developing the NEPTUNE [3] system for multi-vehicle co-operative MCM missions. In this scenario a mine identified by a survey vehicle is then autonomously inspected by a close-inspection vehicle. This brings significant advantages over the traditional, sequential, MCM mission. Traditionally, sonar data is collected in a Search-Classify-Map (SCM) phase. The vehicles are then recovered, an operator reviews the data and close inspection vehicles are launched to conduct the Re-acquire Inspect (RI) phase of the mission. However, with the NEPTUNE system these phases can be performed in parallel. Upon recovery of the vehicles both SSS and video data of targets identified by the ATR is available for an operator to review. However, this scenario raises additional

challenges with respect to the robustness of the ATR algorithm and the context-sensitive interpretation of the data.

In the next section we introduce the context of MCM missions in more detail. This is followed by the objectives for this thesis. Finally, we give an overview of the content of each chapter.

1.2 Context

MCM missions are conducted to detect and eliminate underwater explosive devices. However, the way in which the mission is conducted is dependent on the operational objectives. All of the operations share a number of common phases. The first phase is Inspection. In this phase one or more underwater vehicles conduct a large area survey of a region of sea-floor. Typically this is conducted using SSS sensors. These sensors produce a map of the sea-floor which, in some ways, is similar to that produced by aerial photography. The next phase is Identification; in this phase one or more human operators review the data to identify MLOs. The targets are then Reacquired and Neutralised. In this thesis we focus on the use of ATR algorithms in the Inspection and Identification phases [4].

We focus on three main operational objectives. The first objective is to inspect and neutralise a percentage of the targets in a pre-defined area. To accomplish this objective all the data must be reviewed in a systematic manner. ATR algorithms can assist by improving the Probability of Detection (PD). However, as an operator must still review all the data there is little scope for saving time.

The second common objective is to clear a path through an area that is known to contain mines. The aim is to identify the path which is easiest to clear. Here ATR algorithms can help by rapidly identifying the regions with the lowest density of MLOs. This information can be used to optimise the data set that is sent to an operator for review.

The final objective that we consider is identifying if an area contains mines. In this objective, the most mine-like targets are identified by the ATR algorithm. If any of those targets are mines then a full survey is performed. The ATR algorithm can assist by generating a list of the most likely MLOs. If a mine is found, there is no requirement for an operator to review the rest of the data. In the NEPTUNE system this review can be performed on the vehicle and MLOs sent to the operator via an acoustic modem. This synergy between algorithms and vehicles can offer significant advantages that are not possible with the current generation of MCM systems.

In the short to medium term, operator aided MCM missions are likely to be the standard mode of operation. Therefore, commercially, there are several pressing issues. The first is that of gaining operator trust in autonomous algorithms in regions where the algorithm is known to perform well. The second is reducing the costs associated with data collection for training the algorithms. The third is providing accurate performance es-

timation and allowing the ATR to be rapidly retrained for a new environment. These requirements inform the objectives stated in the next section.

1.3 Statement of Objectives

In this section we describe the thesis objectives and the motivation for each objective. There are three main objectives that we address. The first objective is to improve the speed of ATR and associated algorithms, such as pre-processing and sea-floor analysis. The second objective is to increase the robustness of ATR algorithms to environmental conditions. The final objective is to reduce the large quantities of training data that are required for the current generation of supervised sonar ATR algorithms [5]. We now examine each of the objectives in more detail.

Improving the speed of ATR algorithms: The speed, or computational efficiency, of ATR algorithms is important for post-mission analysis, and embedded processing of sonar data on a vehicle. For post-mission analysis the operator must wait for the ATR algorithm to process the data before they can review the results. For time-critical scenarios it is important that the benefits provided by ATR algorithms are not out-weighed by the time costs. For embedded processing the ATR and associated algorithms are one of many processes that share the computational resources of the vehicle. If the computational load is too high then the algorithms will lag behind the data acquisition. This can lead to the vehicle waiting for algorithms to complete before further decisions can be made.

Increase robustness to environmental conditions: Modern sonar ATR algorithms produce excellent results in flat and moderately complex terrain [6]. However, ripples and rocky regions can create large numbers of false alarms. We aim to address the problem in two ways. The first objective is to reduce or remove sources of noise from the image. The second is to analyse the sea-floor to classify rippled and rocky areas in which the ATR will produce large numbers of false alarms.

Reduce the quantity of training data: ATR algorithms currently in use at SeeByte require over 1000 images of a target to prevent over-training [5]. This target data exists for the current generation of SSS devices, however the cost of collecting data for new models sonar is prohibitive. While a human operator can be trained on the legacy data, this is not possible for many supervised algorithms. We aim to create a sonar ATR algorithm that can be partially, or fully, trained using existing target databases.

This thesis was produced as part of an Engineering Doctorate programme. In the next section we describe how the novel contributions from this thesis meet the objectives and explain their commercial relevance.

1.4 Novel Contributions and Commercial Relevance

In this section we describe the novel contributions from this thesis and the commercial relevance. The first novel contribution is the use of an A* graph search algorithm to detect surface returns in SSS images. The surface return is a line of saturated pixels caused by the sonar reflecting off the surface of the water. This artefact is common in shallow environments (less than 30m) and is shown in figure 2.7. The surface return is a common cause of false alarms from SSS ATR algorithms. Using the A* search algorithm we demonstrate robust detection of surface returns in 199 of the 220 images tested. The runtime for all 220 images is just 2.1 seconds. The A* search algorithm has been integrated into SeeByte's common software library and is used as a pre-processing stage for all ATR algorithms. By removing a common source of false alarms the robustness of the ATR algorithms to environmental conditions is increased.

The next novel contribution is the use of the graph-cuts algorithm for solving the three class Markov Random Field (MRF) segmentation problem in sonar images. The image is segmented into bright object regions, dark shadow regions and background regions. The three class segmentation problem is posed as two binary segmentations. Object and shadow regions are segmented from background and then object regions are segmented from shadow regions. Graph-cut based segmentation of a binary MRF is guaranteed to converge to the optimal solution and is faster than other approaches, such as simulated annealing, which also achieve optimal solutions. Additionally, we introduce a new approach to estimating local MRF class models. This overcomes the common problem of sonar segmentation algorithms failing to segment objects at far range, where there is a reduction in image contrast. The segmentation algorithm is integrated with SeeByte's common software library and has been used as part of a Defence Science and Technology Laboratory (DSTL) research contract. The segmentation algorithm is also used to estimate the height of objects in sonar images. This information is used to identify rocky areas of sea-floor, in which the ATR algorithm is likely to produce a large number of false alarms. This information is used by SeeByte's NEPTUNE system to exclude these regions from automatic processing, increasing the robustness of the system to environmental conditions.

The objective of reducing the quantity of data required by ATR algorithms is achieved using a novel supervised ATR algorithm. This algorithm is the main contribution of the thesis. The algorithm models the sonar image formation process to reduce the dependence on the characteristics of the sonar data for which the algorithm was trained. 2D templates are generated locally from a large set of 3D boxes. The calculation of the 2D template responses is accelerated using a novel algorithm called integer skewed integral images. In this algorithm the standard integral image approach is adapted such that the sum of pixels within skewed rectangular regions can be rapidly calculated. It is shown that the

3D feature based ATR algorithm can be trained entirely on target data from another model of sonar and still achieve a maximum PD within 6% of the results where the algorithm is trained on the same type of sonar as it is tested. This algorithm forms the subject of a patent (1) and a conference paper (2).

Increased robustness to environmental conditions is achieved by filtering the ATR targets based on the local sea-floor type. This requires the calculation of textural characteristics for the sea-floor and the segmentation of the sea-floor into rocky, rippled and flat regions. The approach introduces a number of novel contributions. The main contribution is the use of dual tree wavelets for the calculation of textural characteristics. An implementation of the dual tree wavelets has been written in c++ and integrated with SeeByte's common libraries. The segmentation algorithm is used in SeeByte's NEPTUNE system to perform adaptive behaviours based on the local sea-floor type. The segmentation is integrated into SeeByte's sonar mission review tool and the US Navy's mission review tool. This work allows operators to quickly asses how difficult mine hunting will be in the mission with respect to the sea-floor conditions. The wavelet features were presented in a conference paper (3). Our work on sea-floor features has been used by CMRE to train ATR algorithms that are aware of the local sea-floor conditions [7].

Filtering of ATR contacts is achieved using a Neural Network (NN). The work is the first example of using sea-floor information to filter the results from an ATR algorithm. In this approach the NN learns the sea-floor types that are likely to cause a large number of false alarms. This allows targets with a low confidence value to be retained in simple areas and removed in rocky or rippled areas where they are more likely to be false alarms. The relationship between ATR performance and sea-floor characteristics was presented in a conference paper (4). The NN based filtering was presented in (3) and also forms part of DSTL research contract performed by SeeByte.

(1) O. Daniell, "Sonar imaging (20140079287)," 2014. (US Patent App. 14/030,749)

(2) O. Daniell, Y. Petillot, and S. Reed, "Sonar independent ATR," in *Oceans Conf.*, San Diego, California, 2013.

(3) O. Daniell, Y. Petillot, S. Reed, and J. Vazquez, "Reducing false alarms in automated target recognition using local sea-floor characteristics," in *Sensor Signal Processing for Defence (SSPD)*. Edinburgh: IEEE, 2014, pp. 1–5.

(4) O. Daniell, Y. Petillot, and S. Reed, "Unsupervised sea-floor classification for automatic target recognition," *Proc. International Conf. Remote Sens.*, no. October, 2012.

1.5 Chapter Summary

In this section we give a brief summary of the contents of each chapter. In chapter 2 the principles of operation of sonar are introduced. In chapter 2 we address the robustness of ATR algorithms by correcting sonar imagery to remove sources of noise. In chapter 3 we review existing ATR algorithms with respect to the challenges of robustness, speed and training data requirements. In chapter 4.6 we introduce a new ATR algorithm that reduces the quantity of training data required by over 50% compare to similar algorithms such as the Haar cascade[5]. Finally, in chapter 5 we address the problem of ATR performance with respect to the local environment. We introduce tools to characterise the sea-floor and use this information to filter ATR results based on the local context. These chapters are now summarised in more detail.

In chapter 2 we present an overview of image formation using SOund Navigation And Ranging (SONAR). We first introduce the operating principles of active sonar. Next, we describe the propagation of sound in water and the sources of noise which can degrade the signal. We describe how beam forming and steering can be used to focus a sonar beam on a specific region of the sea-floor. Finally, we explain the conversion of the acoustic signal into an image which can be interpreted by a human operator or machine-learning algorithm.

Next, in chapter 2 we introduce techniques for pre-processing SSS and Synthetic Aperture Sonar (SAS) images. Imaging artefacts are problematic for ATR algorithms and sea-floor classification. For example, in chapter 4.6 on ATR, we show that the bright surface return is often identified as a target. Meanwhile, in chapter 5 on sea-floor characterisation, we show that the beam pattern is mistakenly classified as ripples. We implement and review existing algorithms for image pre-processing and discuss their application to various image processing problems. Next, we show that there are a number of artefacts that cannot be removed consistently with existing algorithms. We implement a novel algorithm for removing the surface return from sonar images using a graph search algorithm. Finally, we introduce a novel approach to image segmentation in SSS images using graph-cuts.

In chapter 3 we review the existing literature on ATR in SSS and SAS images. Historically, the majority of the research on sonar ATR algorithms has been focussed on improving the PD and Probability of False Alarm (PFA). As a result, ATR in simple environments can be considered a solved problem. However there are a large number of additional challenges that arise from the deployment of ATR algorithms in practical systems. These include the long term practicality of algorithm. For example, it is common for an algorithm to work well in the environment in which it was trained. However, in other environments the algorithm can behave unpredictably. In this chapter we present an overview of ATR in sonar images and applicable techniques from video and image

processing.

In chapter 4 we introduce the Haar-Cascade ATR algorithm and show how it performs when it is trained and tested on data from different sonar. In the first half of the chapter we present an overview of the Haar cascade algorithm, the Haar wavelet features and the feature selection. We then present empirical results that demonstrate that the performance of the algorithm is sensitive with both respect to the quantity of training data and the model of sonar. Finally we discuss the reasons for the sensitivity of the Haar cascade to these variables.

In chapter 4.6 we develop a novel algorithm to address the short-comings of the Haar-Cascade. We address the problems by locally generating 2D features from 3D objects. However, unlike template matching, we remove the need for a prior object model by defining a large number of boxes with different dimensions. These are referred to as generating boxes. Unknown objects can be described by a combination of many generating boxes. The specific combination is learnt by the algorithm during training. Each feature is described uniquely by the size of the generating box. Therefore, once we have trained a classifier on one set of Environmental and Operating Conditions (EOCs) we can adapt to a new set of EOCs simply by defining a new projection model for the generating boxes. We introduce the algorithm and then compare its performance to the Haar-Cascade.

In section 5 we introduce a new approach to sea-floor characterisation. As we discussed earlier, this information is useful both for for autonomous MCM missions and operator review of SSS data. The aim of the work in this chapter is to find a robust, computationally efficient approach to characterise and segment rippled and complex regions of the sea-floor. We first discuss the calculation of textural characteristics from sonar images. Three textural characteristics are introduced and the distribution of these features with respect to sea-floor types and sonar types is explored. Next, we develop a segmentation algorithm to group homogeneous regions of the sea-floor. Some regions are not well described by the textural characteristics. For example a flat region may contain a large number of MLOs. Therefore, we introduce a simple approach to estimating the clutter density.

In chapter 6 we introduce a novel approach to filter the detections from an ATR algorithm using the sea-floor information. In this chapter, we assume that there exists a trained ATR system, and that we have no access to the underlying algorithm. We first demonstrate that the sea-floor characteristics can be used to discriminate between regions where the ATR will perform well and those where there will be a large number of false alarms. Next, we introduce a novel approach using a neural network to weight the ATR confidence value according to the local sea-floor characteristics. The results are compared to the ATR algorithm before the filter is applied.

In the next chapter we start by introducing the operating principles for active sonar systems such as SSS and SAS.

Chapter 2

Theory of Imaging Sonar and Image Enhancement

2.1 Introduction

In this chapter we present an overview of image formation using SOund Navigation And Ranging (SONAR) and introduce methods to process the imagery for visualisation. In under-water environments the range of traditional optical technologies is limited by absorption. Absorption in the acoustic band is significantly lower than that for electromagnetic signals. This enables low frequency acoustic waves to propagate over many kilometres through the ocean. Even at the higher frequencies, used by high resolution imaging sonar, the operational range is between 30 m to 150 m.

Sonar devices operate on the principle of converting acoustic energy into electrical signals. Active sonar insonify targets with arrays of transducers (or projectors). These arrays allow sonar such as Sidescan Sonar (SSS) [8], Forward Looking Sonar (FLS) [9] and Multi-Beam Echo Sounder (MBES) [10] to control the direction of the sonar beam in a process known as beam steering. Fine control over the direction of the sonar beam enables a "photographic" picture of the environment to be produced. Hence these systems are referred to as "imaging sonar".

Imaging sonar are employed in a diverse range of civilian and military applications. The long range of SSS and Synthetic Aperture Sonar (SAS) systems is useful for conducting, high resolution, large area scans of the sea-floor. Civilian applications of SSS include large area marine habitat surveys. For example analysing the trawling impact on the sea-floor [11], and the tracking and inspection of oil pipelines. Military applications include the detection and analysis of mines in Mine Counter-Measure (MCM) missions. The high update rate and shorter range of FLS [9] is typically employed in obstacle avoidance [12], feature based navigation for Autonomous Underwater Vehicle (AUV) operations [13] and in object tracking systems for mine neutralisation and ship hull inspection [2].

In section 2.2 we introduce the operating principles of active sonar. We describe the propagation of sound in water and the sources of noise which can degrade the signal. In section 2.3 we describe the how beam forming and steering can be used to focus a sonar beam on a specific region of the sea-floor. In section 2.4 we explain the conversion of the acoustic signal into an image which can be interpreted by a human operator or machine learning algorithm.

In the second half of this chapter we introduce techniques for pre-processing SSS and SAS images. Sonar images are corrupted by artefacts, such as the surface return and beam pattern. These artefacts are problematic for autonomous algorithms. We implement and review existing algorithms for image pre-processing and discuss their application to various image processing problems. We show that there are a number of artefacts that cannot be removed consistently with existing algorithms. In section 2.6.1 we introduce algorithms to remove the intensity variations caused by range and beam pattern. We implement and compare several existing algorithms and discuss their application to MCM missions and sea-floor characterisation. In section 2.6.2 we discuss the removal of the surface return from the image. We implement an existing algorithm for detecting the surface return and show how it can break down in images with complex sea-floor regions. We then introduce a novel approach using the A* search algorithm.

In section 2.7.1 we give a brief introduction to Markov Random Fields (MRFs) and graph cuts. We then introduce a simple approximation to the noise models found in SSS images. The noise models are used to calculate the likelihood of a pixel belonging to a given class. From the class models we derive a set of cost functions that assign an energy cost to a pixel with a given intensity belonging to a specific SSS class. Next, we derive a set of binary segmentation problems from the three class problem and introduce an iterative approach to estimating the class models for a particular image. We, then present a novel approach to estimating local parameters for the class models.

2.2 Operating Principles

2.2.1 Basics of Sonar

The simplest active sonar system consists of a sound projector (or transmitter) and a receiver. Assuming a simple ray tracing model the acoustic pulse will propagate in a spherically symmetric shell through the medium at speed c until it encounters a surface. A surface which varies on a scale smaller than the wavelength of the sound wave will scatter the incident pulse isotropically. A fraction of the energy is reflected towards the sonar and detected by the receiver. The range r to the nearest object is determined by measuring the

time difference t between the outgoing pulse and the first return:

$$r = \frac{t}{2c} \tag{2.1}$$

where the factor of 2 in the denominator comes from the sound wave travelling along the outgoing and return path.

With a single transmitter and receiver, there is no way to distinguish between the return from directly below the vehicle and that from an object positioned in front of the vehicle. A practical sonar system therefore consists of arrays of projectors and or receivers which can focus the beam. These arrays are discussed in section 2.3. First we discuss the propagation of sound through the water and various noise sources that effect the processing of the sonar signal.

2.2.2 Propagation of Sound

The formation of sonar images is complicated by the propagation of sound through the water. A sound wave propagates through water as a series of compressions and rarefactions. This transfers the sound energy to the surrounding medium. In the far field the propagation can be described according to the Hodgson's ray tracing model [14], in the near field the propagation must be calculated according to the Huygens-Fresnel principle. However, any theoretical model is complicated by the properties of the medium. Sonar systems must account for losses due to absorption and spherical spreading, refraction due to changes in temperature, salinity, pressure and diffraction. The speed of sound in water is dependent on the temperature and salinity and can be calculated empirically from Mackenzie's equation [14]. However, these factors are dependent on stochastic properties of the medium and hence represent a fundamental limit to the information which can be extracted from the sonar return.

Energy losses are typically corrected by dividing the sonar return by a function that approximates the energy loss due to spherical spreading and absorption. This approach is explained in more detail in section 2.6.1. However, this correction amplifies both the noise and the signal in the far range. Therefore, while the return is improved visually there is no increase in the entropy of the signal. Additionally, if the approximation differs from the true loss function then range dependent intensity variations can be introduces to the sonar signal. Later in this chapter we investigate a number of post-processing algorithms to normalise these variations. As well as compensating for sonar loss there are also multiple noise sources to contend with.

2.2.3 Noise and Reverberation

The ocean is inherently noisy, signals are received not only from the target but from other vehicles and the environment. Signal processing algorithms must also cope with reverberation noise, where the random scattering from objects on the seabed combines to form areas of constructive and destructive interference. The dominant source of noise in a sonar is dependent on the frequency and bandwidth of the sonar. At low frequencies, less than 500 Hz, the noise is due to shipping. At frequencies above 64 kHz the thermal noise of the ocean is the dominant contribution. Between these frequencies the noise will also have environmental contributions such as rain and waves. This noise can be modelled as an additive Gaussian contribution to the sonar return, however compared to the reverberation noise it is negligible [14].

Reverberation noise is a consequence of the stochastic nature of the seabed [15]. The seabed can be approximated as a large number N_d of independent scattering surfaces each of amplitude a_i and phase ψ_i . In SSS and FLS systems the sonar return over the range $r, r + \Delta r$ is the complex sum of the returns from these scattering surfaces. The response ρ of a pixel volume is

$$\rho = \sum_{i=1}^{N_d} a_i exp(j\psi_i) = Aexp(j\phi) = X + jY \tag{2.2}$$

where as a result of the central limit theorem, for N_d large, the real X and imaginary Y coefficients will tend towards a Gaussian distribution. The amplitude $A = \sqrt{X^2 + Y^2}$ is described by a Rayleigh distribution with standard deviation σ

$$f(A;\sigma) = \frac{A}{\sigma^2} e^{-A^2/2\sigma^2}, A \geq 0 \tag{2.3}$$

The Rayleigh distribution has the property that the mean μ and standard deviation σ are related by a single multiplicative constant

$$\mu = \sigma\sqrt{\frac{\pi}{2}} \tag{2.4}$$

This leads to the multiplicative, or speckle, noise observed in sonar images in which the magnitude of the noise is proportional to the mean sonar return.

In practice, for high resolution SSS and SAS, the assumption that the number of scatterers in a resolution cell is large is not appropriate and the statistics are best described by a Weibull distribution or, more accurately, a K-distribution [16]. The Weibull distribution models the limiting case of an exponential distribution at large grazing angles where there is no shadowing. This differs from the Gaussian scattering at low grazing angles, which is formed from a mixture of object returns and shadow, and results in the Rayleigh model de-

scribed. The k-distribution compounds a model of the sonar cross-section and the speckle characteristics. SAS images, in particular, suffer from increased speckle noise as each resolution element is composed of multiple beam returns, each individually corrupted by speckle noise.

2.3 Beam Forming and Steering

Controlling the shape (beam forming) and direction (beam steering) of the sonar beam is essential to form a picture of the environment. Beam forming and steering can be accomplished equally with an array of projectors or hydrophones. However, the former is preferred as an array of projectors can emit more energy without cavitation (cavitation is the formation of bubbles in the water and is caused by the rapid change in pressure due to the sound waves). A single projector emits sound energy in a spherically symmetric shell (figure 2.1) which can lead to ambiguous detections.

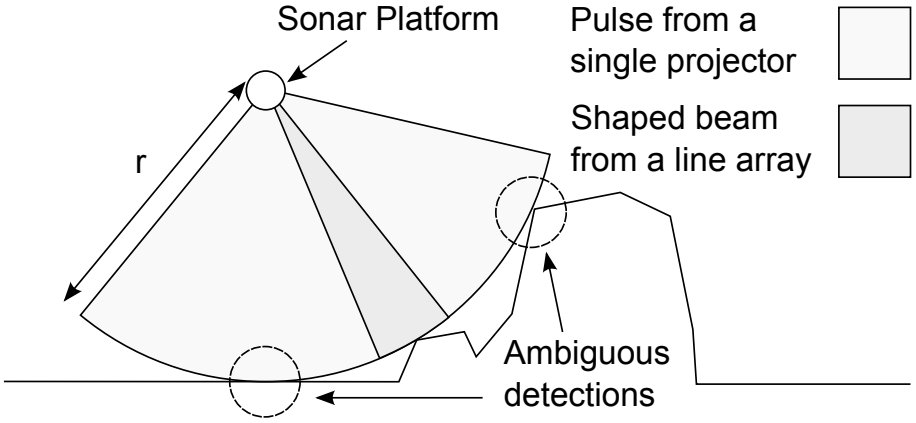


Figure 2.1: Comparison of the sonar beam from a projector and a line array

Arrays of projectors and receivers are used to shape and steer a sonar beam by exploiting the interference of the individual sound waves. Consider a line array of projectors emitting sound waves at a wavelength λ, and a single receiver in the far field at an angle θ to the axis perpendicular to the array. The signal received by the receiver will be the quadrature sum of the individual sound waves emitted by each of the transducers. By varying the number n of, and the spacing d between, the projectors the shape of the beam can be modified such that the energy is concentrated in the main lobe which lies along the $\theta = 0^\circ$ axis. A steer angle θ_s can be introduced by electronically varying the phase between each element of the array, displacing the main lobe about the $\theta = 0^\circ$ axis by an angle θ_s. The equation determining the shape of the beam $b(\theta)$ for a line array is

$$b(\theta) = 20\log\left[\frac{sinn\pi(dsin\theta/\lambda - dsin\theta_s/\lambda)}{nsin\pi(dsin\theta/\lambda - dsin\theta_s/\lambda)}\right] \tag{2.5}$$

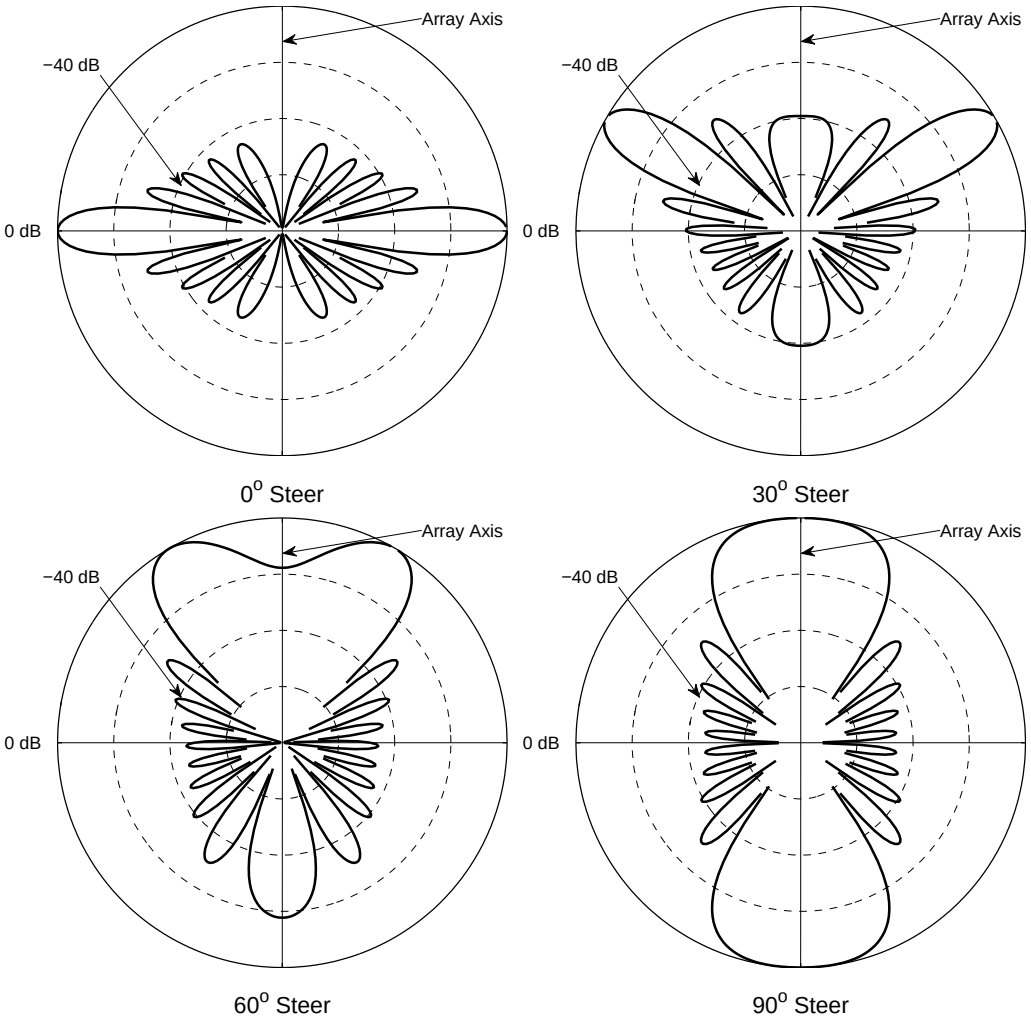


Figure 2.2: Effect of steer angle on the beam pattern

Note that $b(\theta)$ is in units of dB. Beam patterns for an array of 10 elements at a spacing $d = \lambda/2$ are shown in figure 2.2 for a range of steering angles. While the majority of the energy is emitted in the central lobe there is also significant emission in the side lobes. Normalising the intensity of these regions is covered in the next chapter.

2.4 Image Formation

The use of beam forming in one or two dimensions allows individual resolution cells to be insonified, such that a "photographic" image of an environment is produced. In SSS and SAS a projector array confines the sonar beam in the direction of travel (along track) such that a narrow strip of the sea-floor is insonified. MBES sonar utilise projector and receiver arrays, with the major axis of the arrays oriented perpendicular to each other, to insonify a 2D resolution cell. The key difference between the two systems is that in MBES only the first return is processed while in SSS, SAS the entire sonar return is processed. In MBES

systems the direction of the return is confined by the area insonified by the crossed beams. In SSS, SAS and FLS the direction is confined in one dimension by the beam shape and in the other it is determined by the time of flight. By taking multiple 1D slices of the environment a 2D picture is produced. Each 1D slice of the environment is referred to as a sonar ping. The sonar returns are discretised by their distance, or slant range, from the sonar platform.

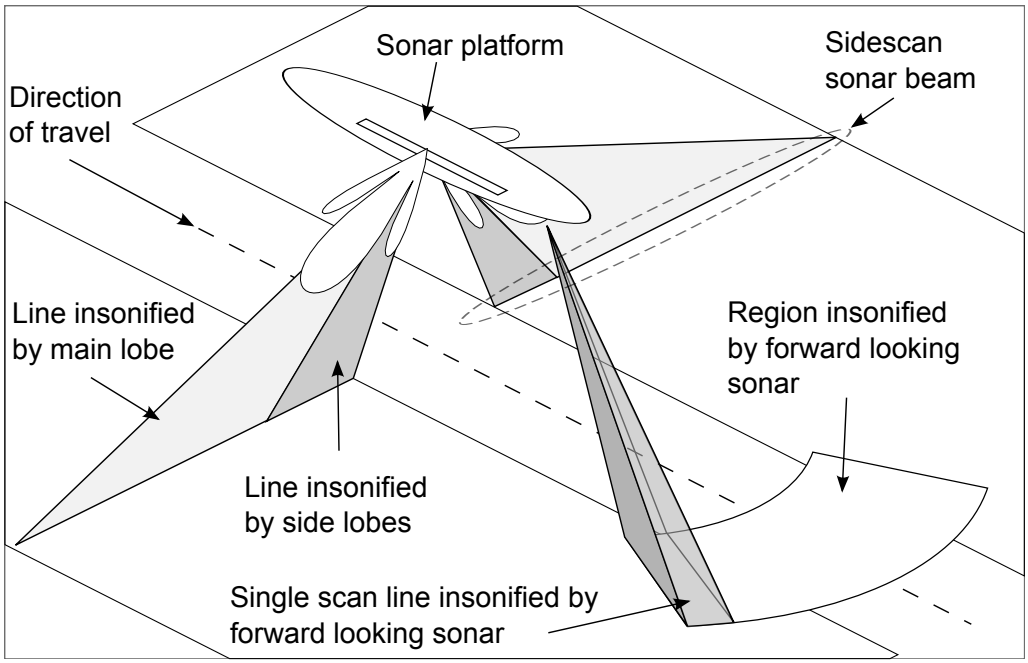


Figure 2.3: Sidescan and Forward Looking Sonar

The resolution of the sonar images is dependent on many factors. For both SSS and SAS, structures smaller than the wavelength of the sonar cannot be imaged. Therefore, there is a compromise between increased resolution and reduced range due to absorption. The resolution is also limited by the resolution to which the vehicle can control its heading. Small changes in heading at far range lead to ambiguity in the position of the return. While the sonar beam is confined in one dimension it is important to note that it is actually a 2D area of sea-floor which is insonified. Due to spherical spreading of the beam the width increases with range. Therefore, the resolution is highest close to the vehicle. In SSS the separation of each successive line of data is produced by motion of the vehicle. Therefore, the speed of the vehicle and the time between pings may limit the resolution. High resolution sonar typically have a resolution of a few centimetres per pixel and a range of around 50 meters.

SAS systems improve on the angular resolution of SSS systems by using sub-wavelength accurate navigation to synthetically extend the length of the sonar array. The signals from previous sonar pings are stored and coherently summed with subsequent pings. This allow beam-forming to be carried out with a synthetic array with a longer base-line than

that which could be installed on a vehicle.

The angular resolution of a sonar is directly related to the length of the array. This can be demonstrated with a simple geometric model as shown in figure 2.4. Two reflectors A and B are insonified by a sonar array of length L. Reflector A lies perpendicular to the array at a range r_0 while reflector B is at an angle $\beta/2$ and range r_1. The angular resolution is defined as the angular difference at which two reflectors will cause destructive interference at the receivers. Destructive interference first occurs when the difference in the path length to the reflector, for two elements of the array, is greater than $\lambda/4$. The greatest difference in path length is found for the elements at the end of the array, therefore only the central element and the end element are considered. These elements are a distance $L/2$ apart.

From figure 2.4 it can be seen that the difference in path length $\delta r = r_1 - r_0$ can be calculated in terms of the array length L and angle β:

$$\delta r = r_1 - r_0 = L/2sin(\beta/2) \tag{2.6}$$

This equation is solved for $\delta r = \lambda/4$ to give the angular resolution β.

$$\delta r = L/2sin(\beta/2) = \lambda/4 \tag{2.7}$$

Applying the small angular approximation and solving for beta we see that:

$$\beta \approx \lambda/4 \tag{2.8}$$

The angular resolution can either be increased by reducing the wavelength or increasing the array length. We have already stated that at high frequency the range is limited by absorption. SAS systems can synthetically increase the array length past the limits imposed by the physical size of the vehicle. Therefore, the angular resolution of a sonar system can be increased without changing the sonar frequency.

To understand how SAS systems reconstruct images from the synthetically extended array, we introduce the back-projection method of beam forming [17]. The aim of beam forming is to coherently sum the sonar return from a pixel x, y in order to estimate the reflectivity function $\gamma(x, y)$ of a scene. The geometry of a two element sonar system with a single receiver and transmitter is shown in figure 2.5. The range from the transmitter to the pixel of interest is r_t and the range from the pixel to the receiver is r_n. From equation 2.1 the signal delay t_n to a pixel is:

$$t_n(x, y) = (r_t + r_r)/c \tag{2.9}$$

where c is the speed of sound in water. In a SAS system the signals at time t_n are coher-

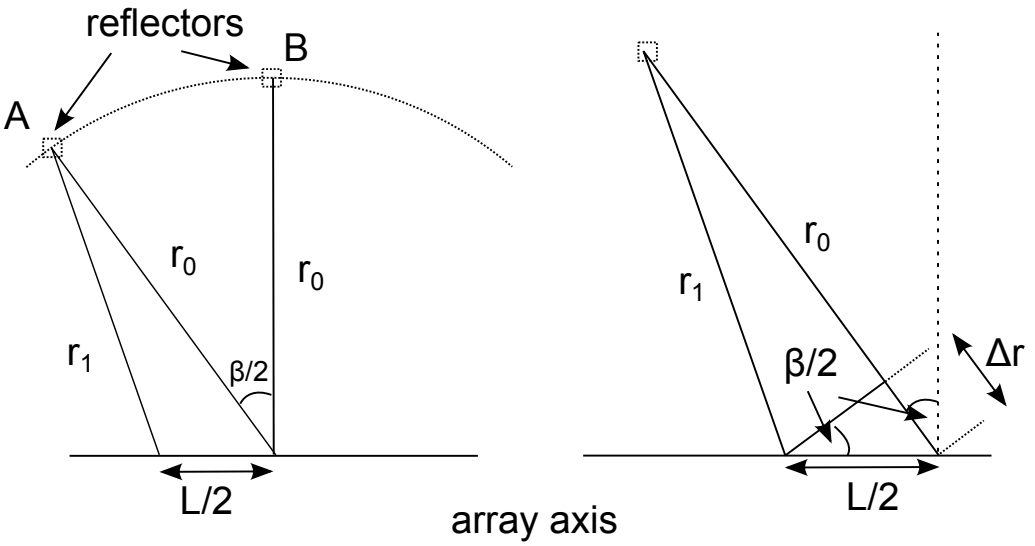


Figure 2.4: Geometrical relationship used to derive angular resolution

ently summed over n vehicle positions to produce an estimate of the reflectivity $\hat{\gamma}(x, y)$.

$$\hat{\gamma}(x, y) = \frac{1}{N} \sum_{n=1}^{N} s_n(t_n(x, y)) \tag{2.10}$$

Unlike SSS, where a narrow beam-width is desirable, SAS systems insonify a target from multiple angles. Each signal s_n at time t_n is the integral of an arc through the scene. If the navigation error is small compared to the wavelength of the sonar then signals will sum constructively for the pixel of interest.

The advantage of SAS systems over SSS is that the angular resolution is independent of the vehicle speed or the sonar range. However, SAS systems are more expensive and sensitive to vehicle stability and multi-path interference. Because of these reasons, the majority of commercial systems in use for MCM missions today still use SSS systems.

2.4.1 Appearance of Objects

Figure 2.6 shows an object imaged in the far field, where it is assumed that the height of the object is negligible with respect to the horizontal range. The object is characterised by a bright region, where sound is reflected from the object towards the sonar, and a dark region or shadow, caused by the object preventing the sonar pulse from insonifying the seabed behind the object.

The intensity of the sonar return is dependent on the material composition of the region which is insonified and the angle the surface presents to the sonar beam. Smooth reflective surfaces, angled towards the sonar, will appear brightest while rough absorbent surfaces will have a much lower return. Smooth regions angled away from the sonar, and shadow regions where the sonar is blocked by a large object, will consist entirely of

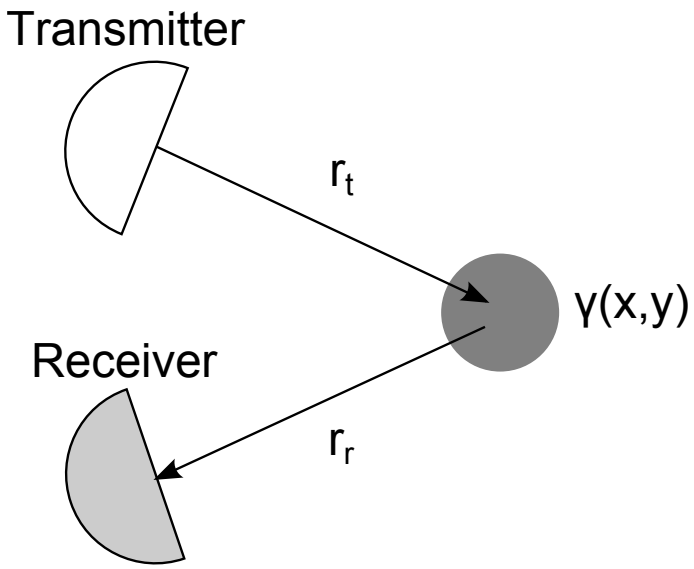


Figure 2.5: Geometry of a two element sonar system with a single transmitter and transmitter

volume reverberation (scattering from the medium). The appearance of objects is also affected by the imaging resolution.

Figure **??** shows the appearance of a cylinder imaged by a SSS and a SAS sonar. Both images show a highlight regions at the position of the target and a shadow region behind the target. However, there are several differences between the two images. The SAS image has a higher resolution and the resolution is high in both the along-track and across-track direction. The resolution of SSS images is limited in the along-track direction by the speed of the vehicle. The sound energy must travel to far-range and back to the vehicle before a subsequent sonar pulse can be emitted. There are also differences in the shadow shape behind the target. The shadow in the SSS image occurs directly behind the target as each ping is emitted perpendicular to the direction of motion of the vehicle. In the SAS image the target is viewed from multiple locations, therefore the shadow is less clear at the edges of the target. The next section describes how targets can be interpreted in the context of a real SSS image.

2.4.2 Interpretation of Sidescan Sonar and Synthetic Aperture Sonar Images

In this section we demonstrate how to interpret a real SSS image from a MarineSonics sonar operating at 30 m range. Figure 2.7 shows a typical SSS image in which Time Variable Gain (TVG) preprocessing has been applied to compensate for absorption and spherical spreading.

The horizontal axis corresponds to the across-track component, or the line insonified

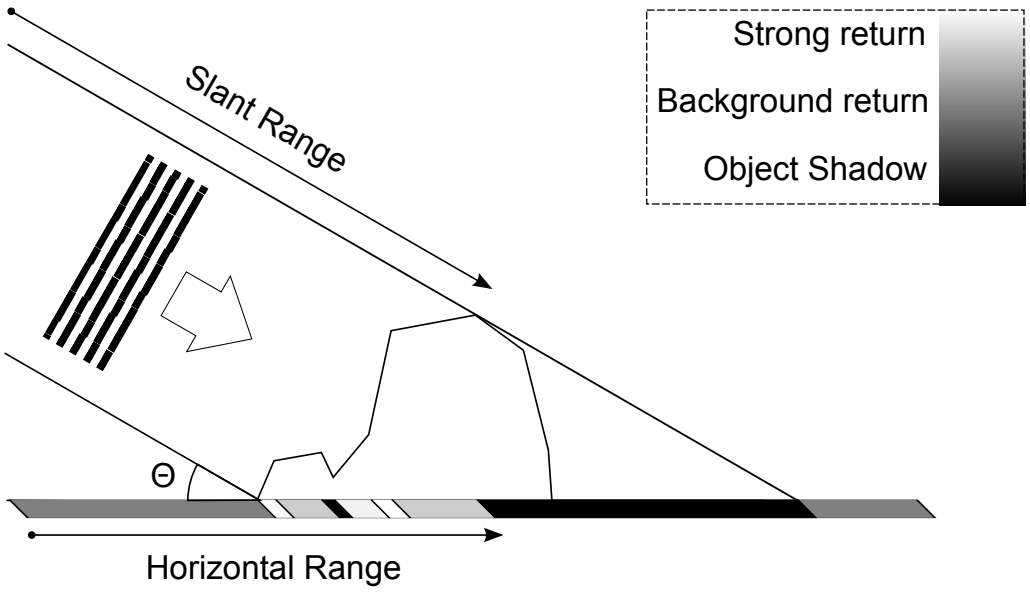


Figure 2.6: Appearance of objects in imaging sonar

by the sonar return. The vertical axis corresponds to the along-track component, with the motion of the sonar platform used to concatenate multiple sonar pings into an image. The image in the across track direction is in units of slant range or time of flight of the sonar ping. Considering the image from the center or nadir to the outside we observe the following features.

Water Column - The central, low intensity, region of the image consists entirely of volume reverberation from the ocean. The width of the water column is dependent on the height of the sonar platform and the minimum beam angle. Objects on the sea-floor such as rocks or fish may appear in the water column.

Side Lobes - For small beam angles the sea-floor will be insonified by the sonar side lobes, this results in several strips of varying intensity near the nadir.

Surface Return - In shallow water, sound energy can reflect from the surface. This causes a bright jagged line in the along-track direction with cross track range proportional to the depth of the sonar platform

SSS can suffer from other artefacts which are independent of range. These can be problematic for algorithms which process the data. The motion of the vehicle leads to several artefacts which are difficult to compensate for in the post processing. The vehicle roll causes one side of the image to appear compressed and the other elongated while yaw results in different areas of the sea-floor being insonified. Other sonar operating in the area can cause bright returns and multi-path effects can introduce artefacts in to the image. We cover the removal of these artefacts in the following chapter on pre-processing.

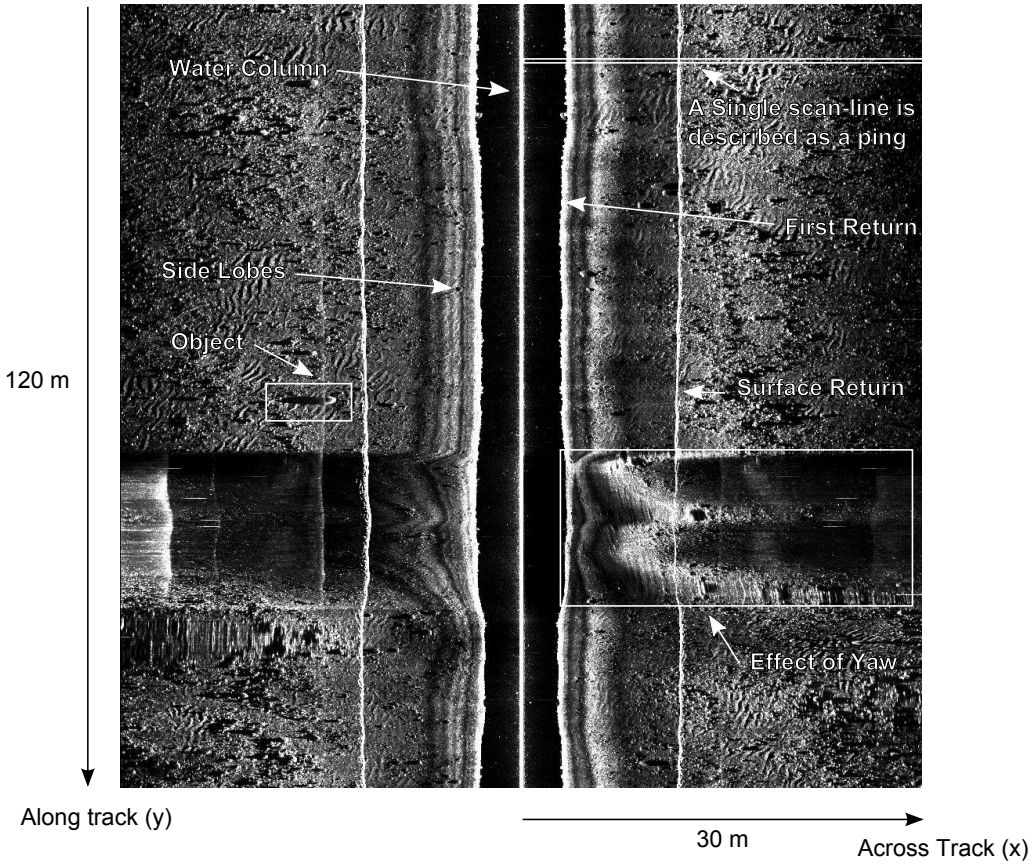


Figure 2.7: Example of a typical sidescan sonar image

2.5 Description of Data Set

In this section we describe the data that we have used to test the algorithms presented in this thesis. The data is draw from three different sonar types, Marine Sonic, Edgetech and MUSCLE SAS.

The **Marine Sonic** images are 1000x1024 pixels and have a resolution of approximately 6x12cm in the across track and along track direction respectively. The Marine Sonic sonar has a range of 30 m and each image represents a 0.007 km^2 region of the sea-floor. An example of a Marine Sonic image is shown in figure 2.7.

The **Edgetech** images are 2310x2000 pixels and have a resolution of approximately 3 x 12 cm in the across track and along track direction respectively. The Edgetech sonar is set to a range of 4 0m and each image represents a 0.019 km^2 region of the sea-floor. The sonar images include flat, rippled and cluttered regions of sea-floor. An example of a full Edgetech image can't be shown due to data restrictions. However, a simulated example of the target appearance is shown in figure 2.9.

The **MUSCLE** SAS system from CMRE represents the state of the art in SAS imagery worldwide. The MUSCLE SAS has a resolution of 2.5 cm x 1.5 cm in across and long track resolution and operates at 300 KHz with a operational range of up to 150 m. An

example of a single channel of MUSCLE SAS image is shown in figure 2.8.

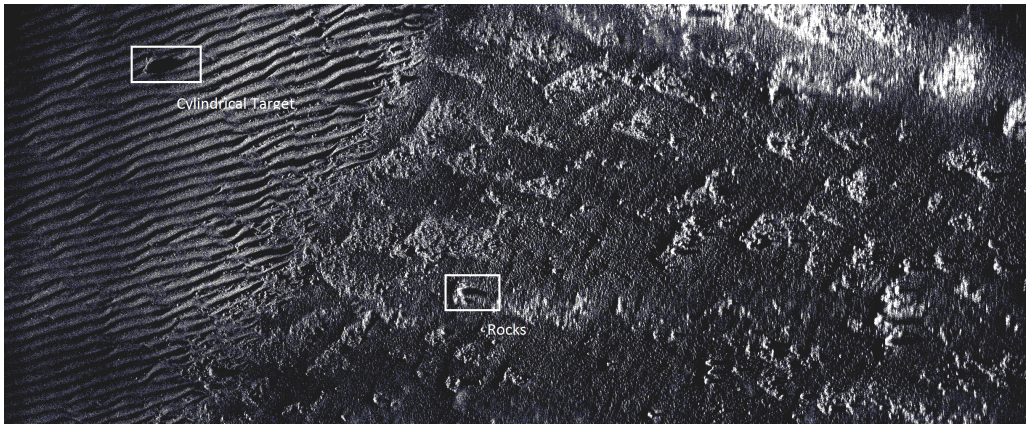


Figure 2.8: Example of a SAS image

The data is split into three data sets. The Edgetech data consists of 250 images randomly sampled from various Edgetech missions conducted using either an AUV or towfish. The Marine Sonic data set consists of images similarly samples. The data contains a range of flat, rippled and rocky sea-floor. Results are presented for all sea-floor types and separately for each sea-floor type. The MUSCLE SAS data set consists of three different sea-floor regions (B, C, D), the data was provided by Defence Science and Technology Laboratory (DSTL) and we have retained their naming convention. Areas B and D are mostly flat, with area D displaying marks from trawling. Area C is a mix of ripples, flat and rocky terrain and has a high density of Mine Like Objects (MLOs). Areas B and D can be considered as easy regions where we expect close to 100% detection. Area C is extremely hard and we would expect a large number of false alarms and missed detections. In the data set there are approximately 400 views of real cylinder, wedge and truncated cone targets (Some of these views represent repeat views of the same target from a different aspect). The MUSCLE SAS data was obtained towards the end of this thesis, therefore results are only presented for the SAS data in the final chapter.

Real data is expensive to collect and, apart from the MUSCLE SAS data set, is commercially sensitive or restricted. For this reason we use an augmented reality simulator to generate data for the majority of the tests [18]. We generate two different object types a 2 x 0.6 m cylinder and a 0.6 m x 0.6 m x 0.6 m truncated cone shape. The simulated targets are shown in figure 2.9. The simulated data represents a reasonable approximation to the real data, however there are a number of differences. Approximately 2% of the targets in our ground truth database fail to render and a further 10% have defects which are not typical of a real target. Additionally the cylindrical targets oriented end on to the sonar are much less visible that real targets in the same orientation. Consequently the object detection challenge is significantly harder than would be expected for real data. However, as we later compare two different methods on the same simulated data set the comparative

results are valid even if we under-estimate the performance of both algorithms.

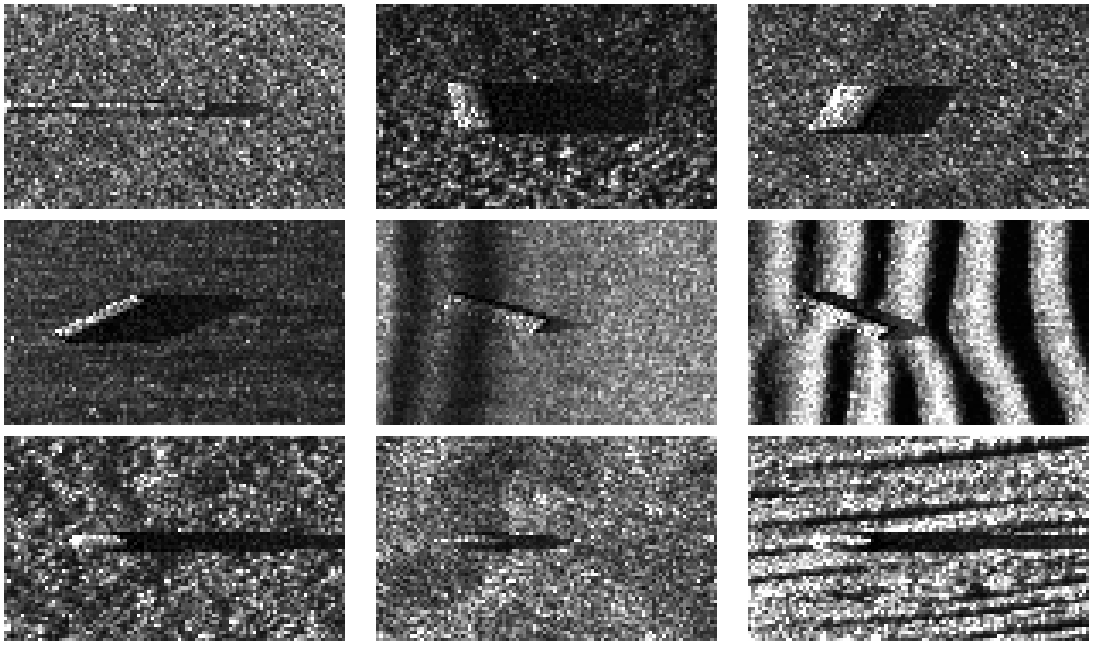


Figure 2.9: Examples of the augmented reality targets. Edgetech Cylinder (Top) Marine Sonic Cylinder (Middle) Edgetech Truncated Cone (Bottom)

In this section we have described how SSS and SAS images are formed, and how they can be interpreted. We have also introduced the three data sets that we use to test the work in later chapters. In the next part of this chapter we describe how sonar images can be processed to remove noise and common imaging artefacts.

2.6 Removing Artefacts from Sidescan Sonar Image

Image pre-processing is applied to improve the visual quality of Sidescan Sonar (SSS) images. However, the measure of visual quality is dependent on the application. For example in Mine Counter-Measure (MCM) operations the pre-processing is designed to increase the visibility of targets and remove target like artefacts [19]. Non-target objects and sea-floor regions can be distorted as long as the overall Probability of Detection (PD) and Probability of False Alarm (PFA) is improved. Alternatively, in a geological survey it is important to preserve sea-floor information such as the local statistics. Therefore, the pre-processing is limited to the removal of range and beam angle dependent effects [20]. In this section we cover correction for range and beam pattern dependent intensity variations, normalisation of the contrast, and removal of artefacts such as surface returns.

2.6.1 Range Intensity Variation and Beam Pattern

Range dependent and beam pattern dependent intensity variations are introduced by the physical process of forming a sonar image. Sonar manufacturers often apply a Time Variable Gain (TVG) function to reduce the effect. The TVG function models the sonar energy loss due to absorption and spherical spreading. Sonar systems, such as Marine Sonic, apply an automatic TVG to the image as it is collected. The TVG boosts the amplitude of the signal at far range to compensate for the energy loss. Other sonar such as Edgetech apply the TVG in software before the image is displayed. However, absorption is dependent on many parameters such as salinity and temperature. Additionally, the range to the sea-floor is dependent on the altitude of the vehicle and vehicle roll. Therefore, typically the TVG correction cannot account for all of these variables.

Moving Average and Curve Fitting

The simplest approach to removing range dependent effects is to calculate the average intensity with respect to range. For a scan-line (ping)with index y the intensity at range x is $I_y(x)$. The average intensity is $\overline{I}(x) = \text{mean}_y(I_y(x))$. A corrected intensity $I_C(x)$ is calculated by dividing each ping by the average profile $I_C(x) = I_y(x)/\overline{I}(x)$. This has the effect of moving the average intensity to 1. The image can then be scaled to an 8 bit image by multiplying the image by a constant and optionally applying some form of contrast stretching or gamma correction.

The approach assumes that if the number of pings used to calculate the across-track profile $I_C(x)$ is large then the average profile will describe the range dependent intensity changes. This is only correct if there are no significant changes in the altitude of the vehicle or the reflectivity of the sea-floor. Otherwise, a moving average filter can be preferable to a static average. The filter size is chosen such that the impact of any objects on the local average is negligible and the filter size is small compared to distance over which any altitude changes occur.

The moving average correction reduces the appearance of the beam pattern when the altitude of the vehicle is constant. However, when the vehicle changes altitude rapidly, or rolls, the correction can introduce object sized artefacts into the image. To prevent this a smaller filter size can be used. However, low pass filtering is required in the across track direction to reduce the impact of objects on the local average. We implement a method based on that of Cervenka and Moustier [21]. However, our implementation only applies the low pass-filtering method described in the paper.

The low pass filter is derived by approximating each ping as a Chebyshev polynomial. A Chebyshev polynomial $T_n(.)$ of degree n is defined as:

$$\tau_n(x) = \cos(n \arccos x), x \in [-1, 1] \tag{2.11}$$

The Chebyshev polynomials of first to fourth order are shown in figure 2.10. The Chebyshev polynomials are orthogonal:

$$\int_{-1}^{1} \tau_r(x)\tau_s(x)(1-x^2)^{-\frac{1}{2}} = N_r\delta_{rs} \tag{2.12}$$

with $N_0 = \pi$ and $N_r = \frac{1}{2}\pi$ if $r \neq 0$.

The orthogonality relation implies that $\{\tau_k\}_{k=0}^{n}$ is a set of linearly independent polynomials and a base of the linear vector space $\mathbb{P}_n$. Given a polynomial $P_n \in \mathbb{P}_n$ that interpolates a function f, we can write P_n as a combination of the base:

$$P_n(x) = \sum_{k=0}^{n}{}' c_k\tau_k(x) \tag{2.13}$$

where the prime indicates the first term is to be halved. The coefficients c_k can be written in the form:

$$c_k = \frac{2}{n+1}\sum_{j=0}^{n} f(\cos\theta_j)cos(k\theta_j), \theta_j = (j+1/2)\pi/(n+1) \tag{2.14}$$

Apart from the factor or $2/n+1$ this is the discrete cosine transform. Hence, the Chebyshev polynomial approximation of a function f can be calculated efficiently using the discrete cosine transform. For each scan-line $I_y(x)$ we divide the intensity by the seventh order Chebyshev polynomial $P_n(x)$ to generate the corrected intensity $I_C(x)$.

$$I_C(x) = I_y(x)/P_7(x) \tag{2.15}$$

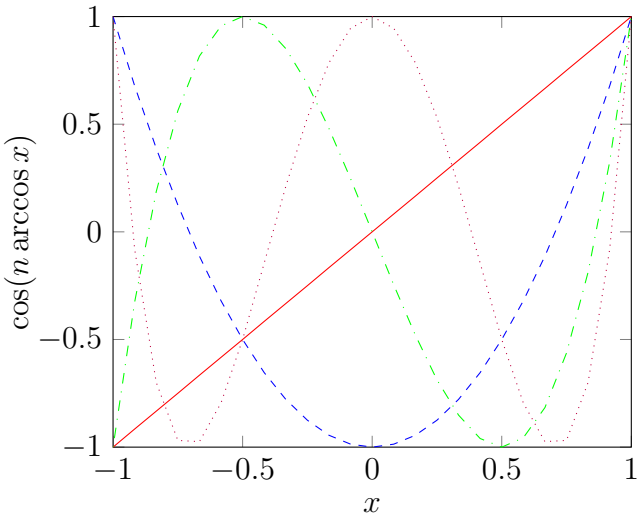
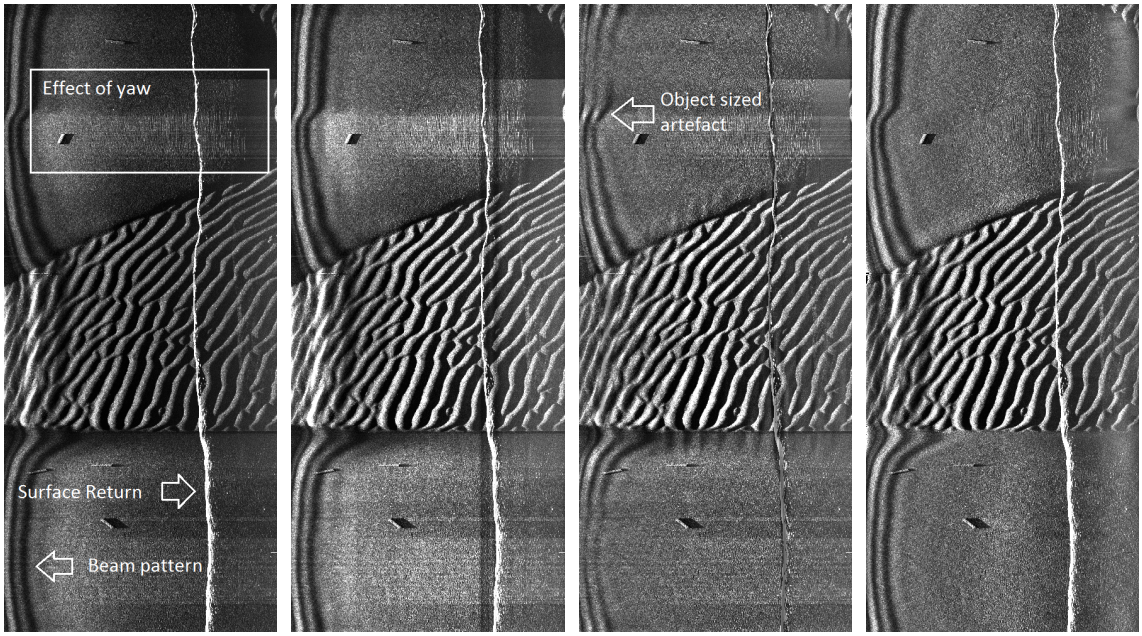


Figure 2.10: Chebyshev polynomials to fourth degree over the interval [-1,1]

In figure 2.11 we can see the comparison of the various approaches applied to the original 2.11(a). Figure 2.11(b) shows the image normalised by the average across-track



(a) Original Image (b) Average (c) Moving Average (d) Chebyshev

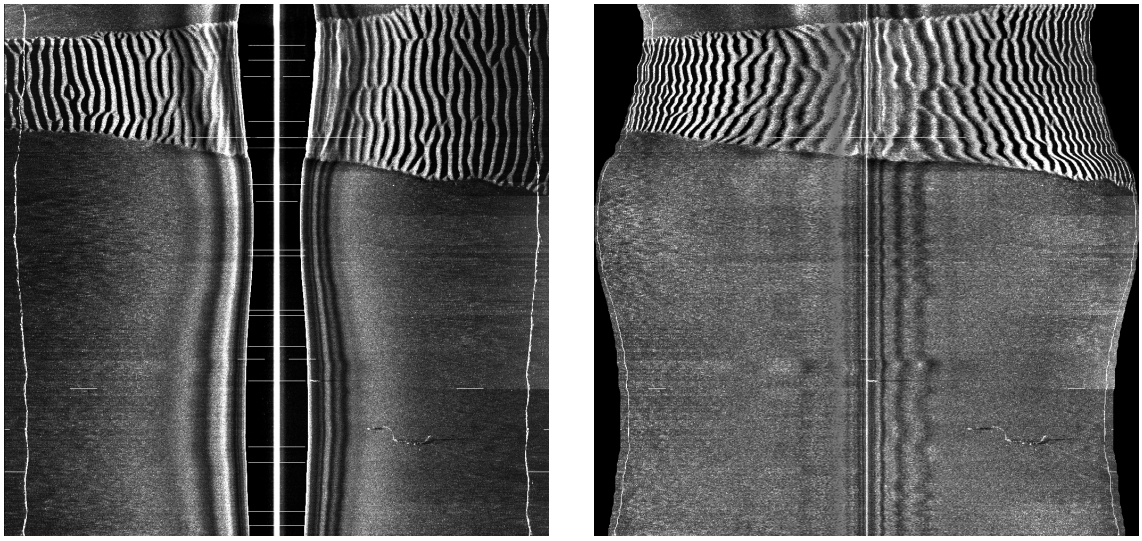
Figure 2.11: Comparison of a Marine Sonic image after is has been corrected using the average intensity, a moving average with a filter size of 100 pings and a 7th order Chebyshev approximation to each ping

intensity from all 1000 pings in the image. There is a slight increase in contrast and intensity at far range, but otherwise there is little change to the image. Figure 2.11(c) shows a moving average with a filter size of 100 pings in the along-track direction. The contrast of the beam pattern has been reduced. However, the approach has introduced significant object sized artefacts in the beam pattern and surface return. Figure 2.11(d) shows the original image after each ping has been divided by its 7th order Chebyshev polynomial approximation (Note that this is not the same as the more complicated approach described in [21]). While there is no effect on the beam pattern or surface return all of the intensity variations due to altitude and yaw have been removed.

Of the approaches presented in this section all except the Chebyshev approximation introduce additional artefacts into the data. The moving average approach is very poor in this respect as it creates a number of artefacts that would be detected by an Automatic Target Recognition (ATR) algorithm. There is a clear trade off between preserving real intensity changes due to the reflectivity of the sea-floor and removing unwanted intensity changes due to yaw and altitude. In the next two sections we investigate a number of approaches that remove both range and beam angle dependent intensity changes.

Re-sampling by Beam Angle

In the previous section we saw that a moving average filter could remove the beam angle dependent effects when the vehicle altitude is constant. If the vehicle altitude changes then



(a) Original Image (b) Aligned by Beam Angle

Figure 2.12: An example of an image re-sampled such that columns are aligned by beam angle. The standard average method is then applied to normalise the image

the intensity pattern caused by the beam forming is no longer aligned by column. Capus et al. [11] address this problem by re-sample the image such that the columns are aligned by beam angle rather than slant range. For each pixel in the image indexed by beam angle θ the corresponding position in the image indexed by slant range x is calculated as:

$$x = \frac{h}{cos(\theta)} \tag{2.16}$$

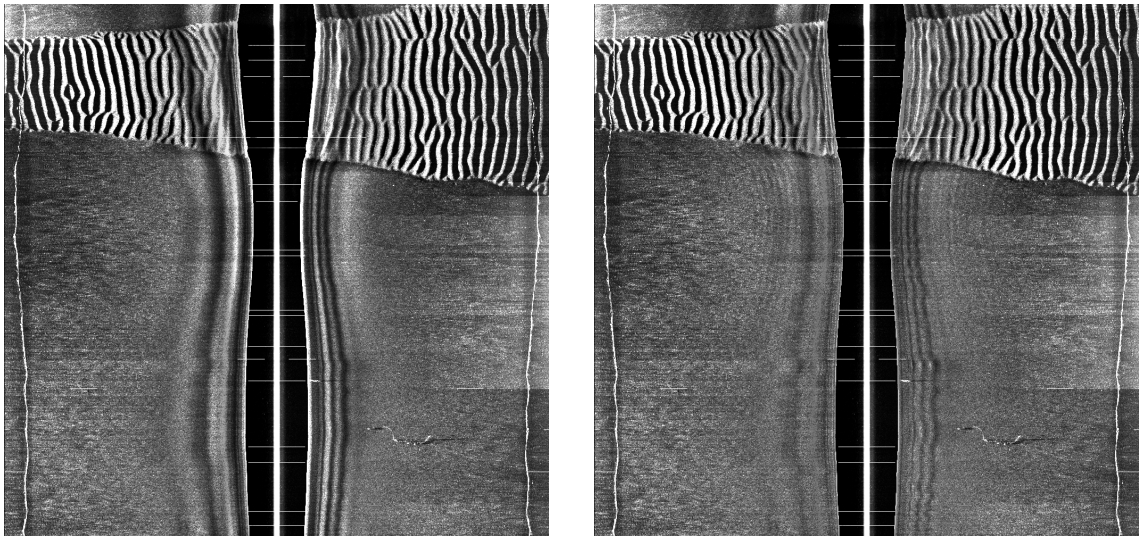
where, h is the vehicle altitude and sea-floor is assumed to be planar.

Figure 2.12 shows an example of an image re-sampled such that columns are aligned by beam angle. The standard averaging normalisation method is then applied to the image. If the beam angle is a linear function of column index then the image will be stretched at close range and compressed at far range. This results in a loss of detail at far range as several pixel values are interpolated to a single pixel in the corrected image. To offset this effect for we calculate $\theta_{corrected}$

$$\theta_{corrected} = \theta^{\gamma} \tag{2.17}$$

where γ is a constant. In this section we set $\gamma = 0.2$ however this is only relevant for viewing the image, the actual normalisation method is not sensitive to γ.

While at short range the intensity pattern is dominated by the beam pattern, at far range absorption dominates. For this reason Capus et al. [11] normalise the image at short range using the image re-sampled by beam angle and the standard normalisation approach at far range. In figure 2.13 the standard average normalisation method is compared to the method which first re-samples the image by beam angle. The appearance of the beam pattern is considerably reduced with respect to the more basic approach. However, a



(a) Average (b) Average + Beam Angle

Figure 2.13: A comparison of the average normalisation method with the normalisation after the image has been re-sampled by beam angle

number of object sized artefacts are created in the beam pattern. This is due to the fact that we have not accounted for the roll of the vehicle. Capus et al. address this problem using an optimisation approach to estimate the roll of the vehicle for each ping [20]. However, this approach is computationally intensive. In the next section we introduce a more efficient approach.

Serpentine Forward Backwards Filter

The Serpentine Forward-Backward Filter (SFBF) [22] is a computationally efficient method for estimating the local background intensity in SSS images. Dobeck's paper [22] first introduce the cross range forward backward filter. This approach is then extended to the SFBF method. In this section we give a brief overview of the algorithm, introduce the filter design and discuss the implementation of the path tracking and image normalisation. The SFBF is then compared to the other methods introduced in this chapter.

The SFBF uses adaptive filters to track paths of similar intensity through the image. A filter is initialised for each column and progresses from the top to the bottom of the image (forward) and from the bottom to the top of the image (backward). For each pixel, the filter with the estimate of the background intensity closest to that of the current pixel is selected. This allows the filter to track paths of similar intensity such as surface returns and beam patterns.

The image is filtered using a second order, critically damped, all-pole, low pass filter. The filter is designed to minimise the distortion of the background intensity due to objects in the image. The difference equation for the filter is

$$f(k) = c_1 f(k-1) + c_2 f(k-2) + d_0 r(k) \tag{2.18}$$

where the parameters are defined as

$r =$ Column of raw image
$f =$ Column of filtered image
$d_y =$ pixel size in cross range direction
$y_f =$ Correlation distance of filter
$a = exp(-d_y/y_f)$
$c_1 = 2a$
$c_2 = -a^2$
$d_0 = (1-a)^2$

The cut-off frequency for the low pass filter is chosen to minimise target distortion, this is achieved by optimising the correlation distance y_f of the filter. The target size is denoted T_s and the background size B_s. Defining a figure of merit for target $G_1(y)$ and background distortion $G_2(y)$, y_f is chosen such that $G_1(y_f) = G_2(y_f)$

$$\begin{aligned} T_1(y) &= 0.5T_s/y \\ G_1(y) &= 1 - (1 - T_1(y))exp(-T_1(y)) \end{aligned} \tag{2.19}$$

$$\begin{aligned} T_2(y) &= (0.5B_s - T_s)/y \\ G_2(y) &= (1 - T_2(y))exp(-T_2(y)) \end{aligned} \tag{2.20}$$

The filter response to an intensity spike similar to that caused by a target is shown in Figure 2.14. While the effect of target size regions is minimised by the low pass filter the background mean increases after the target To prevent this effect, the local mean for a pixel is shifted by T_s pixels such that the target does not compromise the local mean. A similar problem is encountered when the filter crosses a boundary between different sea-floor types. The rapid change in intensity causes artefacts in the image. However, the forwards-backwards filtering creates two estimates of the local mean for each pixel. Therefore, the estimate of the mean which is closest to the pixel is chosen. The implementation of the path tracking can be found in [22] and is not repeated here.

Figure 2.15 shows two examples of images normalised using the SFBF where $T_s = 40$ pixels and $B_s = 100$ pixels. In 2.15(a) the beam pattern and surface return have both been almost completely removed and there is no range dependent variation in intensity. Of particular note is the object in the water column on the left hand side of the image. The object remains undistorted even though the beam pattern has been corrected on either side. However, the highlight on the object on the right hand side of the image that intersects the surface return has been reduced in intensity. Here, the algorithm is unable

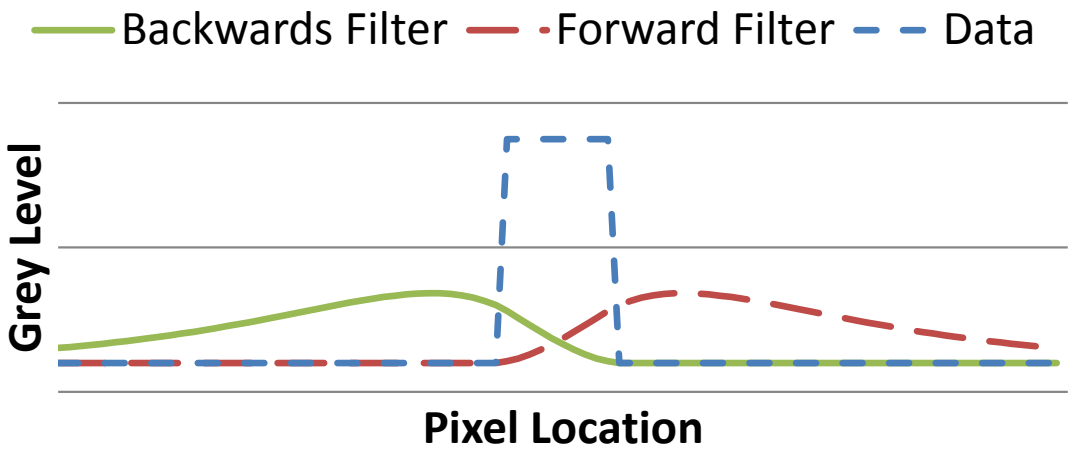


Figure 2.14: Response of the second order filter used by SFBF to an intensity spike similar to that caused by a target

to separate the object highlight from the surface return. In 2.15(b) another problem with the SFBF is demonstrated. Ripples that are longer than the target size preserved by the filter are normalised to the mean background level. Unlike the other methods that we have implemented the SFBF removes many of the common artefacts that corrupt sonar images. Additionally, it does not introduce object like artefacts into the image. However, as large sea-floor features such as ripples are removed by the algorithm it is unsuitable for applications such as sea-floor classification.

Discussion

In this section we reviewed a number of algorithms for removing unwanted artefacts from SSS images. As we will see in later chapters these artefacts can cause false alarms in ATR algorithms and affect the results of sea-floor characterisation algorithms. For object detection the SFBF algorithm is the most effective. The beam pattern and surface return, which are the main cause of false alarms in otherwise flat regions, are removed. In addition no object sized artefacts are introduced to the image by the algorithm. However, the tendency of the algorithm to distort large background features such as as ripples means that the algorithm is unsuitable for sea-floor characterisation.

For sea-floor characterisation we apply the method in section 2.6.1 where the image is first re-sampled by beam angle. This algorithm removes the beam pattern from the image which would otherwise be mis-classified as sand ripples by a machine learning algorithm. Additionally, if a global average is applied then the relative intensity of the sea-floor is preserved over the entire mission. While there are some object sized artefacts introduced due to the yaw and roll of the vehicle, they are too small to affect the results of the sea-floor characterisation algorithm. However, unlike the SFBF, this algorithm does not remove the surface return from the image. This problem is addressed in the next

section.

2.6.2 Water Column and Surface Return

The surface return is a line of saturated (or bright) pixels in SSS images caused by the sonar energy reflecting from the surface of the water. Depending on the surface conditions it can vary in width and intensity, and have multiple disconnected branches. This complexity makes it a prime candidate for false alarms in ATR algorithms. For this reason the current approach at SeeByte is to use the navigation information to remove ATR detections that could be due to the surface return. In this section we present a fast algorithm for identifying the position of the water column and the surface return. These regions are then removed from the image with little distortion to surrounding targets and sea-floor. We also present a modification of the original algorithm that can be used to find the altitude of the vehicle by tracking the position of the water column.

While a number of papers [4, 23, 24, 25, 26, 27, 28] mention problems caused due to the surface return, there are relatively few methods for removing it. One such method, the SFBF [22] was covered in section 2.6.1. Here we consider the method of Crosby and Cobb [29]. Surface returns are detected using using an intensity threshold followed by a morphological operator to join vertically connected regions. A threshold is then applied to the length of each connected component to identify surface returns. The threshold value and the exact morphological operation are not mentioned in the paper. However, we assume from the context that a morphological closing operator is used. Our implementation of the algorithm is shown in listing 2.1.

The effect of the algorithm is to close the gaps between any saturated, vertically connected pixels. However, as we can see from figure 2.16 if the image contains a large number of saturated pixels, for example in complex regions or ripple fields, then the algorithm marks many pixels that are not part of the surface return. In the next section we introduce a graph based path finding method that can identify the surface return even in these difficult regions.

A* Search

The A* search [30] is a graph traversal algorithm which uses a best first search to minimise the cost of travelling from an initial vertex a to a goal vertex b. Given an undirected graph $G =< V, E >$ where $v \in V$ is a set of vertices and $(u, v) \in E$ are a set of edges, we define a path cost function $c(u, v)$ and a heuristic function $h(v)$. The path cost function describes the cost of moving from one node to another while the heuristic function estimates the minimum cost to reach the goal node from the current node. As the algorithm traverses the graph we track a past-path cost $c(a, v)$ and a future-path cost which is simply $h(v)$. The algorithm will always visit the vertex which minimises the sum of past and future

Listing 2.1: Implementation of the method of Crosby and Cobb

```
void FindSurfaceReturnMask( const cv::Mat& r_In ,cv::Mat& r_Out )
{
    cv::Mat mask = r_In > 0.99;
    //Create an hour glass shaped kernel
    cv::Mat_<uchar> kernel(40,10);
    //Draw the top left part of the kernel
    for(int iRow = 0;iRow < kernel.rows/2;iRow++)
    {
          for(int iCol = 0;iCol < kernel.cols/2;iCol++)
          {
              kernel(iRow,iCol) = iRow < iCol + 16 ? 255 : 0;
          }
    }
    cv::flip(kernel(cv::Rect(0,0,kernel.cols/2,kernel.rows/2)),
    kernel(cv::Rect(kernel.cols/2,0,kernel.cols/2,kernel.rows/2)),1);
    cv::flip(kernel(cv::Rect(0,0,kernel.cols,kernel.rows/2)),
    kernel(cv::Rect(0,kernel.rows/2,kernel.cols,kernel.rows/2)),-1);
    //Perform morphological closing with the kernel
    cv::morphologyEx(mask,mask,cv::MORPH_CLOSE,kernel);
    r_Out = mask;
}
```

cost. If the heuristic is an admissible heuristic, it does not overestimate the distance to the goal vertex, then the A* algorithm will always converge to the lowest cost path.

Finding Surface Returns with the A* Search

The surface return can be viewed as a path of saturated pixels that connects the top of the SSS image to the bottom. The path can potentially contain gaps and non-saturated pixels. Additionally, we wish to express our prior knowledge that the surface return is more likely to be connected vertically than horizontally. We start by defining the graph that represents our image. We assume that we are working with either the port or starboard channel of the sonar such that there will only be a single surface return.

We first define the connectivity of the graph representing our SSS image, this is shown in figure 2.17. Each pixel in the image is represented by a node, nodes are connected in a 4-connected neighbourhood such that the search can progress to the North, South, East and West of the current node. An additional source and goal node are connected to the top row and bottom row of the image respectively. A cost $c(u, v)$ is associated with traversing each vertex. The cost is based on the intensity $I(.)$ of the destination pixel v, where I_{Max} represents the intensity of a saturated pixel. There is an additional cost relative to the distance travelled $d(u, v)$. The distance between neighbouring nodes is 1.

$$c(u,v) = \begin{cases} 0 & \text{if } I(v)/I_{Max} > 0.99 \\ 1 & \text{otherwise} \end{cases} + d(u,v) \tag{2.21}$$

From equation 2.21 we can see that there is a compromise between distance travelled and the intensity of the pixels on the route. First we justify the intensity cost. The surface return is usually formed of saturated pixels, or pixels that are close to saturated. Therefore, there is a cost for moving to non-saturated pixels as these are unlikely to form part of the surface return. The distance cost penalises the length of the path. If all neighbouring pixels are saturated the algorithm will obtain the minimum cost by moving vertically towards the bottom of the image. This optimal behaviour is described by the heuristic function.

The heuristic function is the minimum cost to traverse from the current node to the goal. The minimum cost will be reached for a straight vertical line of pixels with intensity I_{Max}. Therefore $h(v) = d_{vertical}(v, goal)$ is the vertical distance to the bottom of the image. We first present the algorithm using this admissible heuristic. However, we can significantly increase the performance of the algorithm by removing the cost for vertical movement. The heuristic function no-longer estimates the minimum cost, therefore the solution may not be optimal. However, we show that by initialising the algorithm using the position of the surface return in the previous image we can obtain close to 100% accuracy over the entire data set

Results and Analysis

In this section we present the results of the A* algorithm over a mission with 200 Marine Sonic SSS images. We examine the performance of the algorithm in complex and rippled areas where there are many saturated pixels that do not belong to the surface return. First we consider the effect of the heuristic on the accuracy and runtime of the algorithm. In figure 2.18 we show the evolution of the algorithm with an admissible heuristic. The algorithm finds the correct path. However, for each vertical gap in the surface return the algorithm must search an entire row at the top of the image.

In figure 2.19 we set the vertical movement cost to zero. Now the heuristic over estimates the cost of moving from any vertex to the goal vertex. The effect is to bias the algorithm to searching the path that is closest to the bottom of the image. Assuming that the surface return is complete and connected this will always give the same result as the admissible heuristic. However, there is a trade off between accuracy and speed. In figure 2.20 we compare the algorithm with an admissible heuristic to the algorithm with the inadmissible heuristic. The algorithm with the inadmissible heuristic follows the ripples rather than the surface return. The cost of traversing the non-saturated pixels at the bottom of the image is not sufficient to force the algorithm to search the top of the image and find the correct surface return.

In some images the beam pattern causes a saturated band of pixels close to the water column. This can often be identified instead of the surface return. The simplest approach

to this problem is to remove the water column from the graph. In 9 of the test images the surface return overlapped with the masked region and the algorithm failed to correctly identify the surface return. Over the entire data set of 200 images, the algorithm with an admissible heuristic correctly identified the surface return in 191 images and completed in 30.8s. The algorithm with zero vertical movement cost identified the surface return in 190 images and completed in 4.4s. The algorithm with zero vertical movement cost initialised with the position of the surface return from the previous image identified the surface return in 191 of the images and completed in 2.1s.

Finding Water Column with the A* Search

A similar approach can be used to find the position of the water column in the image. However, instead of finding a path in the intensity image we first calculate the gradient image $G(.)$ using a 7 pixel wide Sobel filter in the x direction. The algorithm finds the vertical path that maximises the gradient. Unlike the surface return, the water column is not saturated at the maximum gradient, therefore we penalise the distance of the gradient from the maximum.

$$c(u,v) = \frac{G_{Max} - G(v)}{G_{Max}} + \alpha d(u,v) \tag{2.22}$$

To prevent the distance cost from dominating it is scaled by a constant α. The constant α determines whether the algorithm will choose between the shortest path or the path that best satisfies the gradient cost. This allows the algorithm to route through objects that protrude into the water column as can be seen in figure 2.21. An experimentally determined value of $\alpha = 0.2$ was used for the results shown in this chapter.

Results

We now discuss the performance of the algorithm for finding the water column. Figure 2.21 shows an example of the gradient image used to identify the water column. We have already determined that the A* search with zero vertical movement cost initialised from the previous image is as accurate as the same algorithm with an admissible heuristic and significantly faster. Therefore, only this version of the algorithm is tested. In figure 2.21 we show an example of the gradient image and the identified water column. The behaviour of the algorithm is particularly interesting where there are objects in the water column. Here the horizontal movement cost penalty prevents the algorithm from routing round the obstacle. This is a significant improvement over simpler algorithms which use the first return to estimate the position of the water column. These algorithms confuse the first bright pixel with the first-return from the sea-floor. Sonar noise or objects such as fish can lead to the algorithm returning the wrong location for the water column.

There are relatively few algorithms for identifying the position of the surface return and water column and we have shown that these existing algorithms can fail in complicated regions. In this section we introduced a simple and robust approach to accurately track the position of both the water column and the surface return. Figure 2.22 shows an initial attempt at removing the surface return from the image. The local statistics are estimated in an 11 pixel wide region about the water column and a randomly generated pixel is used to replace the water column. In 2.22(c) the surface return in the original 2.22(a) has been completely removed and both object highlight and shadow regions have been accurately reconstructed. However, in 2.22(d) this simple approach is unable to reconstruct the regular ripple pattern in the original 2.22(b).

In this section we have implemented and compared a number of different methods for removing artefacts from SSS images. In the next section we introduce a novel algorithm to segment SSS sonar images into object and background regions.

2.7 Sidescan Sonar Segmentation

In SSS segmentation, for mine detection, the image is segmented into three classes. Sea-floor reverberation (background), return from objects (highlight) and acoustic shadow (shadow). SSS segmentation forms the basis of many object detection and classification algorithms [24, 31, 32, 33, 34, 35]. At SeeByte SSS segmentation is used to calculate shape characteristics from detected objects for ATR and in chapter 5 we calculate characteristics of objects on the sea-floor from a segmented image.

Existing SSS segmentation algorithms fall into two classes. Markovian segmentation based on a statistical model of the classes [31, 36, 37] and an edge preserving de-noising filter followed by an intensity threshold [38]. The current approach to segmenting SSS images at SeeByte is to model the statistical properties of each class using a Markov Random Field (MRF) and find the minimum energy state using hierarchical Iterated Conditional Modes (ICM). In the current implementations [24] of this method there is a trade off between the accuracy and the speed of the segmentation. In this section we modify the algorithm such that it can be modelled as a flow-graph and solved using graph-cuts. Graph-cuts is guaranteed to converge to the minimum energy state of the MRF and is computationally efficient.

2.7.1 Graph Cuts

Many problems in computer vision can be described as an energy minimisation problem. In chapter 3 two such examples of energy minimisation, the MRF and the active contour model, are presented. These methods provide a framework for describing a segmentation problem as an energy equation to be minimised. This section will provide a more formal

introduction to MRFs and discuss two computational methods for solving these systems.

The first method discussed is ICM and the second is the max-flow min-cut graph segmentation algorithm. The ICM algorithm has traditionally been used to solve energy minimisation problems in the underwater imaging field. However, experimental comparisons of these two algorithms show that multi-label segmentation algorithms based on graph-cuts significantly out perform the ICM algorithm [39].

Markov Random Field

The MRF is popular in both physics and computer science for describing the energy interactions in a system. The total energy associated with a MRF can be described the sum of the local interactions over the entire system. Typically to gain insight into the system one seeks to find the state of the system that minimises the total energy of the system. In a segmentation problem the minimum energy will correspond to the set of class labels that describes the optimal segmentation of the image.

A MRF is modelled as an undirected graph $G = \langle V, E \rangle$, where V is a set of vertices and E is a set of edges. An edge $e \in E$ is the set of connected vertices u, v (figure). The graph encodes the interactions between a set of random variables $X = (X_v)_{v \in V}$. If the set of random variables X obeys the local Markov properties then they form a MRF with respect to G:

Pairwise Markov property: Any two non-adjacent variables are conditionally independent given all other variables:

$$x_u \perp\!\!\!\perp X_v \mid x_{V \setminus \{u,v\}} \quad \text{if } \{u, v\} \notin E \tag{2.23}$$

Local Markov property A variable is conditionally independent of all other variables given its neighbours:

$$X_v \perp\!\!\!\perp X_{V \setminus \mathrm{cl}(v)} \mid X_{\mathrm{ne}(v)} \tag{2.24}$$

where ne(v) is the set of neighbours of v, and cl(v) = $v \cup$ ne(v) is the closed neighbourhood of v.

Global Markov property: Any two subsets of variables are conditionally independent given a separating subset:

$$X_A \perp\!\!\!\perp X_B \mid X_S \tag{2.25}$$

where every path from a node in A to a node in B passes through S.

The MRF describes the pairwise interactions between nodes. In the image processing domain a MRF would typically be used for applications such as edge preserving smoothing. For the segmentation problem the MRF must be conditionally dependent on

an external label field. The Conditional Random Field (CRF) is described in the next section.

Maximum a Posteriori Conditional Random Field

The Maximum a Posteriori (MAP)-MRF formulation is used to obtain an estimate of an unobserved population parameter from a set of observations. A segmentation of an image made up of a set of pixels $p \in P$ is described by the label field $L_p|p \in P$. The observations $x_p|p \in P$ could be the grey level intensities of the pixels or a feature vector calculated from the image. There are two assumptions, firstly x_p are conditionally independent give L_p and secondly L_p forms a MRF. Therefore, the joint probability of L_p conditioned on x_p is:

$$P(L_p|x_p, L_{\text{ne}(p)}) = \frac{P(x_p|L_p) \cdot P(L_p|L_{\text{ne}(p)})}{P(x)} \tag{2.26}$$

where $P(x)$ is a normalising constant. It is sufficient to obtain proportional probabilities therefore:

$$P(L_p|x_p, L_{\text{ne}(p)}) \propto P(x_p|L_p) \cdot P(L_p|L_{\text{ne}(p)}) \tag{2.27}$$

The conditions for representing a problems as an MRF can be simplified using the Hammersley-Clifford Theorem. Applying this theorem allows the probabilities to be expressed as potentials, simultaneously guaranteeing that the field is Markovian. The Hammersley-Clifford Theorem states that if a random field x has the local Markov property then it can be written as a Gibbs distribution:

$$P(x) = \frac{1}{Z} exp\left(-\sum_{c \in C} V_c(x_c)\right) \tag{2.28}$$

where C is the set of maximal cliques of a graph G, Z is a normalising constant and $V_c(x_c)$ are the clique potentials. If $P(x)$ can be written in the Gibbs form then it has the global Markov property. The fundamental consequence of this theorem is that the probability $P(L_p|x_p, L_{\text{ne}(p)})$ can be written as:

$$P(L_p|x_p, L_{\text{ne}(p)}) \propto exp(-U(L_p|x_p, L_{\text{ne}(p)})) \tag{2.29}$$

where $U(L_p|x_p, L_{\text{ne}(p)})$ is the posterior energy and satisfies

$$U(L_p|x_p, L_{\text{ne}(p)}) = U(x_p|L_p) + U(L_p|L_{\text{ne}(p)}) \tag{2.30}$$

$$= U(x_p|L_p) + \sum_{v \in \text{ne}(p)} U(L_p|L_v) \tag{2.31}$$

$U(x_p|L_p)$ is the data term and describes the likelihood of observing a feature vector x_p given a label L_p. $\sum_{v\in \text{ne}(p)} U(L_p|L_v)$ is the smoothness term and assigns an energy cost to neighbouring pixels taking on a different label. The MAP estimate is founding by maximising the probability over the label field, or equivalently minimising the energy

$$L = \arg\min_{L} U(L|x, \text{ne}(L)) \tag{2.32}$$

The energy of any system that can be expressed in this way can be minimised using a rich background of mathematical techniques. In this section two techniques, ICM and the minimum cut on a directed flow graph, are introduced.

Iterated Conditional Modes

Iterated Conditional Modes (ICM) is a greedy algorithm that operates by iteratively switching each pixel to the state with the highest probability or lowest energy cost. The algorithm (Algorithm 1) iterates over all the nodes in the graph until convergence. The algorithm is fast but is likely to converge to a local minima. This can be avoided by methods such as simulated annealing or Markov Chain Monte Carlo (MCMC), however these algorithms are known to be much slower to converge. For binary segmentation, advances in network flow algorithms have produced algorithms which are guaranteed to converge to the global minima in approximately linear time. In the next section we describe how a binary CRF can be reformulated as a min-cut problem on a flow graph.

Input: Graph $G = \langle V, E\rangle$, Labels $l_v \in L$, Features $x_v \in X$, Data term $f_D(.) : x_v \mapsto \mathbb{R}$, Smoothness term $f_S(.) : x_{\text{ne}(v)} \mapsto \mathbb{R}$

Output: Labels $l_v \in L$

Set all labels to minimise data cost $L = \arg\min_L \sum_{v\in V} f_D(l_v)$;

while *total energy is not a local minima* **do**

  foreach $v \in V$ **do**

    minimise data and smoothness cost for vertex

    $l_v = \arg\min_{l_v} f_D(l_v) + f_S(l_v)$;

  end

end

return L

Algorithm 1: Iterated Conditional Modes (ICM) algorithm

Flow Graphs

The minimum energy state of a MRF can be calculated by posing the problem as a flow graph. The advantage of this technique is that, for a binary segmentation, the algorithm will always converge to the global minima. A flow graph $G =< V, E >$ is analogous to a network of pipes. The pipe or edges $(u, v) \in E$ connect a source s and a sink t $s, t \in V$,

with fluid flowing from the source to the sink. The flow along an edge is denoted $f_{u,v}$ and the capacity $c_{u,v}$. The flow may not exceed the capacity such that $f_{u,v} \leq c_{u,v}$ and flow must be conserved $\sum_{u:\ (u,v)\in E} f_{uv} = \sum_{u:\ (v,u)\in E} f_{vu}$ for each $v \in V \setminus \{s,t\}$. The value of the flow is the sum of the flow coming from the source.

$$|f| = \sum_{v:(s,v)\in E} f_{s,v} \tag{2.33}$$

The flow graph is solved to find the routing that is associated with the maximum flow being pushed from the source to the sink $f_{max} = \text{argmax}|f|$.

The primal-dual problem to calculating the maximum flow is calculating the minimum s-t cut on the graph. An s-t cut $C = (S,T)$ is a partition of V such that $s \in S$ and $t \in T$. The capacity of an s-t cut is defined by the sum of the capacities of all the edges that connect the source and sink sets. Only edges that go from the source to the sink are include in the capacity.

$$c(S,T) = \sum_{(u,v)\in S\times T} c_{uv} \tag{2.34}$$

If a problem can be posed as a max-flow problem then the min-cut can be obtained from the solution to the max-flow problem. This is the max-flow min-cut theorem: In any network, the value of max flow equals capacity of min cut. This is illustrated this with a simple example.

Consider the directed graph in figure 2.23, the capacity (c) and the flow (f) for each edge are labelled c / f. The flows have been chosen such that the flow from the source (s) to the sink (t) are maximised. The cut set S is shown in red and the cut set T is shown in blue. The cut shown is the minimum cut and has a capacity of $(3+4+2=11)$. It can be seen that any other cut on the graph would result in a higher cut capacity. However, for a larger graph an algorithmic approach is required.

Finding the min-cut from a max-flow solution is achieved by conducting a depth first search from the source along all edges that are not saturated, i.e. the flow is less than the capacity. Vertices that are visited are marked. The min-cut consists of all the edges that have one visited vertex and one unvisited vertex. These edges bust be saturated and belong to the min-cut. Some algorithms derive this information while solving the max-flow problem, therefore this step does not need to be performed explicitly. It is also worth noting that there can be more than one min-cut on a graph. We now look at how a CRF can be formulated as a flow graph.

Representing a CRF as flow graph

In section 2.7.1 it was shown that the state with maximum likelihood could be found by minimising an energy function of the form

$$U(L|x, \text{ne}(L)) = \sum_{p \in V} U(x_p|L_p) + \sum_{p \in V, v \in \text{ne}(p)} U(L_p|L_v) \tag{2.35}$$

For the 2 label problem $L_p \in 0, 1$ it will be shown that this equation can be reformulated as an s-t cut problem. The aim is to rewrite the energy terms as capacities. Let $g_p^{L_p}$ be the energy cost associated with assigning a label L_p to a vertex p and $g_{pq}^{L_p L_q}$ be the neighbourhood potential conditioned on the label assigned to each vertex.

$$\begin{aligned} U(L|x, \text{ne}(L)) &= \sum_{p \in V} U(x_p|L_p) + \sum_{p \in V, q \in \text{ne}(p)} U(L_p|L_q) \\ &= \sum_{p \in V} (g_p^1 L_p + g_p^0 (1 - L_p)) \\ &+ \sum_{p \in V, q \in \text{ne}(p)} (g_{pq}^{00}(1 - L_p)(1 - L_q) \\ &+ g_{pq}^{01}(1 - L_p)L_q + g_{pq}^{10} L_p (1 - L_q) + g_{pq}^{11} L_p L_q \end{aligned} \tag{2.36}$$

The neighbourhood energy can be rewritten as

$$U(L_p|L_q) = K_{pq} + g'_p L_p + g'_q L_q + c_{pq}(1 - L_p)L_q/2 + c_{pq} L_p (1 - L_q)/2 \tag{2.37}$$

where

$$K_{pq} = g_p q^{00} \tag{2.38}$$

$$g'_p = \frac{g_{pq}^{01} + g_{pq}^{11} - g_{pq}^{10} - g_{pq}^{00}}{2} \tag{2.39}$$

$$g'_q = \frac{g_{pq}^{10} + g_{pq}^{11} - g_{pq}^{01} - g_{pq}^{00}}{2} \tag{2.40}$$

$$c_{pq} = g_{pq}^{01} + g_{pq}^{10} - g_{pq}^{00} - g_{pq}^{11} \geq 0 \tag{2.41}$$

To simplify the equations it is assumed that the neighbourhood potentials are symmetric ($g'_p = 0, g'_q = 0$) and that the potential is zero if both labels are the same $K_{pq} = 0$. The neighbourhood potential simplifies to

$$U(L|\text{ne}(L)) = \sum_{p \in V, q \in \text{ne}(p)} c_{pq}(1 - L_p)L_q/2 + c_{pq} L_p (1 - L_q)/2 \tag{2.42}$$

The term $(1-L_p)L_q$ is zero if the two labels are the same therefore it is possible to rewrite the equation in the form of a cut capacity function

$$U(L|\text{ne}(L)) = c(S,T) = \sum_{(p,q)\in S\times T} c_{pq} \tag{2.43}$$

The data term can also be written as a cut capacity. The energy cost of a label L_p is

$$U(x_p|L_p) = \begin{cases} c_{sp} & \text{if } L_p = 0 \\ c_{pt} & \text{if } L_p = 1 \end{cases} \tag{2.44}$$

This is equivalent to

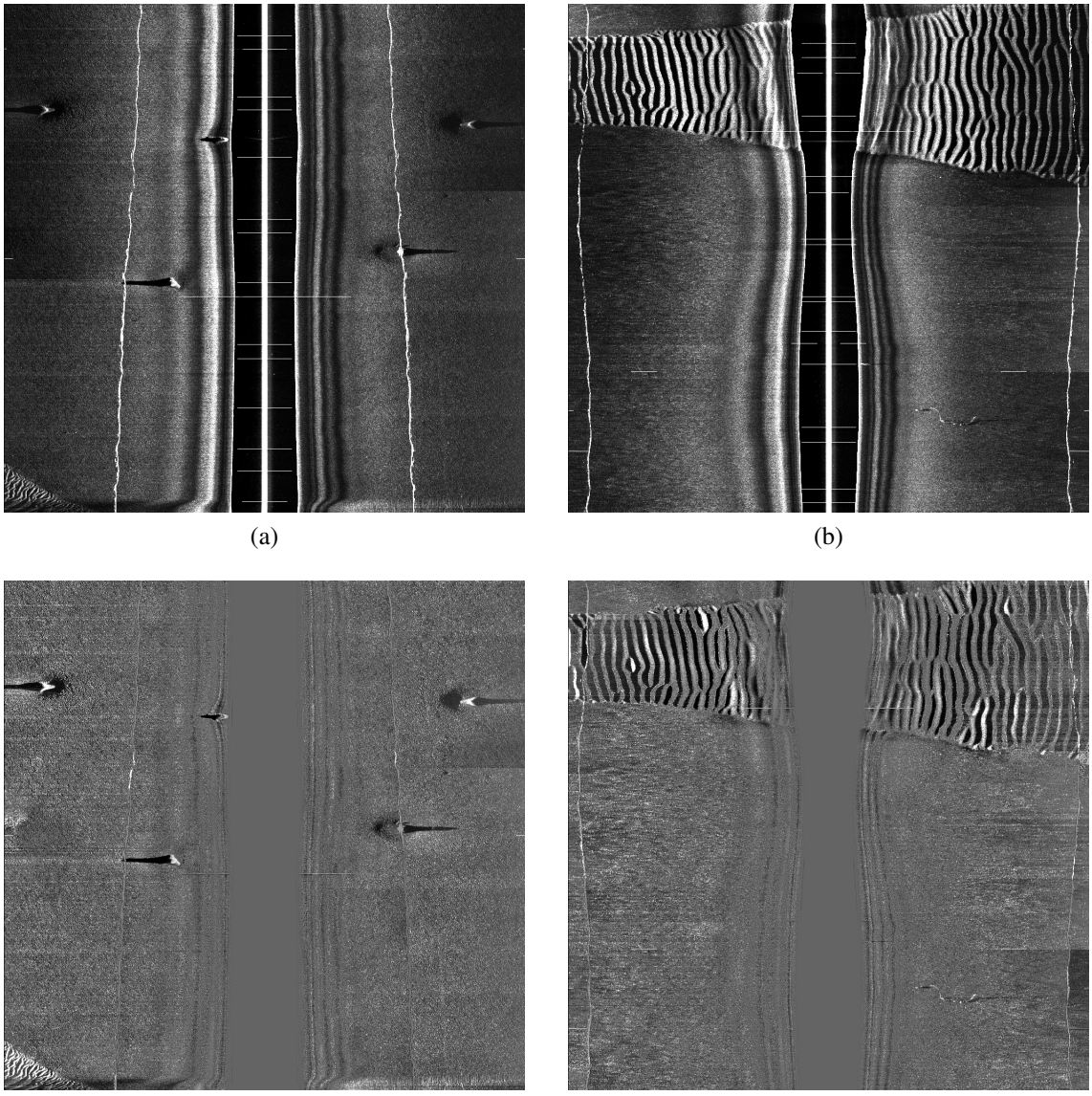
$$U(x_p|L_p) = c_{sp}L_p + c_{pt}(1-L_p) = c(S,T) = \sum_{(p,q)\in S\times T} c_{pq} \tag{2.45}$$

It is therefore possible to pose the energy minimisation problem as a min-cut problem.

$$U(L|x,\text{ne}(L)) = \sum_{p\in V} c_{sp}L_p + c_{pt}(1-L_p) + \sum_{p\in V, q\in\text{ne}(p)} c_{pq}(1-L_p)L_q + c_{pq}L_p(1-L_q) \tag{2.46}$$

From equation 2.46 the structure of the graph is immediately obvious. Each vertex is connected to the source and sink with capacities c_{sp} and c_{pt} respectively. These capacities are given by the energy cost of taking on label $L_p = 0$ and $L_p = 1$ respectively. Each vertex is also connected by an edge to its nearest neighbours. The capacity of these edges is the energy cost for neighbouring nodes taking on different labels. This structure is shown in figure 2.24

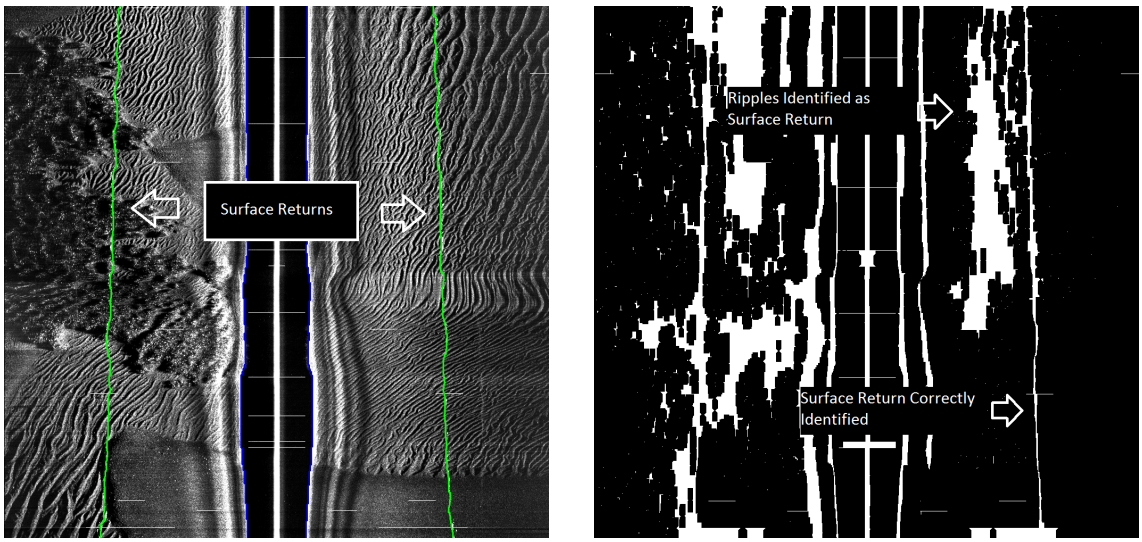
A discussion of the various approaches for finding the max-flow on a graph is beyond the scope of this thesis. In this thesis we have used the approach of Boykov et. al. [40] with the multi-threaded implementation from [41].



(a) (b)

(c) (d)

Figure 2.15: Two examples of images filtered using the SFBF (bottom), original images (top)



(a) Original Image (b) Surface Return Mask

Figure 2.16: Example of the method of Crosby et al. applied to an image with many saturated pixels

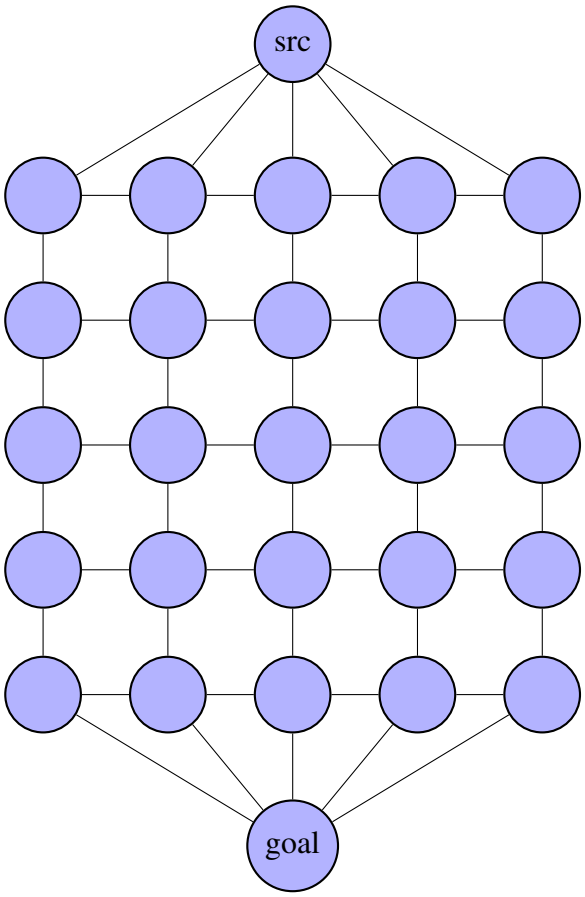
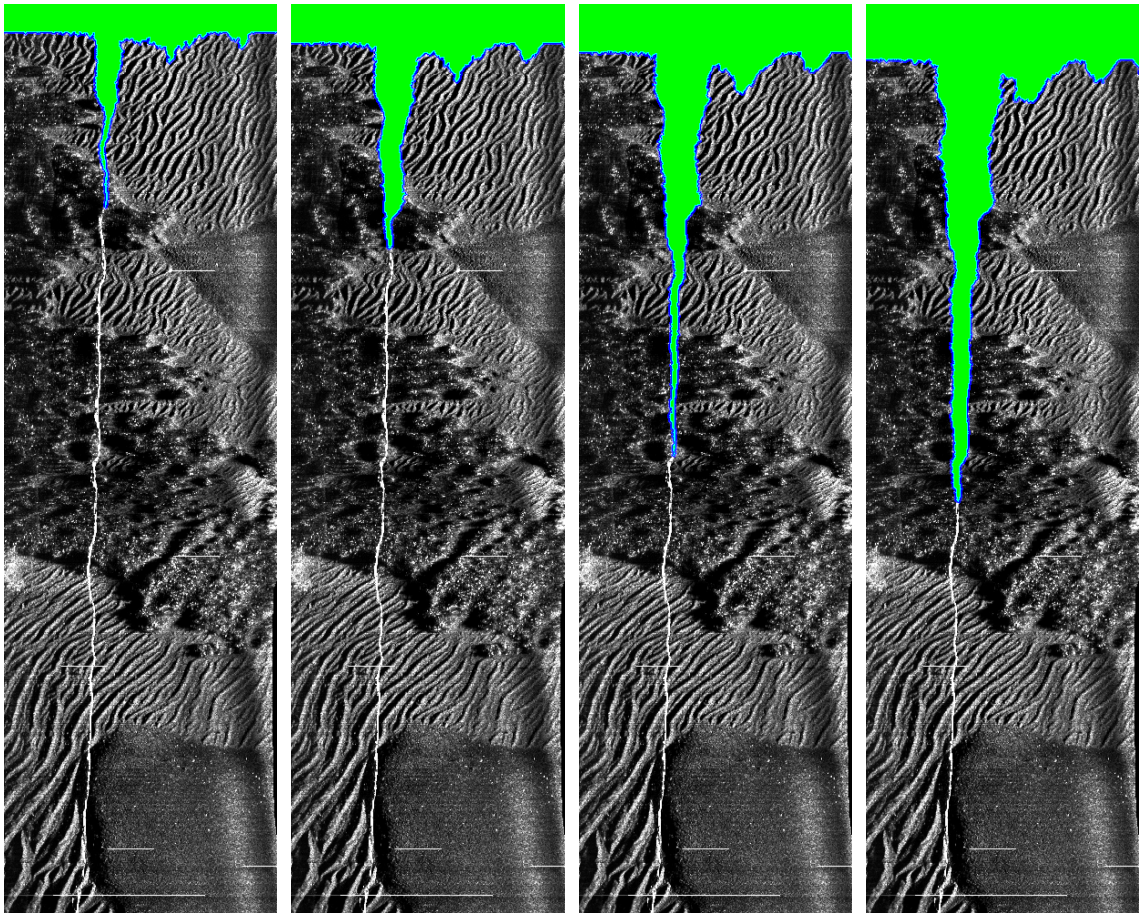
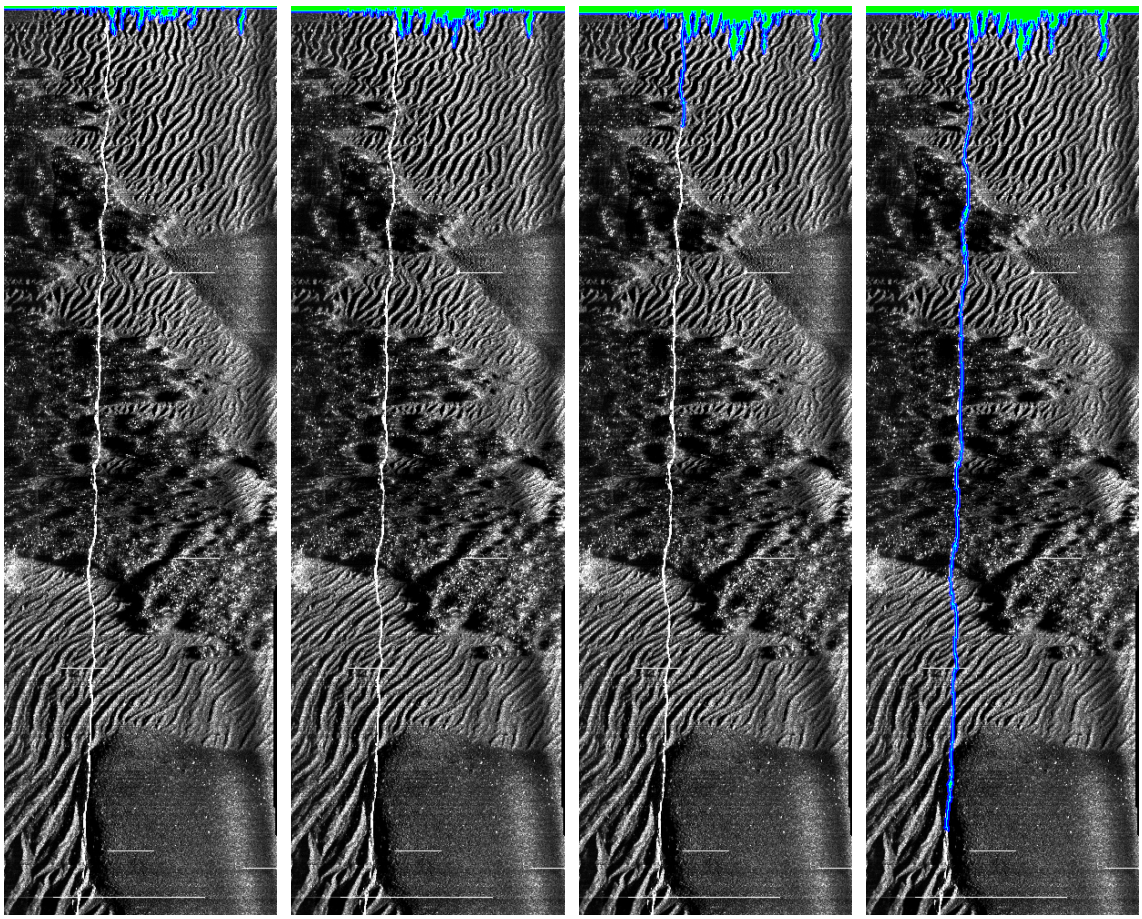


Figure 2.17: An image can be represented as a graph of nodes connected with vertices. In this example there is a node per pixel on a 4-connected neighbourhood. An additional source and goal node are connected to the top and bottom row of the image respectively.



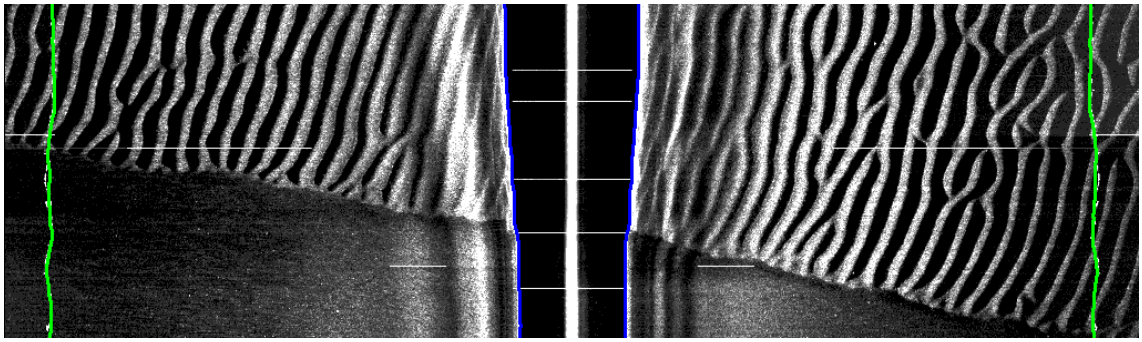
(a) 20000 (b) 30000 (c) 40000 (d) 50000

Figure 2.18: Evolution of the search algorithm with an admissible heuristic. Green pixels are pixels that have been visited, blue pixels are in the open set The number of iterations of the A* algorithm is shown under each image

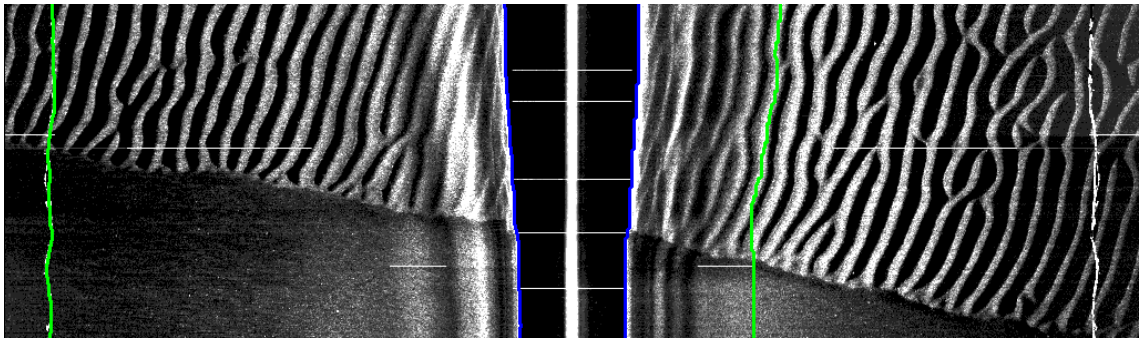


(a) 2000 (b) 4000 (c) 6000 (d) 7000

Figure 2.19: Evolution of the search algorithm with no cost for vertical movement. Green pixels are pixels that have been visited, blue pixels are in the open set The number of iterations of the A* algorithm is shown under each image

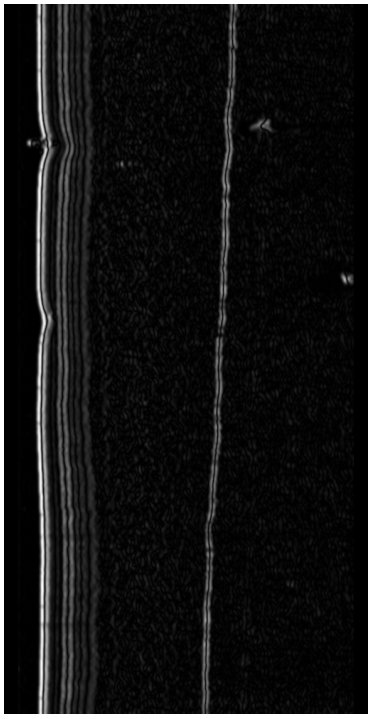


(a) Admissible

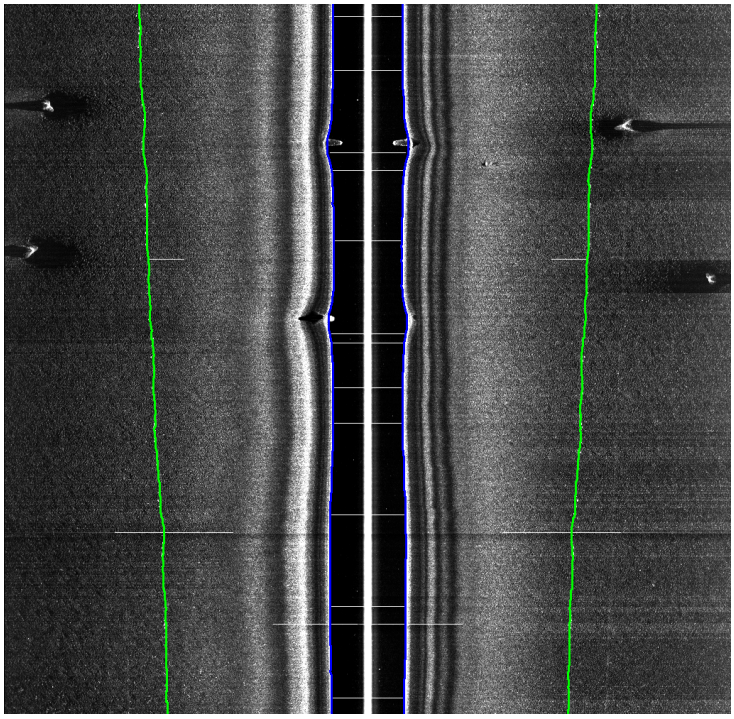


(b) Inadmissible

Figure 2.20: Comparison of the search algorithm with an admissible heuristic and with an inadmissible heuristic in an a difficult image

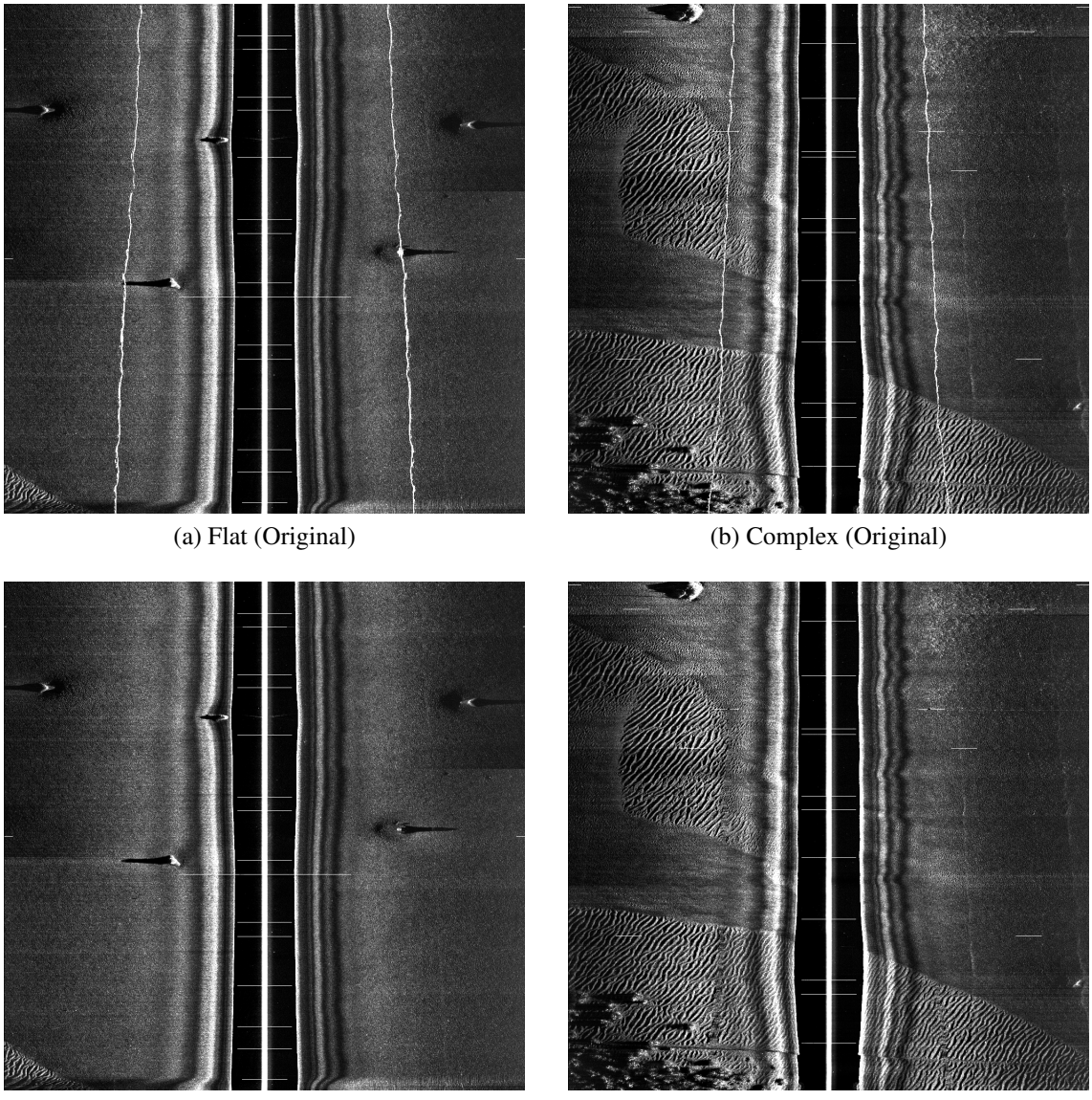


(a) Gradient



(b) Water Column

Figure 2.21: Gradient of the image in the x direction and the position of the water column identified in blue



(a) Flat (Original)

(b) Complex (Original)

(c) Flat

(d) Complex

Figure 2.22: Simple method to remove the surface return from an image.The local statistics are estimated in an 11 pixel wide region about the water column and a randomly generated pixel is used to replace the surface return.

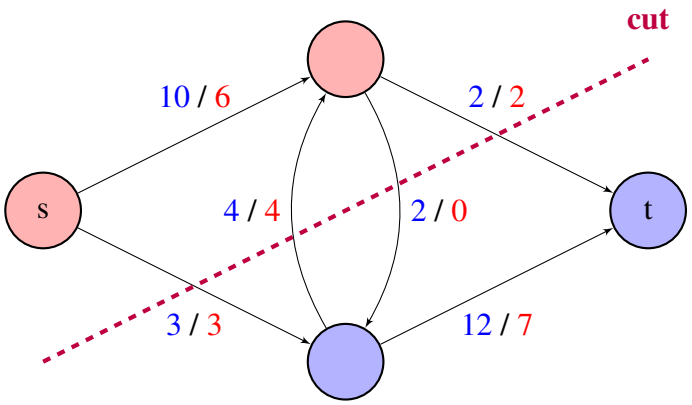


Figure 2.23: The maximum flow from source (s) to sink (t) in a directed graph. Capacity (c) and flow (f) are labelled c / f

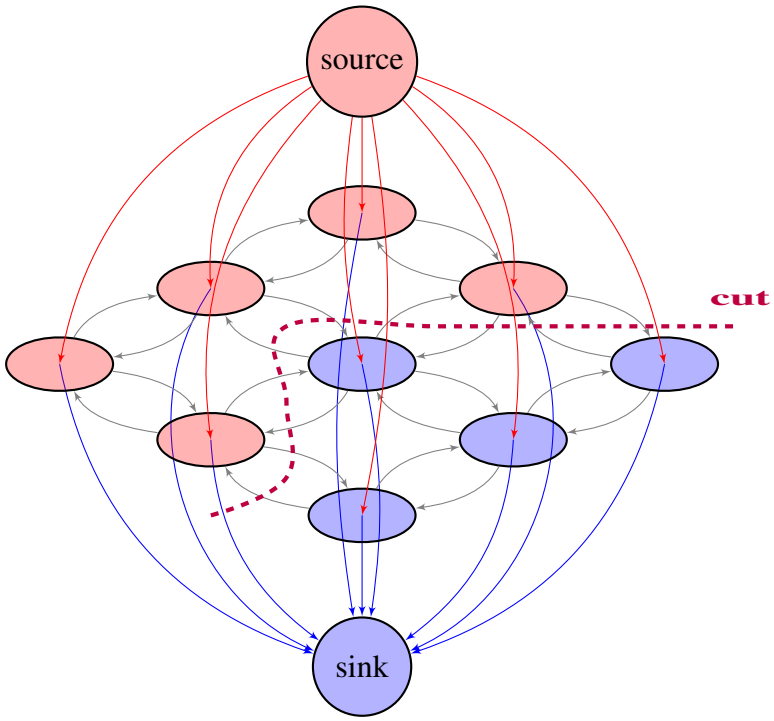


Figure 2.24: Structure of a graph used to solve the min-flow max-cut problem. Nodes are represented by circles and directed edges by arrows.

2.7.2 Sidescan Class Models and Data Terms

In this section we describe the derivation of the data terms from the class likelihood functions. The likelihood functions describing the various sonar classes are well understood and a good overview is given by Mignotte et al.[37]. If we know the likelihood of a pixel belonging to a given class then, according to the Hammersley-Clifford theorem, the data term $U(x_p|L_p)$ is the negative log likelihood.

$$U(x_p|L_p) = -log(\mathcal{L}(x_p|L_p)) \tag{2.47}$$

The background reverberation can be modelled as a Rayleigh distribution, or more accurately a Weibull distribution. The parameters of the distribution are not known a priori and must be estimated from the data. As the parameter estimation is significantly simpler for the Rayleigh distribution we define the background likelihood as:

$$\mathcal{L}(x_p|\sigma_1, L_p = Background) = \frac{x}{\sigma_1^2} e^{-x^2/(2\sigma_1^2)}, \quad x \geq 0 \tag{2.48}$$

where $\sigma_1 > 0$, is the scale parameter of the distribution. A maximum likelihood estimate of σ_1 is given by

$$\sigma_1^2 \approx \frac{1}{2N} \sum_{i=1}^{N} x_i^2 \tag{2.49}$$

The shadow regions contain only reverberation noise and can be modelled as a Gaussian distribution with mean μ_0 and variance σ_0^2. The highlight regions are more difficult to model as the intensity of the highlight is dependent on the material properties and the incidence angle to the object. In the absence of any theoretical underlying statistical distribution we model the highlight as a Gaussian distribution with mean μ_2 and variance σ_2^2.

The class models do not provide sufficient information to segment the image as there is significant overlap in the intensity distributions for each class. Therefore, the neighbourhood term is used to model our prior knowledge of the dependence of the class likelihood functions on the neighbouring pixels.

2.7.3 Neighbourhood Term

The neighbourhood term is defined to model our prior knowledge that neighbouring pixels are likely to belong to the same class. The neighbourhood term penalises neighbouring pixels that have different class labels. The cost is inversely proportional to the first order estimate of the image gradient. Thus, we are more likely to observe different class labels on either side of an edge in the image.

As explained in appendix 2.7.1 the neighbourhood term only contributes to the global

cost when $L_p \neq L_v$. Therefore, the global minimum energy state is a compromise between minimising the number of edges in the label field and maximising the likelihood according to the data term. The neighbourhood term is based on the Potts model as described in [40]

$$U(L_p|L_v) = e^{\frac{|x_p - x_v|}{\sigma_1}} + \alpha \tag{2.50}$$

where α is a regularisation term that allows small edges to be ignored. Normalising by the standard deviation of the background pixels σ_1 allows the segmentation algorithm to operate independently of absolute contrast or intensity.

There is an additional complication introduced to the segmentation by the use of the graph-cuts instead of ICM. Graph-cuts can only be used to find an optimal segmentation of an image if the segmentation is a two class or binary segmentation. Therefore, class models are combined as described in the next section.

2.7.4 Binary Segmentation

Graph-cuts can only be used to find the optimal segmentation for a two class or binary problem. While there exist techniques [42] for multi-class segmentation, they require multiple iterations of the graph cuts algorithm and are covered by several patents. However, for the three class SSS segmentation problem there is little overlap between the distributions of the shadow and highlight pixels. Therefore, we can perform two binary segmentations

First shadow regions are segmented from highlight and background. Next, highlight is segmented from background. The data terms are combined by taking the minimum cost over the two classes. For example the data term for highlight or shadow is defined as

$$U(x_p|\text{Highlight} \cup \text{Background}) = min(U(x_p|\text{Highlight}), U(x_p|\text{Background})) \tag{2.51}$$

In this chapter we have introduced statistical models describing the SSS classes. However, to calculate the data and smoothness terms we first need to estimate the model parameters. These parameters are obtained from an initial guess at the segmentation. The initial segmentation is often poor. However, the model parameters can be updated from the initial segmentation and these parameters then used to further improve the segmentation. This process is known as Iterative Conditional Estimation (ICE) and is described in the next section.

2.7.5 Iterative Conditional Estimation

ICE is an iterative scheme applied to improve the estimation of model parameters from the image segmentation. The input is an initial guess for the class labels. Reed et al. [24] generate this initial guess by assuming that a given percentage of their image is shadow. However, they showed that the segmentation is very sensitive to the initial model generated from this parameter. We therefore generate an initial estimate of shadow and highlight pixels by placing a threshold on the distance of a pixel from the local mean.

The local mean μ_p and standard deviation σ_p are calculated for each pixel in the image. The initial segmentation is then

$$L_p = \begin{cases} \text{Highlight} & \text{if } x_p > mu_p + a\sigma_p \\ \text{Shadow} & \text{if } x_p < mu_p - b\sigma_p \\ \text{Background} & \text{otherwise} \end{cases} \tag{2.52}$$

where a, b are experimentally determined constants. The parameter estimation and segmentation is repeated until convergence or until a fixed number of iterations has been achieved. While there is no theoretical guarantee that the ICE will converge we observe that the majority of the images converge within 10 iterations. Unfortunately, the physical effects that we have already covered in this chapter result in different class distributions at different points in the image. To overcome this problem we introduce an efficient approach to estimating local class parameters.

2.7.6 Local Estimation of Class Models

So far we have considered the estimation of global class parameters over the entire SSS image. We show in section 2.7.7 that this can lead to poor segmentation results at far range. This can be solved through the used of local class model. In this section we describe an efficient method for the local estimation of class models.

The class models require an estimate of the mean, variance and shape parameter. We can obtain these from a local estimate of $E(x)$ and $E(x^2)$. These variables can be estimated using normalised convolution [43]. Given a mask $M(i,j)$ and an image $X(i,j)$ the normalised convolution operation with a filter $G(i,j)$ is defined as

$$F(i,j) = \frac{X(i,j) * G(i,j)}{X(i,j) * M(i,j)} \tag{2.53}$$

where $M(i,j)$ is equal to 1 if the pixel belongs to the class we are performing parameter estimation for and 0 otherwise. Elements of $X(i,j)$ are set to 0 if $M(i,j)$ equals 0. The result of the normalised convolution is a local estimate of $E(x)$ where the local window is defined by the filter $G(i,j)$. In the results presented in the next section we use a Gaussian

window with standard deviation equal to 5% of the image width.

2.7.7 Results and Analysis

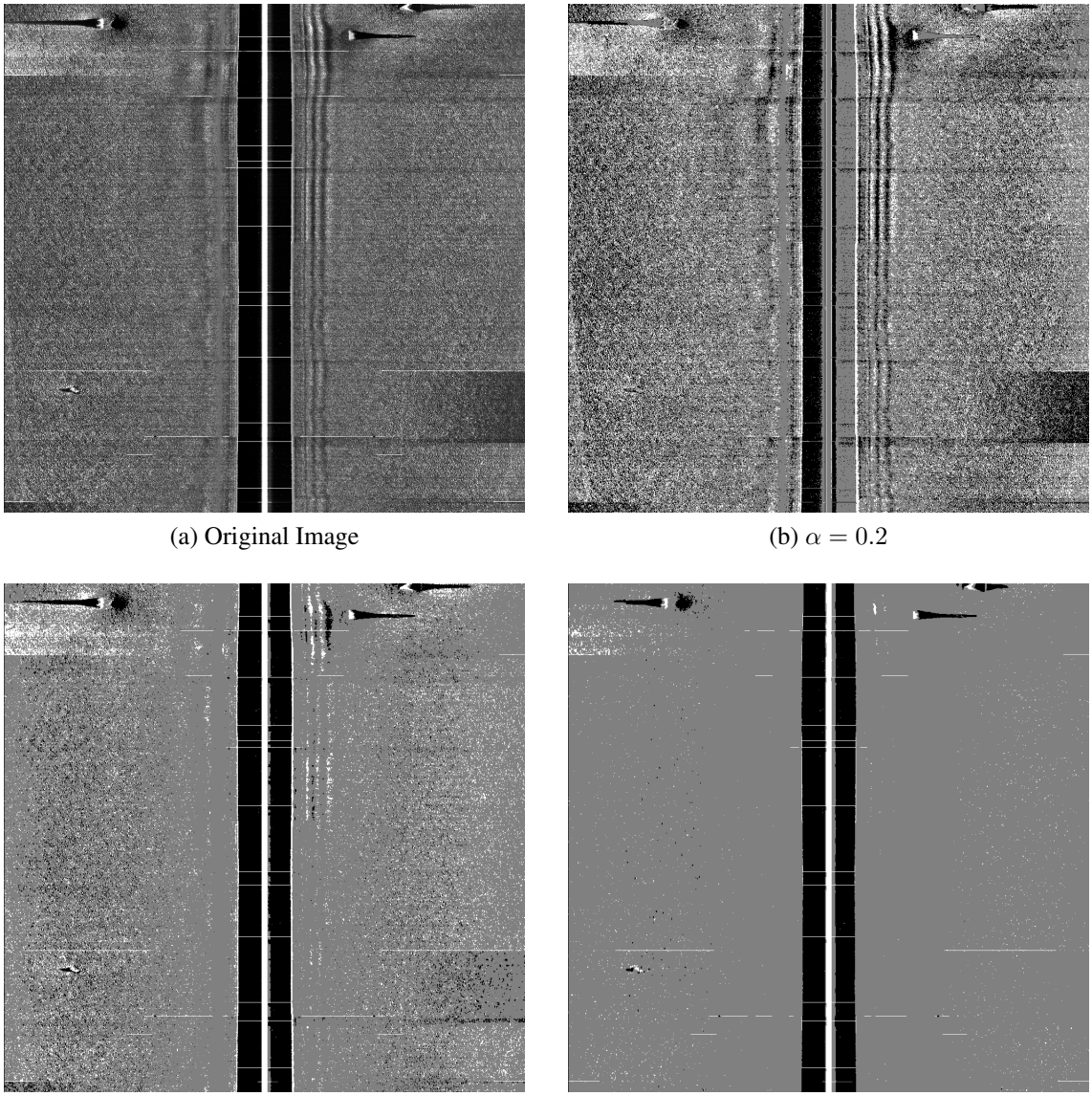
In this section we present the results of the segmentation algorithm. First we investigate the effect of the smoothness parameter α on the neighbourhood term. Next, we show how range dependent effects can result in poor segmentation is some images. We then compare the performance of the segmentation using local and global class estimates. (Note that the water column is not included in the parameter estimation)

Figure 2.25 shows the effect of the smoothness parameter on the segmented image. With $\alpha = 0.2$ the algorithm is not able to recover from the poor initial estimate. As a result the shadow class converges to a distribution describing the dark pixels in the speckle noise. Similarly, the highlight class model describes the brighter pixels in the speckle noise. Increasing the smoothness parameter to $\alpha = 0.4$ favours larger groups of pixels. As a results the speckle noise is classified as background and the highlight and shadow models converge to the correct values. Increasing the smoothness parameter to $\alpha = 1.0$ further smooths the background regions, however some of the shadow from the object in the top right is also lost.

In the rest of this section a smoothness value of $\alpha = 0.4$ is used. The justification for using this value is that it is more important to preserve object shadow than it is to smooth the background regions. Object shadow contains information about the shape and height of the object. If object shadow is not correctly segmented then it can lead to the object being misclassified. In addition it is relatively simple to ignore single pixel regions of highlight and shadow in any object detection method. We now discuss the performance of the algorithm with respect to range.

In figure 2.25 ($\alpha = 1.0$) it is possible to see the problems caused by range dependent effects. There is a slight loss of contrast in the shadow with range, which combined with the high smoothness value results in the shadow being lost at far range. This can be seen more clearly in figure 2.26. The problem is that due to the loss of contrast with range there is significant overlap between the shadow and background classes.

We can recover the shadow at far range by using a local estimate of the class model parameters. A comparison of the local and global approaches is shown in figure 2.27. However, there are a number of draw backs to this approach. The first problem is that the ICE is no longer able to converge to a solution. For example consider a flat sea-floor where a small region has been incorrectly labelled as shadow. In the next iteration the local estimate of the shadow mean will increase and the neighbourhood term will force background pixels close to the incorrectly labelled shadow region to be labelled as shadow. With each iteration the incorrectly labelled region will grow until it encounters an edge which is strong enough to prevent further growth of the region. We can avoid this



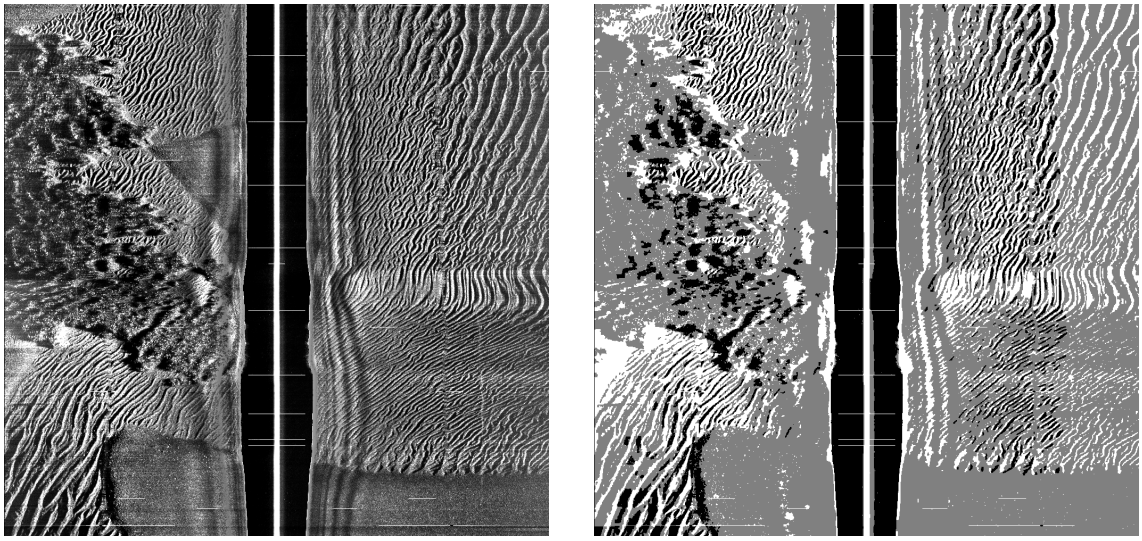
(a) Original Image

(b) $\alpha = 0.2$

(c) $\alpha = 0.4$

(d) $\alpha = 1.0$

Figure 2.25: Comparison of an image segmented with different smoothness values



(a) Original Image (b) Segmented Image

Figure 2.26: Example of an image where the segmentation is dependent on the range

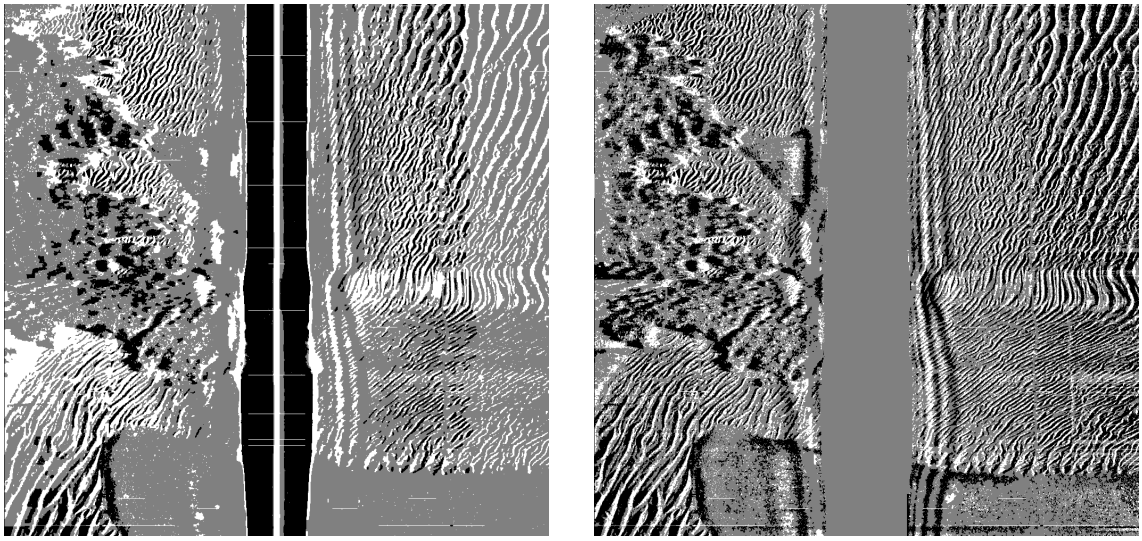
problem by limiting the ICE approach to three iterations. This has the added advantage of reducing the run-time of the algorithm.

We also note that the segmentation of the background regions is less smooth in the local method. This is because once the algorithm has learnt an incorrect shadow model in one region it can not learn the true shadow model from the rest of the image. This can be prevented by increasing the size of the filter used for the local estimation but reduces the robustness of the algorithm with respect to range.

2.7.8 Discussion

In this section we have implemented and compared two SSS segmentation algorithms based on the work of Reed et al. [24]. We have addressed a number of problems identified by the author in the original work. First we have addressed the use of the hierarchical ICM algorithm to solve the MRF. The hierarchical ICM algorithm is a compromise between computational performance and accuracy and has the possibility of converging to local minima. By formulating the problem as two binary segmentations we can use graph-cuts to always converge to the global minima.

The second problem that we have addressed is the range dependence of the segmentation. The change in intensity and contrast with range can result in the algorithm only finding shadow regions at close range. To solve this problem we introduced an efficient approach using normalised convolution to generate local parameter estimates for each point in the image. We showed that this improved the performance of the algorithm with respect to range at the cost of poor segmentation of flat background regions. However, we believe this compromise to be acceptable as the segmentation of real highlight and shadow



(a) Global Segmentation (b) Local Segmentation

Figure 2.27: Comparison of global and local estimates of the class model

regions is very good. The 1 pixel highlight and shadow regions in the background areas can easily be filtered from the final segmentation result if required.

In previous work [24] the author integrates prior knowledge that shadow regions are more likely to occur following a highlight region. While we have not implemented this approach we note that using two binary segmentation stages means that we can use the shadow regions identified in the first stage to improve the segmentation of the highlight regions in the second stage. The work presented in this section form the basis for the object height estimation that we present in chapter 5 on sea-floor characterisation.

2.8 Conclusion

In this chapter we have introduced the theory behind SSS and Synthetic Aperture Sonar (SAS) image and compared a number of algorithms for pre-processing and segmenting sonar data. We introduced the operating principles of sonar and explained how beam steering is applied to form an image. We then introduced the formation of SAS images and demonstrated the differences in object appearance.

Next, we introduced algorithms for correcting range and angle dependent effects in SSS images. We showed that the simple algorithms remove the low frequency range dependent effects but are unable to cope with higher frequency changes due to the beam pattern and the motion of the vehicle. Additionally, they can generate object sized artefacts near the surface return and beam pattern. This is problematic as the artefacts can potentially be identified as target by ATR algorithms in MCM missions.

We then compared two algorithms to remove the higher frequency intensity variations from the image. The first algorithm simplified the problem of estimating angle dependent

intensity variation by re-sampling the image such that columns are aligned by beam angle. Beam angle dependent effects are then estimated and removed with a simple moving average filter. This approach was compared to the SFBF algorithm. The SFBF uses adaptive filters to track paths of similar intensity through the image. While this approach is very effective at eliminating effects such as the beam pattern and the surface return it can distort real features such as ripples. However, this is not a problem for object detection, therefore we apply the SFBF to our images in the following chapter on ATR.

For applications such as sea-floor characterisation distortion of real sea-floor features is undesirable. However, apart from the SFBF, algorithms for correcting intensity variations do not remove the surface return from the image. For this reason we investigated an alternative approach to removing the surface return. We compared a number of existing approaches to this problem and concluded that for object detection the SFBF was a good compromise between computational efficiency and accuracy. Of all the methods implemented the SFBF was the only method that did not introduce additional object sized artefacts into the image. For sea-floor characterisation the most important consideration was retaining the relative intensity and local statistics of different sea-floor types in a mission. We showed that by re-sampling the images by beam angle we could achieve this while also removing the beam angle dependent effects from the image. In future we work aim to implement the full approach of Cervenka and Moustier. [21] and compare this to the methods that we have already implemented.

In the next section we identified a lack or reliable methods for finding the surface return in SSS images. We then implemented a robust approach using the A* search algorithm to find connected paths of saturated pixels through the image. We showed how this algorithm could be extended to visually identify the position of the water column by finding the path of maximum gradient through the SSS image. In our test set of 220 Marine-Sonic images the water column was correctly identified in all the images. The surface return was correctly identified in 199 images, in the other 21 images the surface return intersected the region of the beam pattern that we had excluded from the search region.

In the second half of the chapter we addressed the problem of SSS image segmentation. While there are many approaches to this problem we focussed on using advances in graph based segmentation to improve the performance of SeeByte's existing segmentation algorithm. Using graph-cuts we addressed the problem of the segmentation algorithm converging to a local minima. Furthermore, we introduced an efficient local parameter estimation algorithm that improved the segmentation quality at far range. In chapter 5 we use this algorithm to segment images before calculating the height and density of objects in the image.

The algorithms in this section form the basis for many of the algorithms in later chapters. For example in chapter 4.6 we are able to reconstruct the altitude of the vehicle in a

mission where the altitude data has been corrupted. In the same chapter the SFBF is also applied to images to remove artefacts and normalise the background image intensity. In chapter 5 on sea-floor characterisation the surface return is removed from the images and beam angle dependent effects are corrected.

In the remainder of this thesis we focus on higher-level problems such as ATR and sea-floor characterisation. In the next chapter we depart from pre-processing and begin our discussion of object detection and classification in SSS. However, the low-level algorithms presented in this chapter are an essential component in these algorithms. For example Nelson et al. [44] demonstrate how pre-processing can be used to suppress sand-ripple and dramatically reduce false-alarms from an ATR algorithm. Similarly, Huynh et al. [45] demonstrate how de-noising algorithms can improve the performance of a matched filter detection algorithm. In chapter 5 on sea-floor characterisation we compare the results of the algorithm on images corrupted by surface return and beam patterns. In the worst case the results from 50% of the image can be invalidated by these artefacts.

Chapter 3

Target Classification in Sonar Images

3.1 Introduction

In the first chapter we introduced Mine Counter-Measure (MCM) operations and explained how autonomous systems can be used to increase both the mission tempo and the detection rate. Next we introduced imaging sonar and covered techniques for preprocessing the data. In the remainder of this thesis we focus on the challenges of applying Automatic Target Recognition (ATR) to MCM missions.

In this chapter we review the existing literature on ATR in Sidescan Sonar (SSS) and Synthetic Aperture Sonar (SAS) images. Historically, the majority of the research on sonar ATR algorithms has been focussed on improving the Probability of Detection (PD) and Probability of False Alarm (PFA). As a result, ATR in simple environments can be considered a solved problem. However there are a large number of additional challenges that arise from the deployment of ATR algorithms in practical systems.

An ATR algorithm is only one part of a much larger MCM system. This can impose additional constraints on the algorithm. For example, embedded systems such as Autonomous Underwater Vehicles (AUVs) typically have modest computational power. Therefore, the computational performance of the algorithm and its memory usage are as important as the PD and PFA.

Other challenges relate to the long term practicality of algorithm. For example, it is common for an algorithm to work well in the environment on which it was trained. However, in other environments the algorithm can behave unpredictably. For some algorithms the change in environment can be as simple as a different survey altitude or range setting for a sonar. Unfortunately, this is particularly common for the state of the art, high performance, supervised algorithms [46].

The ease of training for an ATR algorithm is also important. Data collection is expensive. Therefore, it is important that the algorithm does not over-specialise on small data sets. Furthermore, some algorithms are easier than others to adapt to detect new types of

objects. For example, the algorithms described in section 3.4.1 require a 3D model of the target, while the algorithms in section 3.4.2 learn the representation from training data.

The chapter is organised as follows. First we present an overview of ATR in sonar images and applicable techniques from video and image processing. We start in section 3.2 with an introduction to ATR in sonar images. Next in section 3.3 we discuss the process of detecting objects in sonar images. In section 3.4 we introduce algorithms for classifying detected targets in SSS and SAS images.

3.2 Background

In this chapter we describe ATR algorithms for SSS and SAS. Early approaches in imaging sonar stemmed from the field of signal processing. In the mid 1990's the increased resolution of SSS images saw the application of simple image processing techniques such as segmentation [47, 48, 49], and classification based on geometric features [50]. Since then the literature in imaging sonar and video processing has diverged with techniques in sonar focussing on the specific physical [33, 34, 35] and statistical [24, 32, 36, 51] properties of the images.

The explanation for this divergence can be found in both the low resolution and high noise of sonar images. This can degrade the performance of many image processing algorithms. Early systems were typically very low resolution and analysis was performed on the raw sonar data [52, 53, 54, 55, 56, 57, 58, 59, 60, 61]. While some of these techniques remain relevant, this review is focussed on techniques applicable to modern high resolution SSS.

We also review a number of algorithms which are only applicable to SAS images. In SAS images the motion of the sonar platform is used to synthetically increase the extent of the beam forming array. While the actual beam width can be $> 30^\circ$ the synthetic beam width, formed by the coherent addition of multiple pings, is $< 1^\circ$. Platforms such as the NURC MUSCLE vehicle are able to produce images with a resolution of 20 mm. The development of reliable SAS systems has been slow due to the operational complexity of the environment, specifically with respect to accurate motion tracking. However there are currently a number of research groups investigating SAS ATR systems [62]. We include these ATR algorithms in the review as many of them are also applicable to high resolution SSS sonar.

Unlike the video and image processing literature, in which techniques are typically tested against the current state of the art on common data sets, the literature in SSS remains disjoint. Due to the military nature of the data it is difficult to provide a common testing framework. In addition SSS algorithms are heavily influenced by the resolution of the sonar and variability in the object pose and grazing angle. All of these factors make comparison difficult. Therefore, we confine our observations to the relative merits of each

approach rather than attempting a detailed performance comparison. We start by reviewing the detection approaches then in section 3.4 we review the classification of detected objects. The problem of comparing algorithms is also noted by Quidu et al. in [50]. This problem is currently being addressed by several navy research organisations, however due to the sensitive nature of the data we are unable to include the results in this thesis.

3.3 Detection

With a few exceptions [46] SSS ATR algorithms consist of a detection and a classification stage. Detection is typically computationally inexpensive and identifies target-like regions, or regions which are not consistent with the sea-floor. Target-like regions can typically be identified with a far lower computational cost than algorithms which classify the target.

There are several broad areas of research for detection algorithms. Statistical algorithms [24, 32, 47, 48, 49] estimate the likelihood of a pixel belonging to a target using a statistical model of the sonar. Geometric algorithms [33, 34, 35] apply prior knowledge of the simple geometric patterns which can be identified in man made objects. Saliency based detection algorithms [63, 64, 65, 66, 67] are more varied in their approach, however typically they build up a local model of the sea-floor distribution and identify pixels which do not belong to that distribution. We start by reviewing the statistical approaches.

3.3.1 Statistical Segmentation

In statistical segmentation pixels are assigned to one of the three classes we introduced in the previous chapter. The classes are object highlight, shadow and background reverberation. Initial statistical approaches by Calder et al. [47, 48] classified each pixel according to its local neighbourhood using a Bayesian decision framework and a Gibbs sampler. As in chapter 2 they formulate the probability of each pixel x in an image belonging to class of sea-floor as

$$P(x) = \frac{1}{Z} exp\left(-\sum_{c \in C} V_c(x_c)\right) \tag{3.1}$$

where C is the set of maximal cliques of a graph G, Z is a normalising constant and $V_c(x_c)$ are the clique potentials.

A key contribution of their segmentation algorithm is the use of a prior field A to encourage the clustering of object pixels while preventing the growth of these regions becoming too large. An object cluster $(N)_A^O$ and a guard band $(N)_A^B$ are defined as shown in figure 3.1. The prior field in the region a_x is defined as

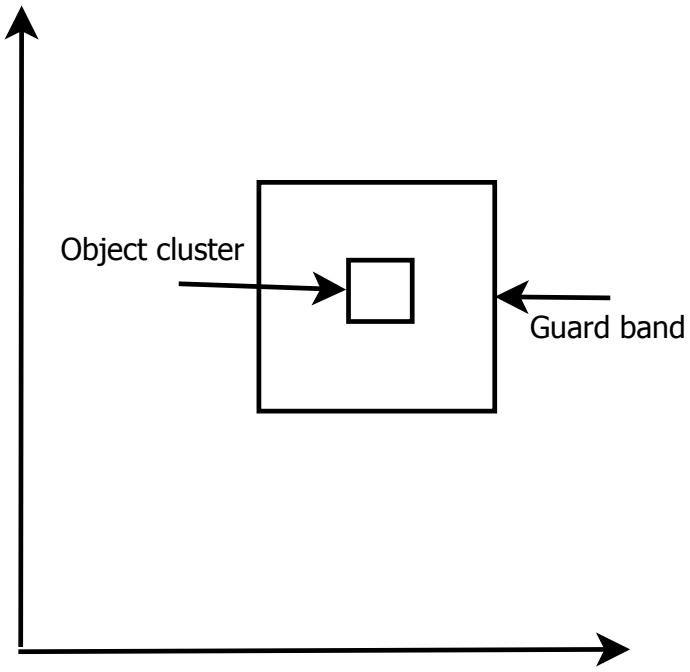


Figure 3.1: Construction of prior field

$$V_c(x) = \begin{cases} -\frac{4}{|(N)_A^O(x)|}(a_x - 1/2)(a_i - 1/2) & i \in (N)_A^O(x) \\ \frac{2}{|(N)_A^B(x)|}(a_i - 1/2) & i \in (N)_A^B(x) \end{cases} \tag{3.2}$$

This prior field assigns an energy cost to large regions of the same class and an energy saving to small regions of the same class. This encourages the clustering of object sized regions.

Next, the clique potentials are defined using a three parameter tanh function. The tanh function approximates the intensity bands for the highlight shadow and background classes. A Metropolis-Hastings sampler is used to to minimise the global energy cost specified by the prior and clique potentials. The sampling is repeated until convergence at a global minima is achieved. On flat sea-floor the algorithm correctly segments 92% of objects with 0.5 false alarms per image (the image size and range is not specified). At a detection rate of 87%, 0.19 false alarms per image are identified. However, due to the use of the Metropolis-Hastings algorithm the calculations are computationally intensive.

A more computationally efficient approach was introduced for active sonar in [49] and adapted for SSS by Reed [24]. In this approach Iterated Conditional Modes (ICM) (described in chapter 2) is used to minimise the energy cost. However, this algorithm converges to local minima, rather than a global minima. Consequently, efficiency is obtained at the cost of the accuracy of the segmentation. Reed also replaces the three parameter tanh function for estimating clique potentials with more accurate models of the sea-floor statistics. The shadow and highlight regions are described as Gaussian distributions while the sea-floor is described by a Rayleigh distribution. Reed also introduces a second phase

to the algorithm to find boundary regions for small objects in the image. This approach is known as the cooperating statistical snake model.

A statistical snake [24], also known as an active contour model, is an energy based approach for finding boundary regions or contours in an image. A snake is defined by a set of n points $\mathbf{v}_i$ where $i = 0 \ldots n-1$. The snake has an internal energy term $E_{internal}$, and an external energy term $E_{external}$. The external energy is comprised of energy costs due to the image E_{image} and constraints E_{con}. These constraints are edge-based while the internal costs are based on values of the pixels enclosed by the snake.

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))\, ds = \int_0^1 (E_{internal}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))) \quad (3.3)$$

Reed et al. identified that the use of a single snake to highlight the shadow and highlight regions resulted in a sub-optimal segmentation. Their approach combines two cooperating snakes to segment both the highlight and shadow. This approach allows the width of the highlight to bound the width of the shadow and vice versa. Reed shows that this algorithm is particularly effective in ripples where the object shadow is indistinguishable from the shadow due to the ripples.

It is not possible to compare the quality of the segmentation algorithms of Calder and Reed directly due to the difference in metrics. However, in [73] the segmentation algorithm is used as an input to a geometry based classification algorithm. The results show that the segmentation algorithm correctly segments 94% of the objects in the test data. Unlike the data shown in [48] the test data contains difficult regions with rocks and ripples. However, as no information is available on the number of false positives in [73] we cannot make any conclusions about the relative performance of the approaches.

The difference between global and local energy minimisation can be seen by comparing the energy functions from the Markov Random Field (MRF) approach in [24] and an active contour approach in [68]. Before introducing these functions, a basic understanding of the Potts interaction model is required. The Potts model is a model of interacting spins on a crystal lattice, it is interesting not because of it accuracy but because it is exactly solvable and fast computational methods exist for solving the system. The Potts Hamiltonian consists of an interaction term between neighbouring particles and a global term from an external magnetic field. The analogue in computer vision is a smoothness term $V_{p,q}$ which forces neighbouring pixels to take on the same class and a data term $D_p(.)$ which forces a pixel towards it's most likely class. The typical energy function E is therefore

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{(p,q) \in N} V_{p,q}(L_p, L, q) \tag{3.4}$$

where $L = L_p | p \in P$ is a labelling of an area P.

In [24] $D_p(.) = -log[Bel(L_p)]$ and $V_{p,q} = B[1 - \delta(L_p, L_q)]$, where $Bel(L_p)$ is the Dempster-Shafer belief for label L_p and B is a weighting term for the smoothness term over the data term (Note that the treatment of unlabelled pixels in [24] has been ignored).

The active contour model used by Lianantonakis et. al. [68] is a simplified version of the Mumford-Shah functional. Consider the textural feature space of an image as a vector field I. For two class segmentation $c1, c2$ bounded by a curve C the average of I inside C ($c1$) is defined as $m_{in}(C)$ and the average of I outside C ($c2$) is defined as $m_{out}(C)$. The simplified Mumford-Shah energy functional is then defined as:

$$\begin{aligned} E(c1, c2, C) =& \lambda \int_{int(C)} |I(x,y) - m_{in}(C)|^2 dxdy \\ &+ \lambda \int_{ext(C)} |I(x,y) - m_{out}(C)|^2 dxdy + \mu Length(C) \end{aligned} \tag{3.5}$$

The first two terms minimise the sum of squares between the image model and the textural feature space within the boundaries. This is equivalent to the data term in equation 5.37 with a euclidean distance classifier. The second term minimises the length of the boundary. This terms does not have an analogue in the Potts energy model as it is calculate from local information that can't be modelled by a MRF. Karoui et. al. [69] apply a similar active contour model to segment sonar images, however they extend the scheme to multiple classes and use a log likelihood membership function based on the Kullback-leibler divergence.

From the two energy equations, the global Potts model and the local Mumford-Shah functional, it can be seen that there are several trade-offs between the two methods. The MRF model does not need to be initialised close to the true solution for it to converge and is computationally more efficient. The active contour model can include priors on the shape of the boundary but can converge to local minima.

Maussang et al. [15, 70] investigate a different statistical approach to segmenting SAS images based on the relationship between the mean and the standard deviation of the Rayleigh noise found in sonar images. Modelling the sea floor response with a Weibull distribution, of which the Rayleigh distribution is a special case, the Mean - Standard Deviation (MSD) are related by a multiplicative constant (figure 3.2). In highlight and shadow regions the statistics are less constrained. In [15, 70] a dual threshold is applied to the MSD plane such that high MSD regions are labelled as highlight and low MSD regions as shadow . The threshold in standard deviation is optimised by finding minima

in the rate of change of entropy of the thresholded image. The approach is well justified theoretically however only a single test image is presented. It is possible that the entropy minimisation technique may be less effective in more complex images where deviations from the Weibull distribution will be observed at small grazing angles. While the technique is presented for SAS the statistical laws are well defined for SSS and it is likely that the approach would be equally valid in high resolution SSS.

Once an image has been segmented, object detection is relatively simple. For example in chapter 5 we detect objects by matching highlight and shadow pixels to calculate the height of the objects.

3.3.2 Salient Region Detection

Salient region detectors detect objects by modelling the difference between a patch of sea-floor and its surroundings. In [63, 64] the authors look for anomalies in otherwise homogeneously textured regions. Background reverberation in SSS can be modelled as a fractional Brownian process[71]. A fractal Brownian signal S(x) can be characterised as:

$$\langle (S(x) - X(x + \Delta))^2 \rangle \propto \Delta^{2H} \tag{3.6}$$

where $\langle .. \rangle$ represents the expectation operator, Δ represents a displacement and H is related to the fractal dimension D and Euclidean dimension E as:

$$D = E + 1 - H \tag{3.7}$$

where in a 2-Dimensional image $E = 2$.

For a truly fractional Brownian process, plotting the log of the left hand side of the equation vs displacement yields a straight line graph with gradient equal to the fractal dimension. The authors describe a methods for detecting objects by identifying regions where the fractional dimension differs from that expected for sea-floor. However, while the approach is interesting, no results are provided.

In [65] the Shannon entropy is used to detect salient regions. For a point x and scale s with local gray level amplitude $d \in D$ and probability density function $p(d/s, x)$ the Shannon entropy is:

$$\mathbb{H}_D = -\sum_{d \in D} p(d/s, x) log_2 p(d/s, x) \tag{3.8}$$

If the region consists of only one gray level then it will have a Shannon entropy of zero. Therefore, there will be a peak in the entropy in a region surrounding an object at a scale which characterises the object. A threshold on the maximum entropy with respect to scale provides a robust detection and scale selection of a salient object. The results for

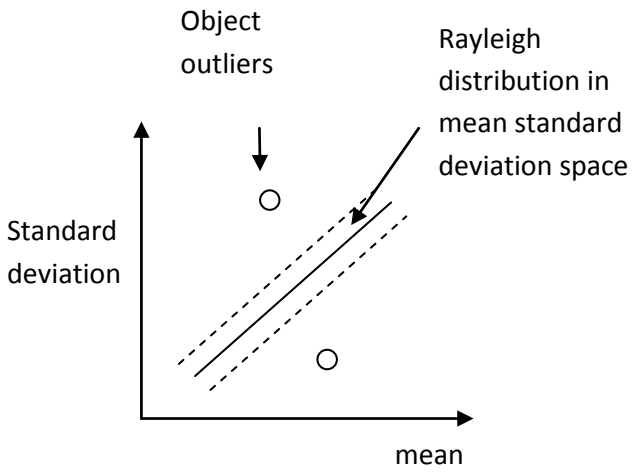


Figure 3.2: Detection of object regions in the mean standard-deviation space

this method are only presented for a single small image, therefore they are not directly comparable to the other methods.

In [66, 67] coherent correlation analysis is used to extract regions which are highly correlated. This method also exploits the fact that man-made objects form ordered regions in sonar images. In this technique the image is first split into overlapping target sized regions and the correlation between columns in the region is determined. Target regions will typically display a higher degree of correlation and these regions can be detected by a threshold on the degree of correlation. In [66] the algorithm is tested over a number of terrain types. For an object detection rate of 99% the algorithm achieves a false alarm rate of 3%, 5% and 6% for easy, medium and hard terrain respectively. This represents between 200 and 300 detected regions per image.

Finally, in [72] Higher Order Statistics (HOS), skewness and the kurtosis, are used in conjunction with the MSD. The method is applied to detect target-like pixels in SAS imagery. In chapter 2 we explained how the mean and standard deviation of Rayleigh distributed noise is related by a single multiplicative constant . Small deterministic regions, situated in a stochastic background, will lead to a skewed probability distribution (high skewness) and a peaked probability distribution (high kurtosis). This can be localised using matched filtering with prior knowledge of the target and window size. In the results each pixel is counted as a false alarm therefore the false alarm rate is much higher than other algorithms which report false alarms after pixels have been grouped into object regions. At a detection probability of 90% they report a false alarm rate of between 5% and 10%.

3.3.3 Geometric Detection

Geometric algorithms detect blobs of highlight and shadow pixels and reject those which are not similar to the target size. The detection algorithm in [34] could equally be considered as a salient region detector or as an intensity based detector. The local seabed intensity envelope is estimated by applying morphological opening and closing operators to remove target sized objects. The estimated intensity is used to derive robust thresholds for highlight and shadow regions. Only qualitative results are provided.

Finally, in [35] pixels are segmented by intensity in to highlight,background and shadow and the a threshold on the number of each pixel type in an object/shadow mask is used to determine the probability that the mask contains a detection. At a PD of 92% they report 1.5 false alarms per $900m^2$. The difficulty of the terrain is not described.

3.3.4 Summary

Comparison of detection algorithms in SSS and SAS data is complicated by the large number of sonar designs and sea-floor types. Even if two papers use the same sonar it is not valid to directly compare the performance of two detection algorithms as the data set must be controlled for, among other factors, the sea-floor type, target appearance, burial depth and quantity of marine growth. The problem of comparing algorithms is further complicated by the way in which the performance is reported. For example some authors reports results for an algorithm which is a combination of a segmentation algorithms and a classification algorithm, other authors report the quality of the segmentation algorithm but use different metrics. Some papers report false alarms after pixels have been grouped into target sized regions, other report false alarms directly for each pixel. Without access to the source data and algorithm implementation it is impossible to compare these results.

In this section we have grouped the algorithms by similarity of approach and analytical method. Therefore, while we cannot identify the ”best” algorithm we can identify approaches which have been successful. Statistical segmentation detection algorithms apply variations on the MRF method of segmentation to label pixels as highlight, shadow or background. This simplifies later analysis as object features such as size and height can be simply extracted from the image. However, the best PD, reported by Reed [73], is 94%. These targets are lost before the classification stage and cannot be recovered.

Higher PDs are reported using the salient region detection algorithms. These algorithms exploit both the properties of sidescan sonar physics and the ordered appearance of man-made objects to detect targets. These algorithms can be tuned to achieve PDs of 100%. While a large number of false alarms (up to 200 per image) are detected it may be possible to eliminate these false alarms with a classification algorithm and therefore achieve a higher PD than the more selective statistical segmentation algorithms.

The ”best” detection algorithm is dependent on the information required by the clas-

sification algorithm. For example, if the classification algorithm requires contours to be extracted from the image then it make sense to use a segmentation approach for detection. In general the PD of the algorithm is more important than the PFA, as once a location has been discarded by the detection algorithm it can not be retrieved in classification. In this respect all of the algorithms reviewed perform well on flat sea floor. For uniformly textured areas such as ripples the saliency based approaches give good results, however their performance drops in cluttered regions where the statistical and intensity based segmentation give the best results.

If the classification algorithm is computationally inexpensive then simple algorithms based on geometry or an adaptive threshold are sufficient as the classification algorithm can rapidly reject false alarms. These algorithms typically have both a high PD and PFA. For more computationally expensive algorithms a selective approach such as [65] may be applicable. However, in this case targets can be lost at the detection stage. As we see later in the chapter the detection process can be replaced by an attention focussing algorithm such as the Haar Cascade [46]. In this approach a multi-stage classifier is used to reject simple, then more complicated, background regions. The computational performance and accuracy of this approach is such that in most cases it can replace a traditional detection stage.

Detection algorithms are unsupervised, therefore the challenges associated with re-training are not applicable. Similarly, the majority of the algorithms are robust with respect to the environmental and operational parameters of the MCM mission. Although, the statistical segmentation approaches can be degraded by range dependent intensity changes. These issues are more apparent when the detected objects are classified. This is examined in the next section.

3.4 Classification

In this section we review model-based and supervised classification algorithms. Model-based algorithm extract information about the target, such as its shape or exterior contour, and compare this to a model of the target. Supervised algorithms extract similar information but learn the difference between objects and background from a training set. We start by looking at a number of model-based algorithms.

3.4.1 Model-Based Algorithms

Model-based algorithms [24, 32, 35, 36, 74, 75] rely entirely on a priori information about the object to be detected. Therefore, only a model of the target is required. As a result the algorithms are useful for the detection of new object types for which there is little or no training information. Typically the classifiers use simple decision making algorithms and

the principles of operation are transparent. In the SSS literature these methods are based exclusively on template matching. However, the implementations are varied with some algorithms matching previously segmented contours [24, 32, 36, 74] and other operating directly on the pixels using a sliding window approach [75]. Here we consider two such approaches.

The contour matching algorithm of Reed. et al. [24, 36] uses Dempster Shafer decision theory to match simulated and observed shadow shapes via the Hausdorff distance. Dempster-Shafer belief theory is applied to combine multiple, single feature, Gaussian classifiers. Instead of directly considering probability, evidence supporting a hypothesis is assigned a normalised mass, such that the sum of all masses is equal to 1. In Dempster-Shafer theory the mass functions are combined to define concepts of belief and plausibility. The belief in a hypothesis is the sum of all masses in the set enclosed by the hypothesis while the plausibility is the sum of all the masses in the set not enclosed by the hypothesis. In other words the belief is the sum of the evidence supporting the hypothesis while the plausibility is the extent to which the evidence contradicting the hypothesis leaves room for belief in the hypothesis. Therefore, the belief over all geometric features can be combined to produce a classification. As a result a simple rule-based classified can be produced from prior knowledge of the target object dimensions.

The algorithm [24, 36] performs well at separating known object classes, however the algorithm is less efficient when classifying against clutter. As a result the algorithm produces a large number of false alarms in these regions. Additionally, the segmentation is more likely to fail in cluttered and rippled regions which can cause the PD to drop. The classifier accurately classifies a small set of similar object classes, $> 90\%$ of objects are correctly classified, however the algorithm is less efficient when classifying against clutter.

In [74] object highlight and shadow regions are segmented using region growing. The segmentation is course and the highlight and shadow boundaries extracted from the segmentation are corrupted by noise. Templates are matched to object highlight and shadow regions using non-linear correlation or correntropy. In correntropy based matching the cross correlation between two random variables is modified by a kernel function, which is chosen such that the effect of outliers is minimised. As a result good detection results can be achieved from a poor segmentation. Results are given for distinguishing between four object classes, cube, cylinder, cone and regular cuboid. However, no results are provided for distinguish between mine-shapes and non-mine clutter shapes.

The two algorithms are conceptually similar, however in [24, 36] the focus is on accurate segmentation while in [74] the focus is on accurate matching of poorly segmented data. The main benefit of these algorithms with respect to the challenges presented in the introduction is that there is no requirement for training data. Additionally, geometric distortions of the target due to range and resolution are handled by projecting a 3D model

into the 2D image with known range and rotation. This makes the algorithms very robust to environmental changes. However, the algorithms are only applicable to flat sea-floor as the PD and PFA are both poor in cluttered and rippled areas. In the next section we discuss how performance in these regions can be improved by using supervised classification.

3.4.2 Supervised Classification

Supervised classification algorithms classify targets based on a vector of features extracted from a local image patch. Unlike unsupervised classification algorithms, the classifier learns the distribution of both target and background appearance[6]. Conceptually the classifier can learn features to reject background objects as well as learning the appearance of target objects. This provides an advantage for classification in similar environments to the training environment. However, there is no guarantee that a classifier trained on one sea-floor type will generalise well to another sea floor.

This disadvantage can be negated, in part, by training the classifier on many different background environments, with targets rendered by augmented reality simulators [25, 32, 76]. However, care must be taken that the augmented reality images demonstrate the same variability as real world objects. This is especially true in approaches which rely heavily on machine learning for the selection of features. Here, it is possible that the classifier will learn artefacts of the simulation process.

The performance of supervised classification algorithms is heavily dependent on the selection of features and, to a lesser degree, the choice of decision making algorithm. The choice of features can broadly be divided into simple features [77], geometric features[31, 50], filter responses [5, 76] and spectral responses [78].

Simple features cover a number of techniques [31, 77] which are computationally efficient and conceptually simple. In [31] Mine Like Object (MLO) are classified using super ellipse fitting. Super ellipses can describe a range of simple shapes from circles to rhomboids on the basis of three parameters defining the length and width of the shape and sharpness of its corners. The super-ellipses are then fitted to the shadow contours. While the classification performance is average the method is not computationally intensive and could provide a useful set of features for a more complex classification scheme. This is demonstrated by results from [31] where the super-elipse is only correct for 70.8% of the targets.

Methods based on simple feature are transparent in their operation and typically have low computational requirements. However, they discard much of the information contained within the target region. The information retained is sufficient for classification in easy environments with simple object classes. However, it is not sufficient to differentiate between complex cluttered region. Neither are the methods extensible to more complicated objects which can not be described by a simple geometric shape.

Geometric features were introduced previously for model-based classification. Typically geometric features are extracted from a segmented image. For example, the length of the shadow and the area of the highlight are all salient features. In this section we focus on the work of Quidu et al. [50] due to the consideration of the invariance of the features with respect to grazing angle and the aspect of the object. Normalisation with respect to grazing angle is accomplished using a simple geometric transformation of the segmented image. All objects in the transformed image appear to have a grazing angle of 45 degrees. Aspect invariance is provided by using moment invariants to describe regions in a rotation invariant manner. However, like the unsupervised approaches, the performance is highly dependent on the quality of the segmentation.

Rotation invariant features are also derived in [78] by considering phase relationships between the 2D Fourier components of image patches. In the method Chandran et al. [78] spectral responses are calculated from the Fourier transform of the image patch. Objects are indicated by non-random phase relationships between the Fourier components.

Filter response algorithms remove the need for segmentation by calculating features directly from the image. The feature response is calculated by convolving a 2D filter with the image. While the information from any single filter is often limited, multiple filter can be combined to create a very strong classifier. However, the plurality of features in filter based methods increases the likelihood that the classifier will over fit the training set. In both [5, 76] augmented reality data is used to mitigate this risk.

In [76] central features, derived from differentials of Gaussians, are used as an input to a Bayesian classifier. The central features are derived from differentials of the Gaussian function and examples of the used features are shown in figure 3.3. On real data the central features approach detected 81% of real targets with only 0.001% chance of false alarm per snippet. However, the authors note that all of the real data is from flat sea-floor. On simulated rippled and cluttered sea-floor the probability of a false alarm is 0.016%. While the classification performance is good on complex backgrounds the calculation of the 16 central features over the entire image is computationally intensive. This is addressed in [5] where Haar like features are calculated from an integral image representation of the sonar image.

The Haar Cascade, originally developed for face detection [79], also introduces the attention focussing technique that we referred to earlier. A cascade of increasingly more complex classifiers are used to process the image. The first classifier only has to reject 50% of the background pixels while retaining 99% of the target pixels. This can often be achieved by evaluating a single Haar filter response. As more background is rejected the remaining background becomes more difficult to separate from the targets. Therefore, each stage of the cascade takes progressively longer to evaluate. However, as stage n only has to process $\frac{100}{2/^n}\%$ of the image the process is very fast. In the next chapter we examine the Haar Cascade in more detail. Specifically we examine the approach to see if we can

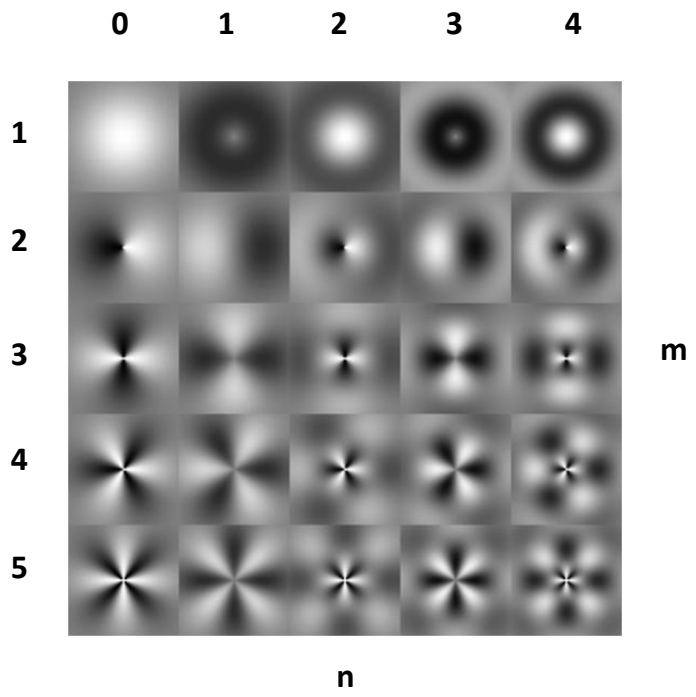


Figure 3.3: Example of the central features for different values of m and n

address the problem of filter based algorithms over-specialising on a single environment or sonar type.

3.5 Synthetic Aperture Sonar

There are a number of approaches that can only be applied to SAS data due to the fact that the SAS image is an analytic signal. In one such method [80] object and shadow regions are segmented using the Hilbert Transform, a technique closely related to the Fourier transform. The Hilbert transform $\tilde{f}(t)$ of a signal $f(t)$ is a complex valued function with the real part $f(t)$ given by the original signal and the complex part $\hat{f}(t)$ given by

$$\hat{f}(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{f(\tau)}{t-\tau} d\tau \tag{3.9}$$

where $P\int_{-\infty}^{\infty} f(x)$ is the Cauchy principal value of the integral. The modulus of $\tilde{f}(t)$ gives the envelope of an oscillating function therefore a threshold on this value gives a reliable segmentation on a highlight region corrupted by noise. In the shadow region the real part of the function tends to zero, therefore the phase will tent to $\pi/2$, and the shadow can be reliably segmented. The Hilbert transform provides a course segmentation which is equally sensitive to objects of interest, topographical features and clutter.

Hilbert transform segmentation is used by the authors to extract geometric features from simulated objects rendered at different aspects, grazing angles and burial depths

[80]. Target regions are separated from ripples and other topographical features by applying an iterative curve fitting technique to each detection. A curve is fitted, to the maximum pixel in each cross track-line, in the along-track direction, the maximum pixel is removed from the evaluation and the process repeated. A threshold on the standard deviation of the set of curves, for each line in the across-track direction, is used to remove background regions where the position of the curves is unrestrained. The longest curve satisfying the standard deviation criteria is used to represent each object.

A total of 24 features are extracted from the highlight, low backscatter and shadow regions. The curve extracted from the standard deviation criteria is reduced using Ramer's algorithm and 4 corners extracted, these corners are used to derive a set of 12 geometric features. A similar algorithm is used to extract 9 shadow features and a further 3 low backscatter features. The classification is performed with a set of binary decision trees, which are used to distinguish between a single target type and all the other targets. Receiver Operator Characteristics (ROC) curves are presented for each of the classifiers and classification percentages of up to 95% with 0.01% of clutter object classified as targets. It should be noted that the analysis is performed entirely on well defined targets situated on a flat, uncluttered sea floor and the object is only detected against a background of other well defined objects classes. Additionally the data used for training and testing is simulated and therefore the variability encountered in real missions will not be reproduced.

3.6 SSS Methods Applied To SAS Data

In practice all of the methods used for Computer Aided Detection and Classification (CADAC) in SSS images can, with minor modifications, be applied to SAS images. They must, however, account for the increased speckle noise in SAS images. The reverse however is not always true, for instance the HOS presented in [72] would be unstable given the lower resolution of SSS images.

The statistical snake technique of Reed et. al. [36] is applied to SAS images [32, 46] with good initial results, however the authors were unable to apply the final shape matching stage due to a lack of specific information concerning the resolution of the image and the slant range of the target. The focus of [46] is on methods that require little or no real training data as such methods have a tendency to over-fit the, generally small, set of real training samples. In addition to the statistical snake technique, which requires no training data, the Haar features method reviewed in section 3.4 is trained and tested on augmented reality data and additionally on a real data set. The method achieves a PD of $> 95\%$ on symmetric targets on a simple background. The performance is reduced for detection of non-symmetric targets as the algorithm is not rotationally invariant and must effectively learn each orientation as a separate object.

3.7 Conclusion

In this chapter we introduced a number of model-based and supervised classification methods for SSS. These results are summarised in table 3.1. In general the supervised algorithms achieve a higher PD than model-based algorithms. However, there are significant practical advantages to the model-base algorithms. All that is required to classify a new target is a 3D model of the target. This is particularly important in situation where training data is limited, such as when a new target type needs to be detected. The model-based algorithms also have significant advantages with respect to robustness to the operational parameters of the mission. Factors such as the speed and altitude of the vehicle and the range of the sonar are all modelled when the image is compared to the model. Given all of these advantages it is perhaps surprising that, to our knowledge, the majority of operational systems use supervised approaches.

Table 3.1: Summary of results for papers reviewed in this section

Author	Algorithm Type	Method	Data Type	Terrain Type	Results Type	Max PD	PFA at 90% PD
Grasso and Spina [34]	Object density detection	Morphology	Real SSS				Results not comparable
Linnet et al. [64]	N/A	Fractal texture analysis					Discussion paper only, no results
Calder et al. [48]	Detection	Bayesian segmentation	Real SSS	Flat	ROC Curve	100%	30 - 80 FA/km^2
Maussang et al. [72]	Detection	Higher order statistics	Real SAS	Not specified	ROC Curve	100%	0.1% of pixels incorrectly classified
Atallah et al. [65]	Detection	Entropy	Real SSS	Not specified			Sample set too small to draw a conclusion
Tucker et al. [66]	Detection	Coherent Correlation Analysis	Real SSS	Not specified	Table	99%	86 -213 detection per image (no image size given)
Rao et al. [35]	Detection	Pattern matching	Real SSS	Varied	ROC Curve	91%	1.5 false alarms per image (No image size given)
Del Rio Vera et al. [80]	Classification	Hilbert transform and feature extraction	Simulated SAS	Targets and Clutter Objects Only	ROC Curve	98%	0.01% of simulated cluttter incorrectly classified
Reed et al. [73]	Classification	Bayesian segmentation and feature extraction	Real SSS	Varied	ROC Curve	94%	48% of detected cluttter incorrectly classified
Petilot et al. [46]	Classification	Haar Features	Real SAS	Flat	Table	99%	0-5 FA/km^2
Hasanbelliu et al. [74]	Classification	Matched Filtering	Real SSS	Not specified	Confusion matrix	N/A	N/A
Dura et al. [31]	Classification	Superellipse fitting	Real SSS	Object snippets only	Table	70%	Not specified
Coiras et al. [76]	Classification	Central filters	Real SSS	Varied	Table	94%	0.001% at a PD of 81%
Chandran et al. [78]	Classification	Spectral Features	Real SSS	Varied	Graph	89%	9% of clutter objects classified as mines

While the current generation of model-based algorithms perform well in simple controlled environments, they are not robust with respect to difficult sea-floor types and significant target-like clutter. For this reason supervised algorithms are preferred, even given the disadvantages which we recap here. Supervised algorithms require large data sets (we show in the next chapter that the Haar Cascade requires at least 1000 views of a target to avoid over-training). Additionally, many of the algorithms calculate features directly from the images. As such they are not invariant with respect to the operational parameters of the MCM mission. Therefore, while they achieve excellent results on the training missions they do generalise well to new data.

While the supervised geometric algorithms are invariant with respect to the image

formation process they suffer from the same problems as the model-based algorithms. The performance is heavily dependent on the segmentation quality and in cluttered regions the geometric features are not sufficient to describe the difference between target and non-target objects.

In this chapter we have introduced a broad over-view of the approaches used for ATR in SSS and SAS images. We first introduced methods for detecting objects in SSS sonar images and then looked at methods for classifying those objects. From recent algorithms for high resolution SSS and SAS an interesting pattern can be observed. Algorithms can be selected for excellent performance over tightly constrained data-sets. Typically these algorithms belong to the class of supervised filter response algorithms. Alternatively, algorithms can be selected to generalise well over different data sets. However, the performance of these geometric feature based algorithms does not match that of the best filter response algorithms.

In the next chapter we look at the leading filter based ATR algorithm, the Haar-Cascade, in more detail. In the next chapter we show that the Haar-Cascade requires very large training data sets, does not perform as well for asymmetric objects and is very sensitive to the operating conditions and model of sonar. In the following chapter we introduce concepts from model-based algorithms to reduce the dependence of the Haar-Cascade on the sonar image formation mode. We demonstrate that this allows us to train a supervised filter response algorithm on one model and obtain good performance with another model of sonar.

Chapter 4

Target classification with 2D and 3D Features

4.1 Introduction

In the previous chapter we reviewed the literature on Automatic Target Recognition (ATR) algorithms for Sidescan Sonar (SSS) and Synthetic Aperture Sonar (SAS). From this review it is clear that, in simple environments, ATR algorithms are able to compete with human operators. The potential advantages gained from using ATR algorithms to assist human operators and drive behaviours in Autonomous Underwater Vehicles (AUVs) has resulted in considerable interest in these algorithms outside of the research community. At SeeByte Ltd. ATR algorithms are currently deployed to aid multi-vehicle autonomous Mine Counter-Measure (MCM) missions [3]. Additionally they are being evaluated by several navies to assist operators in traditional MCM missions.

The problem with the current state of the art supervised ATR algorithms, such as the Haar-Cascade [5], is that they require very large training data sets. The training and evaluation of a new ATR algorithm for MCM missions is time consuming and expensive. This is primarily due to the cost of data collection. To generate the training data real targets must be manufactured, deployed, surveyed and finally recovered. This process may need to be repeated both for training and validation if the ATR is applied to a new environment. SeeByte Ltd. have introduced augmented reality algorithms [18] which can reduce the need for real data. However, the ATR algorithms can often overspecialise on the simulated data and ignore real targets.

An additional problem, that has been identified recently, is the need to retrain an ATR algorithm when sensors on board vehicles are upgraded. ATR results [5, 46, 72, 80, 81] from SAS systems clearly demonstrate the advantages that can be gained from increases in resolution and image quality. However, if new target data must be collected for every potential threat type this can add a significant cost to upgrading the vehicles. However,

these problems can be mitigated if target data acquired from older sensors can be used to train a new system.

In this chapter we introduce the Haar-Cascade and show how it performs when it is trained and tested on data from different sonar. The Haar cascade was originally developed for face recognition in video [82]. It has been applied to target classification in SSS and SAS images by Sawas et al. [5, 46]. Currently, due to its low computational requirements and excellent performance, it is one of the leading algorithms at SeeByte Ltd. for embedded ATR in AUVs.

The Haar-Cascade has been applied to classify sonar data at SeeByte Ltd. However, in long term use it has been shown to have a number of practical short-comings. The novel contribution in this chapter is that we collect experimental data to demonstrate some of the practical limitations to the Haar Cascade algorithm. While there are other competing 2D classifiers such as the central filter approach of [76], the Haar Cascade is extremely computationally efficient. This is due to the use of integral images which allow feature calculations to be performed in just six look-ups of a summed area table. This combined with a multi-stage classifier approach makes it ideally suitable for work on low power embedded processors.

In the first half of this chapter we present an overview of the Haar cascade algorithm, the Haar wavelet features and the feature selection. We then present empirical results that demonstrate that the performance of the algorithm is sensitive both with respect to the quantity of training data and the model of sonar. Finally we discuss the reasons for the sensitivity of the Haar cascade to these variables.

The majority of state of the art ATR algorithms [5, 72, 76] for SSS and SAS calculate features directly from the image using 2D filters. As such the performance of the algorithms is highly dependent on changes in the Environmental and Operating Condition (EOC). This includes changes to the altitude or speed of the vehicle, the range setting of the sonar and the model of sonar. These algorithms must be retrained if the EOCs deviate significantly from the data on which the algorithm was trained. As a result very large training data sets are required and data collected from one sonar cannot be used to train an algorithm for a different sensor.

In the second half of this chapter we introduce the main novel contribution of this thesis, an algorithm that is not dependent on the EOCs and that has similar performance to state of the art filter based algorithms. We also introduce a novel method of calculating the sum of skewed rectangular regions in images. This allows our algorithm to process images at the rate they are collected, even on low performance embedded hardware.

4.2 Haar Wavelets and Haar-Like Filters

The Haar-Casacade algorithm is a filter-response algorithm similar to algorithms such as [76]. It is effective due to the generality of the feature set, the "attention focussing" behaviour of the machine learning algorithm, and the computational efficiency of the features. Unlike other filter response algorithms for SSS [58, 59, 76], the Haar feature set is over-complete. During training the classifier models the difference between the target object and the background from a combination of the most informative features. While any single feature is insufficient to describe the target, the ensemble of features creates a very strong classifier. The advantage of this approach is that any object can be classified if training data is available. However, as we show later in this section, the large number of features can lead to the algorithm over specialising on the training set. These Haar-Like features are introduced in section 4.2.

In section 4.3 we introduce the cascade classifier. The "attention focussing" behaviour of the cascade classifier is a key factor in the algorithms computational efficiency. Each classifier in the cascade is trained to reject 50% of the background while retaining 99% of the targets. When the cascade classifier is used to classify an image, sub-images rejected by a classifier are not processed by the remaining stages of the cascade. Target like objects are processed by the cascade until they are rejected or classified as targets. Therefore, the majority of the computational effort is focussed on target-like objects. Conceptually the first few stages can be considered as an object detector while the later stages act as a classifier.

In this section we introduce the 2D Haar-Like features used by the Haar-Cascade. The features are calculated by convolving a set of Haar-Like filters with the image in a window about the target. These filters are shown in figure 6.1. The reason for the filters being termed Haar-Like can be seen by looking at the "Edge Features". Features a and b are simply the 2D Haar wavelets described in appendix 5.3.

Haar-Like features are calculated at all possible scales and sizes within a window of size WxH. As such the number of features grows rapidly with window size. For example in a 24x24 window 162,336 features are calculated for only 576 pixels. The majority of these features will not describe useful information about the target. Therefore, during training, the ada-boost algorithm [83] is applied to select salient features and reduce the size of the feature set. Typically, we find that for object detection in SSS 50-200 features are retained. The feature calculation is further accelerated through the use of integral images.

In an integral image the value at any point (x, y) is the sum of all the pixels above and to the left of (x, y), inclusive:

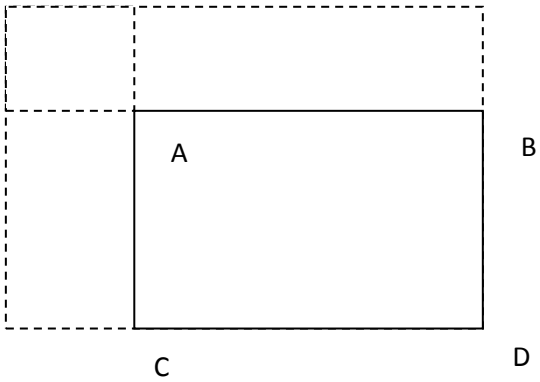


Figure 4.1: Notation for integral image calculation

$$I_{\sum}(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y') \tag{4.1}$$

The integral image can be computed in a single pass over the image, using the relationship

$$I(x, y) = i(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1) \tag{4.2}$$

Evaluating the sum over any rectangular area is a constant calculation time that is independent of the size of the rectangular area. Using the notation in figure 4.1 $A = (x0, y0)$, $B = (x1, y0)$, $C = (x0, y1)$ and $D = (x1, y1)$, the sum of $i(x, y)$ over the rectangle A, B,C and D is:

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} i(x, y) = I(D) + I(A) - I(B) - I(C) \tag{4.3}$$

There are a number of implementation of the Haar-Cascade [82, 84, 85] which include additional edge features, line features and rotated features. These are also shown in figure 6.1. In this chapter we use the OpenCV [86] implementation with the BASIC feature set. We make a number of additional modifications, as suggested in the approach of Sawas et al.[5], which we describe later in this section. In the next section we explain the training and feature selection in more detail.

4.3 Feature Selection and Classification

The Haar feature set is large and it would be intractable to calculate all features for every pixel. In this section we show how the number of features is reduced in training. First we introduce the concept of weak and strong classifiers. Weak classifiers are those for which

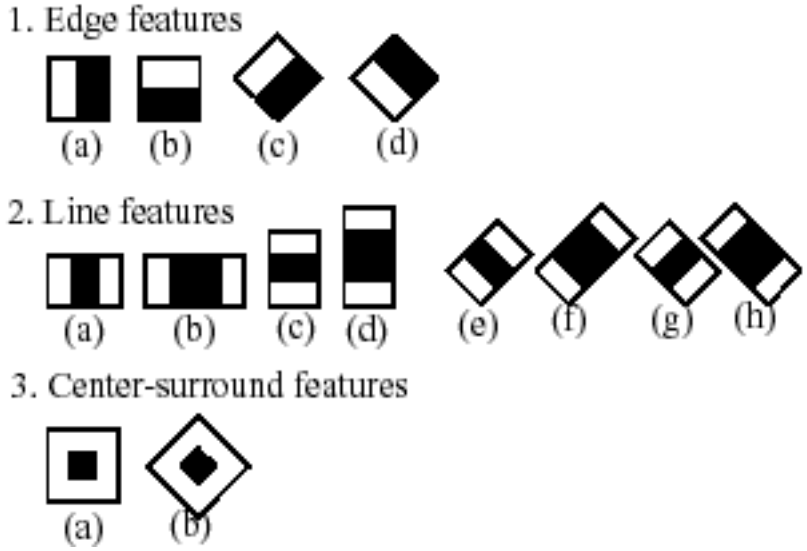


Figure 4.2: The Haar-Like feature set used in the OpenCV cascade. Black regions represent -1 and white regions +1 *Image obtained from* `http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html`

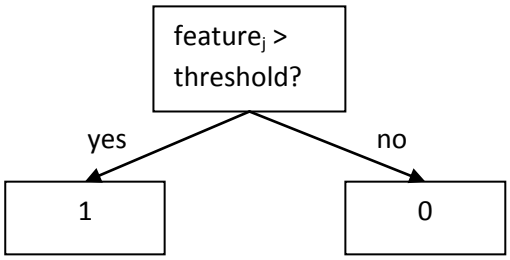


Figure 4.3: A stump decision tree is simply a cut over a single feature

the output is only weakly correlated to the true class (i.e. the classifier is slightly better than a random guess). Strong classifiers can be arbitrarily well correlated to the true class.

In the OpenCV implementation the weak classifiers are stump decision trees. Each weak classifier is essentially a cut over a single feature (figure 4.3). Given a set of images x_i with label $y_i = 0, 1$, where 0 represents a negative sample and 1 represents a positive sample we can train a set of classifiers. For each feature j we train a classifier h_j. This stump decision tree classifier simply assigns a label y based on a cut on the feature value. The training error ϵ_j for each classifier is therefore:

$$\epsilon_j = \sum_i |h_j(x) - y_i| \tag{4.4}$$

A strong classifier can be created from a set of weak classifiers through boosting. Specifically we use the ada-boost variant of the boosting algorithm. A good overview of this can be found in [87]. First stage weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ are initialized, where m and l are the number of positive and negative samples respectively. For stages $t = 1, ..., T$:

1. Normalize the weights $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$

2. Train the stump decision tree classifiers and evaluate the error with respect to $w_t, \epsilon_j = \sum_i w_{t,i}|h_j(x) - y_i|$

3. Choose the classifier, h_t with the lowest error ϵ_t.

4. Update the weights, $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$, where $e_i = 0$ if x_i is classified correctly and $e_i = 1$ otherwise. $\beta_t = \frac{e_t}{1-e_t}$

5. The strong classifier is then:

$$h(x) = \begin{cases} n/2 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq 1/2 \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases} \tag{4.5}$$

   where $\alpha_t = log(1/\beta_t)$

To summarise the process described above, for each weak classifier the classification error is calculated over the training set. The weak classifier with the smallest error is added to the boosted classifier. The weighting of the training samples is then altered such that samples classified correctly receive a lower weighting when evaluating the error for the next weak classifier. The process is repeated to build a strong classifier from a subset of the weak classifiers. Therefore, we can see that the boosting algorithm selects the most informative combination of features as a consequence of building the strong classifier.

Next we describe the training of the cascade. The majority of the background in SSS images consists of flat featureless sea-floor. Evaluating an image on a pixel by pixel basis is costly. Therefore, to minimise the number of features evaluated for simple background regions, a cascade of classifiers is used. A cascade is a sequence of strong classifiers where each classifier in the sequence is trained and evaluated on the regions that are evaluated as target like by the previous classifier. Weak classifiers are added to the strong classifier until it is able to retain greater than 99% of the positive samples and rejects 50% of the background samples.

In figure 4.4 we show how a window or sample is classified by a trained classifier. The input to the cascade classifier is a window in the image. The window is evaluated by the first stage of the cascade which we denote C1. C1 is a strong classifier formed from a boosted set of weak classifiers. Only the features required by the weak classifiers are calculated to evaluate this stage of the cascade. A weighted sum of the output from the weak classifiers H is calculated and compared to a threshold. If H exceeds the threshold then the window is classified as target like and passed to the next stage of the classifier. Otherwise, the window is rejected. In this way only the most difficult or target like background is evaluated by the later stages of the cascade.

The output of the cascade classifier is a binary decision, i.e target or non-target. Sawas et al. have demonstrated that the performance of the algorithm can be improved by calculating a confidence value from the cascade. This approach is referred to as the "Score Cascade". The confidence value is obtained by weighting the output of the boosted decision tree stages of the cascade. Each stage n produces a confidence value H_n. The final confidence value C applies a greater weighting to those stages that occur later in the cascade.

$$C = \frac{\sum_n n * H_n}{N} \tag{4.6}$$

where N is the number of stages in the cascade.

Additionally, the cascade is modified such that the last 5 stages of the cascade do not reject samples. For example in a cascade that has 20 stages, a sample that is accepted by stage 15 will be evaluated by the final 5 stages of the cascade. This approach boosts the Probability of Detection (PD) of the algorithm at the expense of a small increase in the Probability of False Alarm (PFA).

While this is by no means a full treatment of the Haar-Cascade algorithm it introduces a sufficient theoretical basis to interpret the preliminary results in the next section. Full details of the data set and the full methodology are described in the next chapter in section 2.5. Readers who are not familiar with ATR algorithms in SSS may wish to read these sections before reading the next section.

4.4 Results for Haar Cascade

In this section we evaluate the performance of the Haar-Cascade algorithm with respect to the model of sonar and the size of the training data set. As described in the previous section each stage of the Haar-Cascade is trained to accept 99% of the target objects and reject 50% of the background objects. Each cascade is trained with 20 stages.

First we make some comments about the data set. The data used in this section is produced using real background images and simulated targets as described in chapter 2. There have not been any studies comparing the difficulty of evaluating targets simulated using the sidescan simulator [18] to real data, however we can make a few qualitative observations. First the height of any objects in the original image are estimated using the classical Lambertian diffuse illumination model, this assumes that the illumination of the source is only dependent on the angle of the surface to the source. By reconstructing the original seabed height the algorithm can model partial or complete obstruction of the target. Next, the intensity of the sonar return from the targets is estimated by modelling the reflectivity of the target material and the incidence angle of the sonar to the target. Additive Gaussian noise is then applied to model the sonar noise. The sonar shadow is

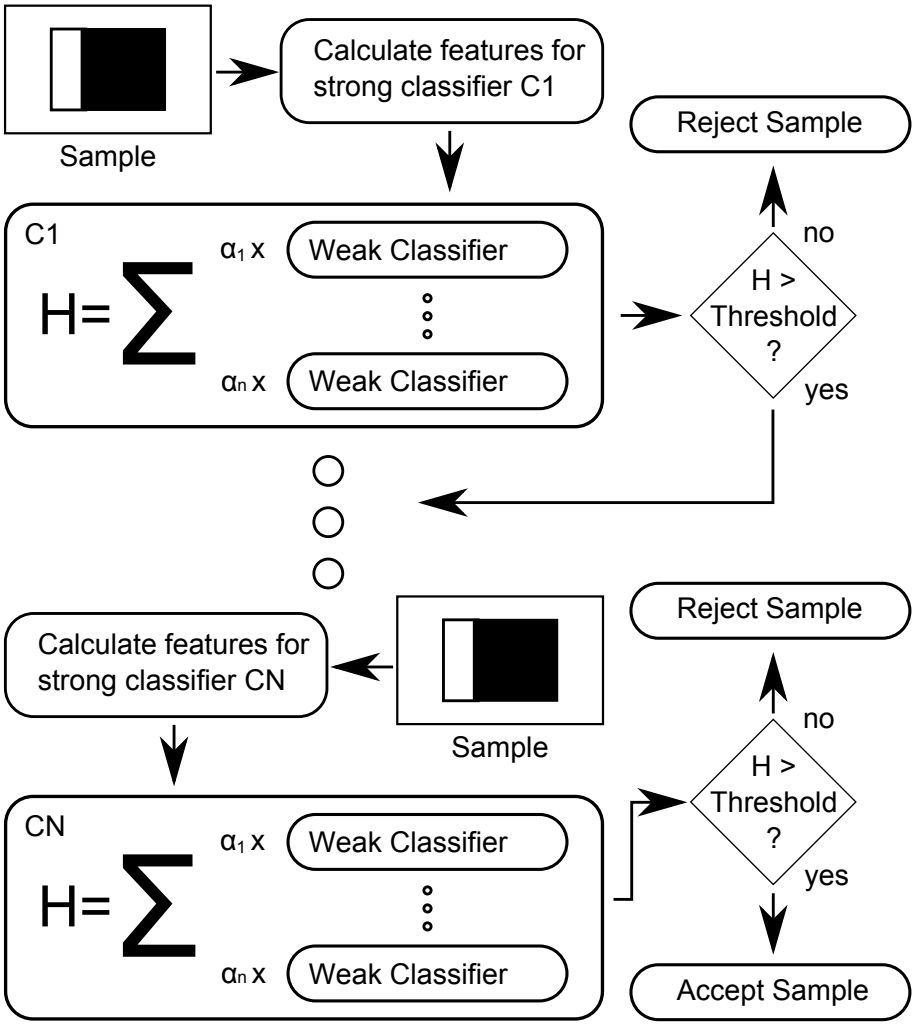


Figure 4.4: A cascade of boosted weak classifiers for classifying a window in an image. The window is evaluated by the first stage of the cascade C1. A weighted sum of the output from the weak classifiers H is calculated and compared to a threshold. If H exceeds the threshold then the window is classified as target like and passed to the next stage of the classifier. Otherwise, the window is rejected.

simulate as Gaussian noise to model the noise from volume reverberation as described in chapter 2.

As we described in the introduction we are interested in seeing if it possible to train the Haar-Cascade using data collected from a different model of sonar. Additionally, we investigate how many training samples are required for different target types. We start in figure 4.5 by comparing the performance of the Haar-Cascade trained on 1000 truncated cone targets from a Marine Sonic sonar and tested on targets from an Edgetech sonar. The number of background samples for each stage is constant at 1000 samples.

We can see in 4.5 that there is a drop in maximum PD of 34% between the two sonar types. The most obvious reason for this is the change in resolution. Marine Sonic sonar has a resolution of 6x12cm in the along and across track respectively where as Edgetech sonar has a resolution of 3x12cm. By changing the resolution the features are no longer

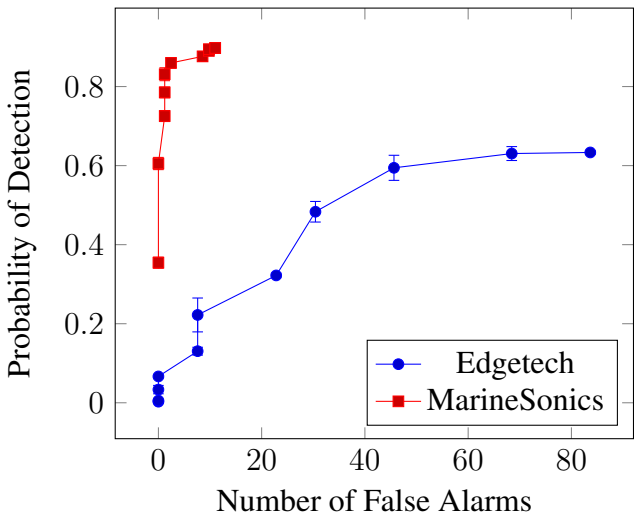


Figure 4.5: ROC curves for the Haar-Cascade algorithm when the algorithm is trained on a Marine Sonic data set of truncated cones and tested on an Edgetech and a MarinesSonics data set.

measuring the same point on the object. However, there are also more subtle effects, the training data from the Marine Sonic sonar was collected at an altitude of approximately 2.5 m while the Edgetech test data is collected an altitude of approximately 6 m. As a result the shadows are much shorter at near range and there will be a change in the intensity of the sonar return. Indeed we note that the majority of the missed targets are at near range.

The interesting question is, given the difference between the appearance of targets in the two sonar, why are any targets detected at all? This can be attributed to variability in target appearance in the training data set. The Edgetech data is sufficiently similar for some of the targets to fall within the acceptable range of features to be classified as targets. However, there is no guarantee that this would be the case for different sonar types. The problem is that the Haar-Cascade is strictly a 2D classification algorithm. It makes no allowances for the image formation model. As such the performance between any two sonar types will be unpredictable. The use of 2D features is also a problem for classifying asymmetric objects. This is demonstrated by the large data sets that are required to train the cylinder detector described in the next paragraph.

Figure 4.6 shows the dependence of the Haar-Cascade on the size of the training set. We use the Marine Sonic cylinder data set which is described in more detail in section 2. The number of background samples for each stage is constant at 1000 samples. Each stage is trained to accept 99% of the target objects and reject 50% of the background objects. Each cascade is trained with 20 stages. The cascade is trained with 1000,500,250,125 and 62 target samples.

The first observation is that reducing the number of training samples reduces the maximum PD of the algorithm. The maximum PD achieved is 92%, 86% and 83% for 1000,

500 and 250 training samples respectively. We can test the hypothesis that this is due to over-training by comparing the error on the training and test sets. We know that the PD over the training set is 0.99^{15} or 85% as we accept any sample that passes 15 stages of the algorithm. Therefore, as the maximum PD for training sets of less than 500 images is between 5% and 36% lower than 86% we can conclude that the algorithm is over specialising on the training data.

There is a surprising reduction (300 false alarms at 80% PD) in the PFA between the cascade trained on 1000 target samples and the cascade trained on 500 target samples. This could be explained by the fact that the complexity of the background increases towards the end of the mission. As the sample order was not randomised the cascade trained on 500 samples did not train on the samples with more difficult backgrounds. However, we observe the same result when we randomise the sample order and repeat the training.

In figure 4.7 we compare the Receiver Operator Characteristics (ROC) curves for the training and testing data sets with a cascade trained on 1000 target samples and a cascade trained on 500 target samples. As the training and testing ROC curves agree to within error we conclude that neither cascade has been over-trained. Comparing the results from the two classifiers shows that one detects end on cylinders while the other one does not. This explains both the increase in PD and the increase in PFA. End-on cylinders are often low contrast and only 1 pixel across in the along-track direction. Therefore, by detecting end-on cylinders the classifier must also allow a significant amount of clutter that looks similar to the end on cylinders.

From our experience it is rare that for a data set to contain 500 real views of a target. Therefore, we explore the reasons why such a large training set is required. The most likely candidate is the fact that Haar features are not invariant with respect to rotation. When the Haar-Cascade is used for face detection, faces rotated by more than 30 degrees with respect to the training set are not detected. A fact that lead to the development of the integer rotated Haar features [84].

The cylinder varies considerably in appearance with respect to orientation. As a result there is a large overlap in the feature space between target and non-target objects. This increases the number of features that are required to detect a target. The curse of dimensionality means that the larger the number of features the higher the chance that the classifier will be over-trained. We can test this hypothesis by training the cascade to detect a much simpler object such as a truncated cone. In figure 4.8 we see that there is only a 4% reduction in PD between the cascade with 1000 target samples and the cascade with 250 samples.

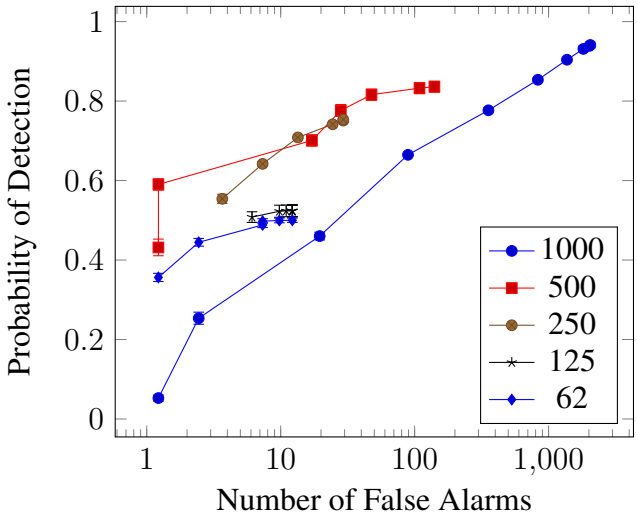


Figure 4.6: ROC curves for a Haar-Cascade trained to detect cylinders in a Marine Sonic image. The number of training samples was varied each time the training was repeated

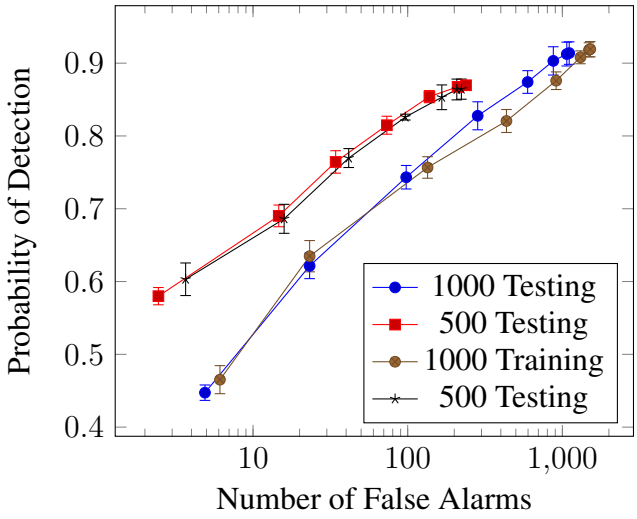


Figure 4.7: ROC curves for a Haar-Cascade trained to detect cylinders in a Marine Sonic image. The order of the training samples has been randomised. A cascade trained on 500 or 1000 target samples is tested on the training data and a previously unseen test data set

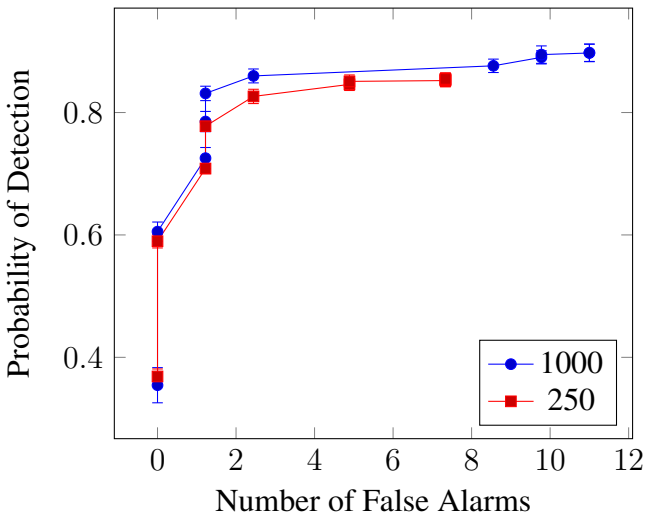


Figure 4.8: ROC curves for a Haar-Cascade trained to detect truncated cones in a Marine Sonic image. The number of training samples was varied each time the training was repeated

4.5 Summary

In this section we have introduced the Haar-Cascade algorithm. We first demonstrated that the algorithm has a low PD when it is trained and tested on different sonar types. Next we showed that for asymmetric objects the Haar-Cascade requires large training sets to prevent over-training. We identified that these issues are a result of both the appearance of the object with respect to the EOCs and with respect to the object pose.

The reduction in performance of the classifier when it is trained on one model of sonar and tested on another is due to a number of factors. Changing the sonar type causes a change in the relative intensity of the highlight and shadow regions and a change in the across track resolution. A change in the survey altitude results in a change in shadow length and the intensity of the return from an object. Finally varying the survey speed causes a change in the along track resolution. If the target features, after a change in EOCs, are not contained with the distribution of features for the training set then the target will not be detected.

The requirement for large training data sets is due to the dependence of the Haar features on the pose of an object. Conceptually the classifier must learn a different combination of features for each orientation of the target object. Alternatively, for any given feature there will be a greater overlap between the target and background distributions. This reduces the information gain from adding a feature to a classifier and therefore results in larger classifiers. The curse of dimensionality tells us that a large number of features results in a feature space in which the training data is sparse and a classifier is prone to over-training.

The variance in target observed in SSS images is largely due to the image formation

process. In theory we could produce an infinite number of different views of a target simply by varying the altitude, range and orientation of the imaging device with respect to the target. Although each view of the object is a projection of the same 3D target there is no 2D feature that will have a consistent response for all views of the object. As a result, for any given Haar feature there is a large overlap between the distribution for target and non-target samples. In theory, if we can reduce the variance of the feature space on the imaging process then we can reduce the number of features required to identify a target. This reduces the problems of over-training. Additionally by modelling the image formation process we can potentially improve the generalisation of the algorithm to new data. This approach is investigated in the next chapter.

4.6 Target Classification with 3D Features

In the previous section we demonstrated that the Haar-Cascade algorithm is sensitive to changes in the Environmental and Operating Conditions (EOCs) and is more likely to over-train for asymmetric objects. In chapter 3 we observed that some algorithms cope better than others with changes to the EOCs. For example, model based algorithms such as [32] abstract the feature space from the sensor space using an image formation model. If the Automatic Target Recognition (ATR) algorithm is not dependent on the EOCs then it may be possible to re-use real target data to train an ATR algorithm. This reduces the amount of data that needs to be collected using the new sonar system and therefore reduces costs. Unfortunately, as we showed in chapter 3, model based algorithms do not currently approach the performance of filter based algorithms.

We have already discussed the fact that 3D model based algorithm do not perform as well as 2D filter based algorithms as many targets are lost in the segmentation process. To address this problem we first need to remove the need for a segmented image. The question is how to model changes in the appearance of an object using a 2D filter. This problem has been addressed for template and contour matching algorithms [32, 36, 72, 80]. Given a 3D object we can predict the position of the highlight and shadow regions using a simple ray-tracing approximation. We refer to this as projecting the 3D object into the 2D Sidescan Sonar (SSS) image. If the projected regions are approximated as rectangles then we produce an approximation to the Haar-Like filter we described in the previous section. An example of this is shown in figure 4.9. However, there are several problems with this approach. Firstly we require a model of the target to generate the projection and secondly we need to model the appearance of asymmetric objects under rotation.

We address these problems in two ways. We remove the need for a prior object model by defining a large number of boxes with different dimensions. These are referred to as generating boxes. Unknown objects can be described by a combination of many gener-

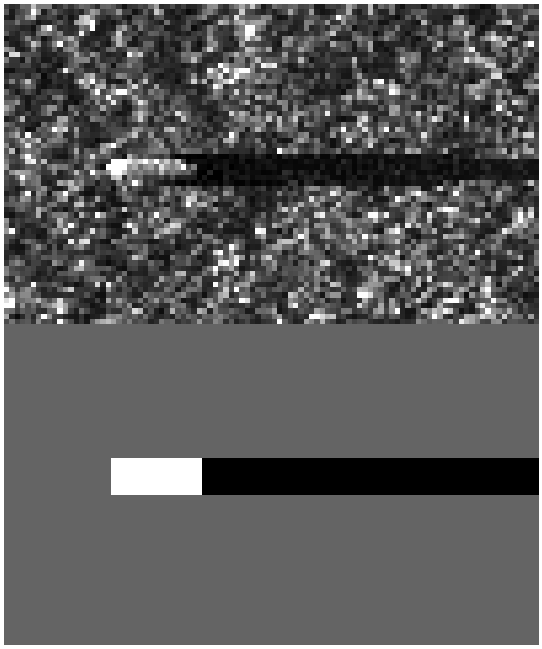


Figure 4.9: A Haar-Like feauture that approximates the position of the highlight and shadow regions of a target in a SSS image

ating boxes. The specific combination is learnt by the algorithm during training. Each feature is described uniquely by the size of the generating box. Therefore, once we have trained a classifier on one set of EOCs we can adapt to a new set of EOCs simply by defining a new projection model for the generating boxes.

The problem of modelling objects under rotation is solved by rotating the generating boxes and calculating the projection. We show that the projection of the rotated boxes can not be described accurately by a rectangular region. Consequently the standard integral image can not be used to accelerate the calculation of the sum of the Haar-Like filters. However, in section 4.9 we show that the projection of the rotated generating boxes can be approximated as a skewed rectangle. Therefore, we introduce a novel method to calculate integral images for skewed rectangular regions. The final modification that we make to the standard Haar-Cascade is to calculate both the 1st and 2nd order statistics over the highlight and shadow regions. It has been demonstrated in a number of papers [15, 70, 88] that the ratio of the 1st and 2nd order statistics is a salient feature for target detection in SSS and Synthetic Aperture Sonar (SAS) images. Therefore, we include this information to improve the classification accuracy of the algorithm.

In this chapter, we start in section 4.7 by discussing the image normalisation. In section 4.8 we introduce the integer skewed integral images. In section 4.9 we introduce the 1st and 2nd order features. In section 2.5 we describe the data sets that we are testing and in section 4.11 we present comparative results for the standard Haar-Cascade and our 3D features. Finally in section 6.6 we summarise the chapter and present our conclusions.

4.7 Image Normalisation

In this section we discuss the effect of image normalisation on the ATR algorithm. The Haar-Cascade is designed to be invariant to illumination. This is achieved by normalising the features response with respect to the local background intensity. This local approach to normalising the intensity of the image artificially boosts the contrast of noise in dark regions. Additionally, it prevents the absolute value of the pixels from being used as a feature. However, some form of image normalisation is required to correct for factors such as beam pattern and sonar loss [20] and to adjust for the illumination difference between sonar types.

In chapter 2 we reviewed a number of methods for normalising SSS images. The advantage of these algorithms over the local approach used by the Haar-Cascade is that we can retain relative intensity information between different areas while removing range and beam angle dependent effects. In this chapter we use the Serpentine Forward-Backward Filter (SFBF) method of Dobeck [22], this is described in chapter 2. The SFBF uses adaptive filters which tracks paths of similar intensity through the image. The filters follow surface returns and beam patterns effectively removing unwanted artefacts from the image. The SFBF algorithm is applied to the images before training for both the standard Haar-Cascade and our approach. After the images are normalised the next step is to calculate the integer skewed integral images that are used to accelerate the calculation of the 3D features.

4.8 Integer Skewed Integral Images

In section 4.2 we introduced the integral image (or summed area table) that can be used to accelerate the calculation of the sum of a rectangular region in an image. Approaches such as [85] extend the integral image method to calculate the sum of rotated rectangles in an image. However, as we show in section 4.9 neither rectangular or rotated rectangular regions are a good approximation the appearance of the projection of the generating boxes. For this reason we introduce a new approach which we refer to as Integer Skewed Integral Images.

Integer skewed integral images are images in which subsequent rows of the image have been skewed by an angle $arctan(\alpha/\beta)$ where α and β are integer constants. The integer skewed integral images can be used to calculate skewed rectangular regions which approximate the appearance of the projection of the generating blocks in SSS. The formulation is similar to that of the standard integral image with the addition of a skew function $s(y, y')$ to calculate the horizontal shift between each row of the image.

The integral image $I(x, y)$ is calculated from the normalised image $i(x, y)$ as

$$I(x,y) = \sum_{\substack{y' \leq y \\ x' \leq \max(0, x - s(y,y'))}} i(x',y') \tag{4.7}$$

where

$$s(y,y') = \left(\left\lfloor \frac{y}{\beta} \right\rfloor - \left\lfloor \frac{y'}{\beta} \right\rfloor \right) \times \alpha \tag{4.8}$$

A number of skewed integral images, corresponding to different skew angles, are calculated. The skewed rectangular region is defined in figure 4.10. The four corners of the region are labelled A, B, C and D. The region is skewed through an angle $\theta = tan(\alpha/\beta)$ where α and β are integer shifts. The sum of the region can then be calculated as:

$$\sum i(x',y') = I(A) + I(C) - I(B) - I(D) \tag{4.9}$$

where

$$\begin{aligned} A(y) < y' \leq C(y) \\ A(x) - s(y', A(y)) < x' \leq C(x) - s(y', A(y)) \end{aligned} \tag{4.10}$$

And valid positions for A,B,C,D are

$$\begin{aligned} A(y) &= B(y) \\ C(y) &= D(y) \\ A(x) &= D(x) - s(D(y), A(y)) \\ B(x) &= C(x) - s(B(y), C(y)) \end{aligned} \tag{4.11}$$

In the remainder of this section we adopt the short hand notation $I(A, B, C, D)$ to represent the sum of a region defined by points A, B, C and D.

The skewed integral images are used to approximate the highlight shadow geometry produced by a box located on the sea-floor. By varying α and β as in table 4.1 an approximately uniform distribution of box angles relative to the sonar array can be created. In the next section we introduce a number of different features that can be calculated from the Integer Skewed Integral Images.

4.9 Normalised 3D Features

In this section we introduce the normalised 3D features. We first define the set of generating boxes that are used to calculate the location of highlight and shadow regions in the

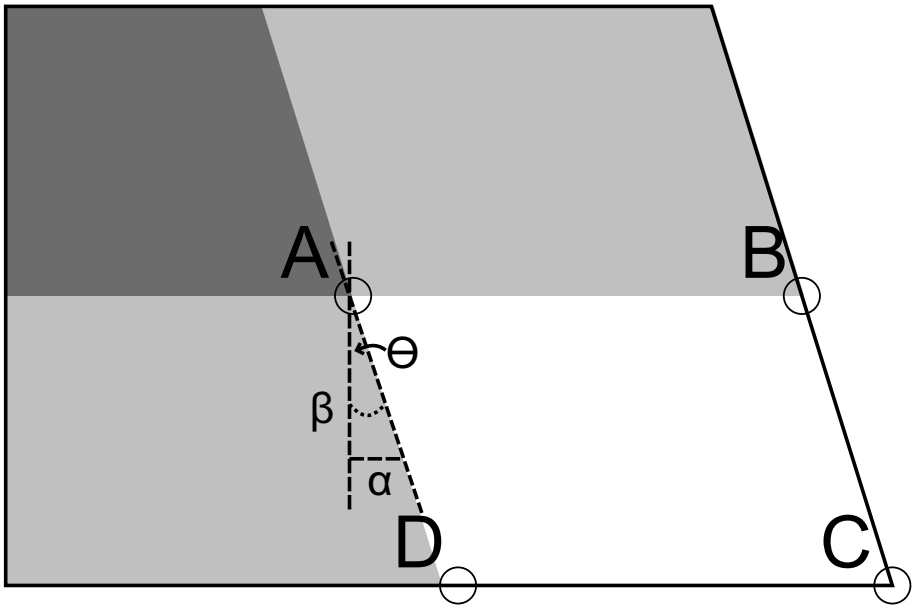


Figure 4.10: An integral of a region can be reduced to a function of four points in the integer skewed integral image representation

image. Next, we define Haar-Like features and features with additional skewed rectangles that can be viewed as analogues for the edge-detect and blob-detect features in the standard Haar-Cascade. Finally we introduce the 2nd order features which are used to calculate the variance over the highlight and shadow regions.

It is easy to confuse the terminology therefore we define a number of terms that are used throughout this chapter.

Definition 1. *Generating Box: A box with across-track dimension t_x along-track dimension t_y and height t_z.*

Definition 2. *Template: A projection of a generating box into a sonar image using a simplified image formation model. The template has one or more skewed rectangular regions that approximate the highlight or shadow regions of the generating box.*

Definition 3. *Template Response: The convolution of a template with an image at a particular point in the image. For a point (x, y) in the image the template response T is:*

$$T(x,y) = \sum_{i=0}^{T_{rows}} \sum_{j=0}^{T_{cols}} |I_s(x+i, y+j) - I_t(i,j)| \tag{4.12}$$

where $I_s(.)$ is the intensity of the source image and $I_t(.)$ is the intensity of the template image.

Definition 4. *Feature: The most target like template response for all rotations of the generating box. The set of features form the feature space that is used to classify the target.*

α	β	Angle
-4	1	-76°
-2	1	-63°
-1	1	-45°
-1	2	-27°
-1	4	-14°
0	1	0°and 90°
1	4	14°
1	2	27°
1	1	45°
2	1	63°
4	1	76°

Table 4.1: An approximately uniform distribution of skew angles between $\pm 90°$

Using these definitions, the method for calculating feature responses for a patch in an image is described below.

Generating boxes A set of generating boxes with across track dimension x, along track dimension y and height z are defined (figure 4.11)

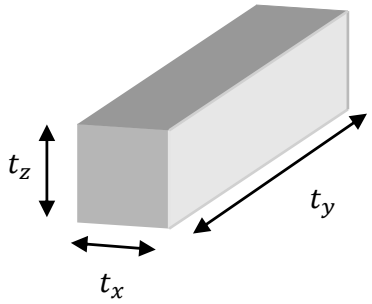


Figure 4.11: Dimensions of a generating box.

Project generating boxes into image The projection of the generating box is calculated locally using a simple image formation model. For each pixel we know the grazing angle ϕ and the local image resolution. The sonar image formation is approximated using ray tracing as shown in Figure 4.12. The size of the highlight region in the across track direction h_x is related to the grazing angle ϕ as $h_x = t_x / \cos(\phi)$. The shadow length s_x is $s_x = t_z / sin(\phi)$.

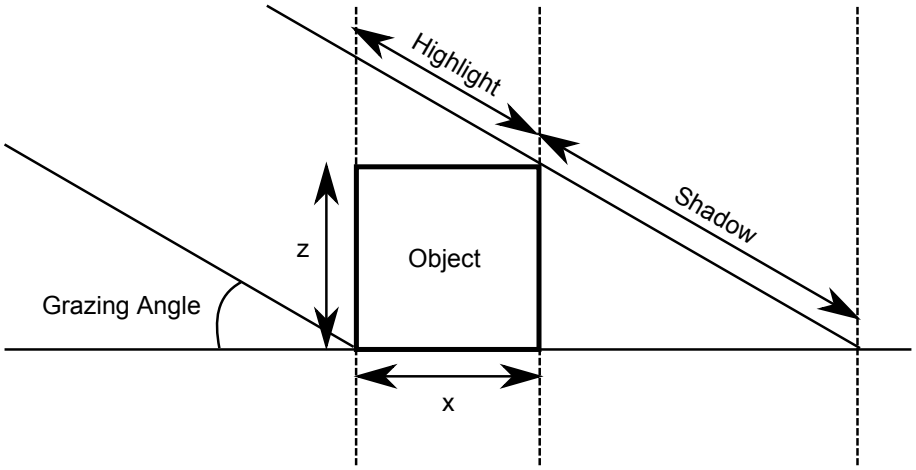


Figure 4.12: A model for the size of the highlight and shadow regions for a cube like object. The object has height t_z and width t_x and the Highlight and Shadow length are calculated for an image which has not been slant angle corrected.

The orientation of the target is not known a priori, therefore we project the generating box at a number of angles defined in table 4.1. A rotation θ of the target is approximated as a skew of the highlight and shadow regions. This allows the features to be calculated using the integer skewed integral images. For an angle θ the highlight size in the across track direction is $h_x = \frac{x}{\cos(\phi)\cos(\theta)}$, the shadow size s_x is invariant under rotation. The Highlight and shadow size in the along track direction varies with θ as $h_y = s_y = t_y \cos(\theta)$. This is shown in figure 4.13.

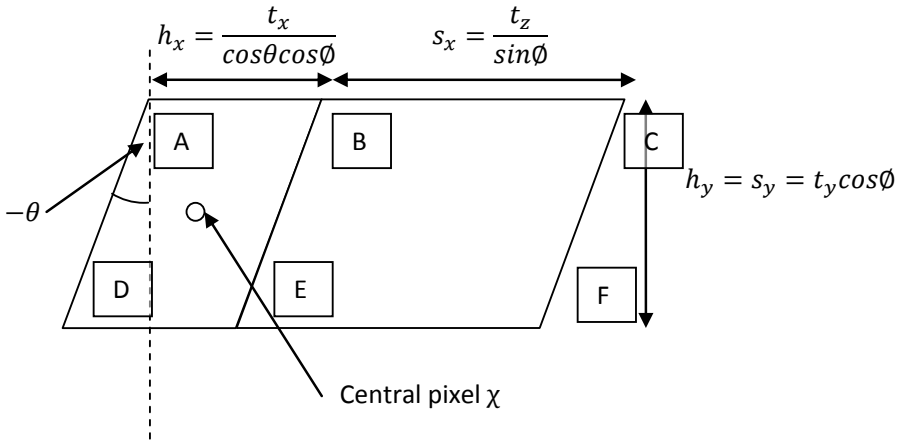


Figure 4.13: A model for the generation of a template from the projection of a generating box.

In the example shown in figure 4.13, six points A,B,C,D,E,F are defined. These points are defined relative to the central position χ for which the template response is calculated.

$$(A_x, A_y) = (\quad \chi_x - h_x/2 + \frac{h_y}{2} cos\theta, \chi_y - h_y/2 \quad) \tag{4.13}$$

$$(B_x, B_y) = (\quad \chi_x + h_x/2 + \frac{h_y}{2} cos\theta, \chi_y - h_y/2 \quad) \tag{4.14}$$

$$(C_x, C_y) = (\quad \chi_x + h_x/2 + s_x + \frac{h_y}{2} cos\theta, \chi_y - h_y/2 \quad) \tag{4.15}$$

$$(D_x, D_y) = (\quad \chi_x - h_x/2 - \frac{h_y}{2} cos\theta, \chi_y + h_y/2 \quad) \tag{4.16}$$

$$(E_x, E_y) = (\quad \chi_x + h_x/2 - \frac{h_y}{2} cos\theta, \chi_y + h_y/2 \quad) \tag{4.17}$$

$$(F_x, F_y) = (\quad \chi_x + h_x/2 + s_x - \frac{h_y}{2} cos\theta, \chi_y + h_y/2 \quad) \tag{4.18}$$

First order template response The template response S_i for a skewed rectangle R_i is calculated as $S_i = \sum_{x,y \in R_i} I_t(x, y) = I(.,.,.,.)$ where i represents one of the 11 possible rotation of the generating box. This calculation is performed using the integer skewed integral images introduced in the previous section. For the example shown in figure 4.13 the sum of two regions is calculated. The highlight region is defined as IA, B, D, E and the shadow region as $I(B, C, E, F)$.

Normalisation of template responses The size of the templates varies with respect to orientation of the generating box and the grazing angle. For the template response to be independent of the these parameters the template response must be normalised. The sum S_i of the region R_i is dependent on the area of (or the number of pixels in) the templates N_i. Therefore, to normalise the template response with respect to grazing angle and orientation, the mean response $\mu_i = \frac{S_i}{N_i}$ is calculated.

Additional template responses Figure 4.13 showed the projection of the highlight and shadow regions for a box like object and in the previous section we showed how to calculate the mean of these regions. To detect the edge regions at the front and side of the object we define a number of other features. The simplest feature is the object template. The "Object Template" feature is the shadow response subtracted from the highlight response. The "Bounded Front" and "Bounded Side" templates are shown in figure 4.14. The features calculated from these templates are designed to remove larger objects or patterns that contain a target like signature as part of the larger object. For the "Bounded Front" object an additional skewed rectangle of the same size as the highlight region is added to the front of the standard two rectangle template. The mean of this rectangle is subtracted from the template response. If this additional rectangle is in the highlight region of a larger object then it will reduce the overall template response compared to a target object on flat sea-floor. A similar logic is used to define the "Bounded Side" template, if the template is placed on part of a larger pattern such as a ripple then this

will reduce the template response compared to an object on flat sea-floor. These features are summarise in table 4.2.

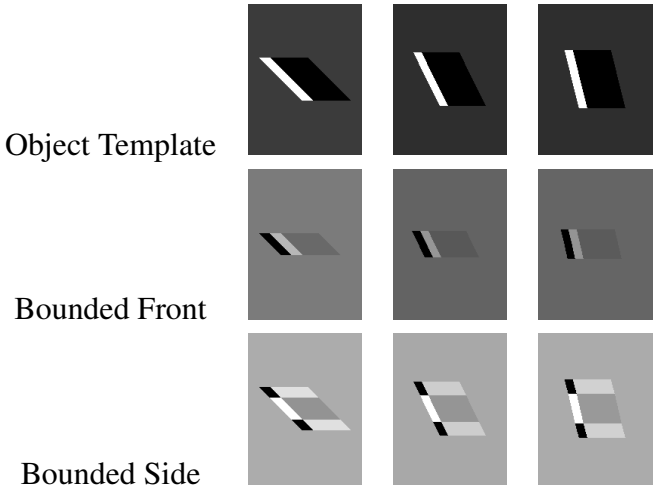


Figure 4.14: Example of the extended feature set. The Bounded Front and Bounded Side features can be used to eliminate objects which are larger than the target.

Feature calculation For each generating block and template type there will be 11 template responses corresponding to 11 rotations of the block. The rotation angle which best matches the rotation angle of the target will have the highest ratio of the backscatter to background or shadow pixels. As the highlight region is brighter than the surrounding sea-floor the maximum template response is most likely to match the orientation of the target. Similarly the shadow is darker than the surroundings.

This allows us to define two rules for calculating the best feature value from the 11 template responses. Features F are calculated from the template responses T_i by reducing them to a single feature value. The response are combined by selecting the most target like value over all the template responses. The most target like response is dependent on the feature that we are calculating. Two rules are used to select the best template response the Maximum rule, $F_i = \max_i T_i$ and the Minimum rule $F_i = \min_i T_i$.

Second order template response The variance or second order template response are calculated in a similar manner. The mean value μ_i of a rectangular region R_i is known from the first order template response. The expectation of the square of the image values $E[I^2]$ can be calculated by finding the sum of the square of the image values $I(x, y)^2$. Again, we can accelerate this process using integral images. The variance σ_i^2 on the pixels in a rectangle R_i is then $\sigma_i^2 = E[I^2] - \mu_i^2$.

The second-order template response calculates the variance over the highlight and shadow regions. These features are referred to as “Highlight Variance” and “Shadow Variance” respectively. The addition of the “Highlight Variance” to the ”Shadow Variance” is referred to as the ”Object Variance”. When these regions match the rotation of the object it

is assumed that the majority of the pixels will be generated from the same stochastic process. Therefore, the variance will reach a local minimum. These features are summarise in table 4.2.

Name	Description	Selection rule
Highlight	The mean of the highlight region	Maximum value
Shadow	The mean of the shadow region	Minimum value
Object Template	Shadow subtracted from highlight	Maximum value
Bounded Front	Object template with additional rectangle bounding the front of the object (fig 4.14)	Maximum Value
Bounded Side	Object template with additional rectangle bounding the side of the object (fig 4.14)	Maximum value
Highlight Variance	Variance of the highlight region	Minimum value
Shadow Variance	Variance of the shadow region	Minimum value
Object Variance	Highlight Var + Shadow Var	Minimum value

Table 4.2: Description of the 8 features calculated for each block. The most target like value over all rotations is selected using the selection rule

We compare the response from the "Object Template" feature to the equivalent Haar Feature. Figure 4.15 shows the difference between the actual projection (top) and the skewed rectangle approximation (middle). The difference between the two images is shown in white(bottom). In figure 4.16 we compare the variation in feature response for the 3D feature in figure 4.15 and a Haar feature with the same dimensions as the 3D feature with $\phi = 0$. The feature values are normalised such that the maximum response is equal to 1. The 3D feature response varies by at most 10% over all rotations while the Haar feature response, in comparison, varies by as much as 90%.

Finally we introduce a practical consideration for any implementation of the 3D features. Unlike the Haar features which operate on a fixed size sliding window the size of the 3D features is dependent on the altitude of the vehicle and the range of the pixel of interest. As a result we must deal explicitly with the image boundaries. The size of the largest possible feature is calculated for the image and the image is padded by reflecting the image about the boundaries. In this way the algorithm will always be able to access valid image pixels, however the feature response at the image boundary are not always representative of the rest of the image.

4.10 Experimental Set-Up

In this section we describe the experimental set-up. We use the same set-up for training and testing both the Haar features and the 3D features. Both implementations use the

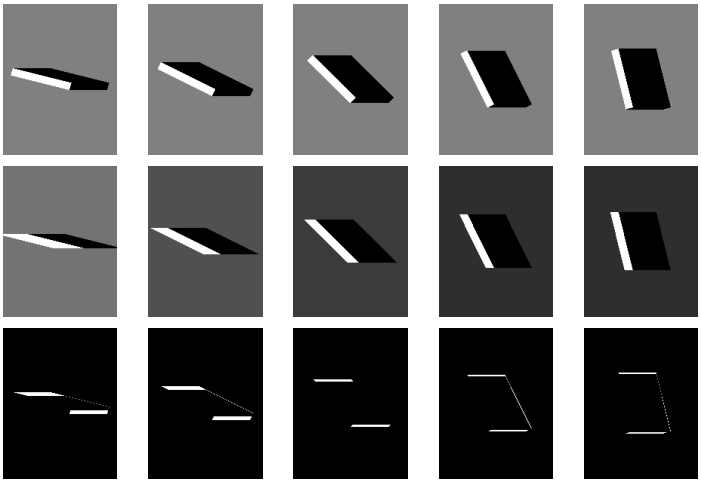


Figure 4.15: Example of the highlight shadow geometry for a cube (top) the skewed integral image approximation (middle) and the difference between the two representations (bottom)

OpenCV 2.1 [86] implementation of a cascade classifier and the OpenCV implementation of the Haar features. A number of changes are made to the OpenCV framework to allow the integration of the 3D features. The sonar images are split into 2 images. Each sub-image contains a single channel of the sonar image oriented with zero range on the left hand side of the image and far range at the right hand side of the image. For each image we also provide a data file which contains the altitude of the vehicle, the image resolution and the range of the first column in the image. The values are assumed to be constant for each image. This information is used by the 3D features and discarded by the Haar features. No other changes were made to the training process. For testing we implemented the score cascade approach described in section 4.3.

We perform a large number of tests with different configurations, to avoid confusion these are described before the respective results. The default configuration is as follows. The cascade is trained with 1000 positive and 1000 negative samples for each ada-boost stage. Additional stump decision-trees are added to the ada-boost algorithm until at least 50% of the background samples are removed with more than 99% of the positive samples retained. A window size of 80x30 pixels is used for the Haar-Cascade, the 3D features do not require a window size. The basic Haar feature set is used as described in section 4.3.

The Haar-Cascade is always trained with 20 stages in the cascade. We observed that training the Haar-Cascade with more stages reduced the maximum Probability of Detection (PD). The theoretical minimum PD over the training set for the Haar-Cascade using the Score Cascade is $0.99^{N_{stages}-5}$. In practice we observe, for asymmetric objects, the Haar-Cascade obtains results close to the theoretical minimum. Consequently, training the Haar-Cascade with 30 stages reduces the maximum PD to approximately 80%. A PD this low is unsuitable for many applications in mine warfare. Unlike the Haar features we can reduce the false alarm rate without impacting the PD of the 3D features algorithm by

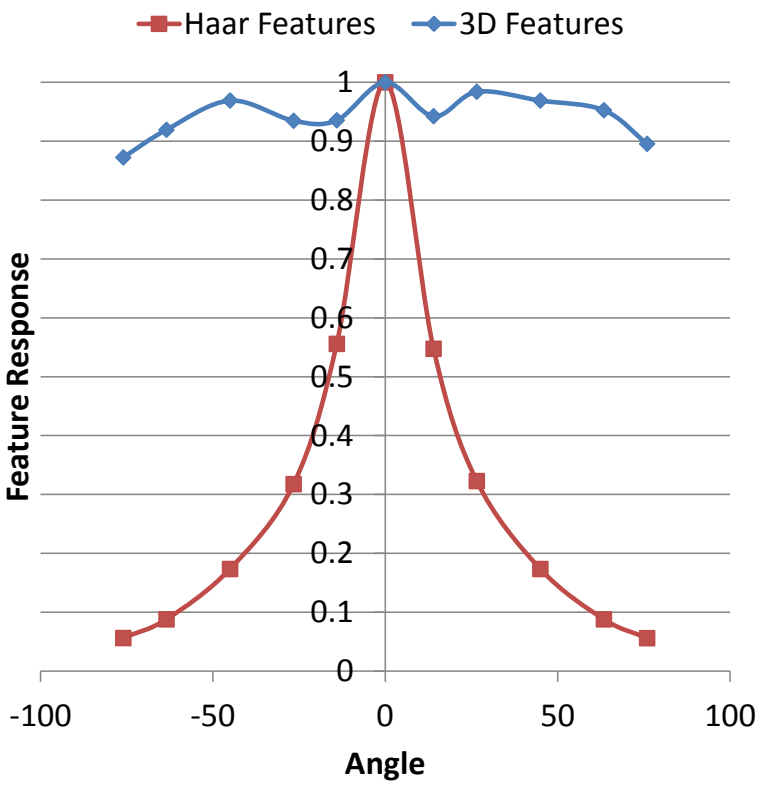


Figure 4.16: A comparison of the skewed rectangular feature and the standard Haar feature response for the elongated cube shown in figure 4.15.

training 30 stages. Therefore, unless otherwise stated the 3D features are trained with 30 stages in the cascade.

The generating boxes used to create the 3D features are defined by their x, y and z dimensions. For each dimension a range is defined from a minimum size to a maximum size with a given step size. The various configurations that we use are shown in table 4.3. The first Feature Set tested is the "Small Low Res" feature set. This contains a large step size and a maximum feature size equivalent to the object size. In the "Small High Res" feature set we double the resolution of the step size. In the "Large High Res" feature set we double the maximum feature size. In theory the optimal feature set would have no limits on the size or resolution of the generating boxes. However, the feature sets that we test are limited by practical considerations such as training time and memory usage. In the next section we compare the performance of the different configurations of 3D features with the Haar features.

	X (m)	Y (m)	Z (m)
Small Low Res	0.2:0.2:1.0	0.2:0.2:2.0	0.1:0.1:1.0
Small High Res	0.2:0.1:1.0	0.2:0.1:2.0	0.1:0.05:1.0
Large High Res	0.2:0.1:2.0	0.2:0.1:4.0	0.1:0.05:2.0

Table 4.3: Distribution of generating box dimensions for the three features sets used in this chapter. The dimensions are in meters. The format is Min:Step:Max

4.11 Results and Analysis

In this section we compare the performance of the 3D features with the current state of the art, the Haar-Cascade. In section 4.11.1 we compare the performance of the two algorithms when they are trained and tested on images from the same data set. By varying the size and resolution of the generating boxes we show that the 3D features can obtain similar performance to the Haar-Cascade. In this section we also compare the performance between different target types and investigate the number of training samples that are required to optimise the performance of the cascade. In section 4.11.3 we compare the performance of the two features when they are trained on one type of sonar and tested on another.

4.11.1 Performance

In this section we compare the performance of the 3D features and the Haar features when they are trained and tested on the same model of sonar. This represents the typical operating conditions for the majority of ATR algorithms that have been evaluated to date. First we look at the effect of the size of the training data set of the performance of the 3D features algorithm. We compare this to the results we produced for the Haar-Cascade in chapter 4. Next, we compare the performance of the algorithm with respect to different resolutions and sizes of generating boxes. Finally we look at the effect of the number of cascade stages on the performance of the 3D features algorithm.

Training Size

In chapter 4 we discussed the fact that the Haar-Cascade requires a large number of training samples to avoid over-training on asymmetric objects. We observed a significant reduction in the performance when we moved from 1000 training samples to 500 training samples. We have repeated this test with the 3D features using the “Small Low Res” feature set and a cascade trained with 20 stages. The results are shown in figure 4.17. The Receiver Operator Characteristics (ROC) curves are produced using the MATLAB perfcurve function. Unlike the Haar features there is little difference between the performance of the cascade trained on 1000 or 500 training samples. However, like the

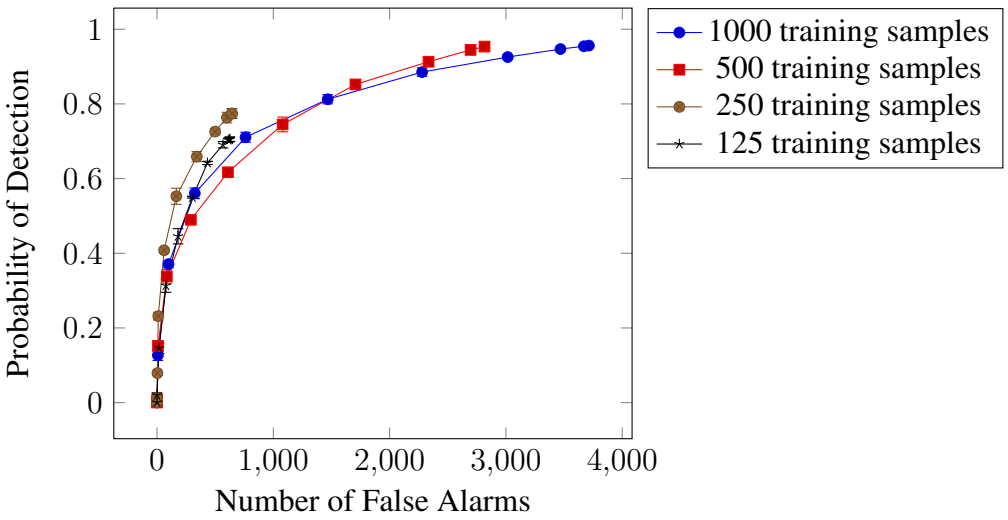


Figure 4.17: Comparison of the performance of the 3D features algorithm on cylindrical targets in Marine Sonic data with respect to the size of the training data set.

Haar-Cascade, moving to 250 training samples or less has a significant impact on the performance.

The reduction in the number of training samples can be attributed to the fact that the cascade using the 3D features has significantly fewer stump decision-trees per cascade stage. For all configurations tested the Haar-Cascade uses 140 ± 10 stump decision trees while the 3D features cascade uses 81 ± 7 stump decision trees. The reduction in the complexity of the cascade stages reduces the likelihood that the cascade will over-train. Therefore, we required fewer training samples.

Size and Resolution of Features

In figure 4.18 we evaluate the performance of the 3D features with respect to the number and resolution of the generating boxes. The results shown are for cylindrical objects in Marine Sonic images and we compare the results to the Haar-Cascade trained with 20 stages. First we look at the results for the "Small Low Res" feature set described in table 4.3. With this set of features the cascade trained with 20 stages produces approximately double the number of false alarms compared to the Haar-Cascade. This can be improved slightly by training an additional 20 stages. However, the results are still significantly worse than the Haar-Cascade. Interestingly we note that unlike the cascade trained with the Haar features the cascade trained with the 3D features does not over-train even with 40 stages.

The performance of the 3D features can be improved by increasing the resolution of the generating boxes. This is shown in the results for the "Small High Res" feature set trained with 20 stages. Here the results agree to within 1 standard error to those for the Haar-Cascade. Increasing both the resolution and the maximum size of the generating

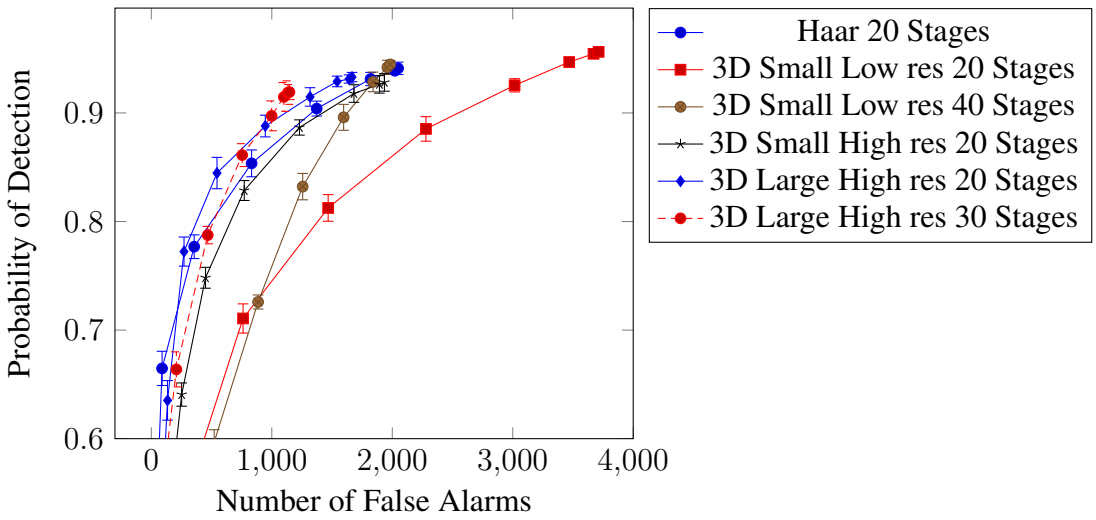


Figure 4.18: Comparison of the performance of the 3D features and the Haar features algorithm on cylindrical targets. The effect of the resolution and the maximum size of the features on the performance is compared

boxes, as in the "Large High Res" feature set, results in a small increase in performance over the Haar-Cascade. Again further improvements are seen by increasing the number of cascade stages to 30.

4.11.2 Symmetric and Asymmetric Objects

We have shown that the "Large High Res" feature set with a cascade trained with 30 stages produces the best results for the cylindrical target from the 5 different configurations that were tested. However, it would not be correct to say that this is the optimal configuration as the training is extremely computationally intensive and it would be impractical to run the number of tests required to produce a truly optimal configuration. Regardless, we conclude that for cylindrical targets the 3D features are at least as good as the Haar features. Additionally, for the cylindrical data set, the 3D features can be trained with half the number of training samples.

These results are repeated when we test the 3D features on cylinders in the Edgetech images. In figure 4.19 we compare the performance of the 3D features to the Haar features on cylinders in an Edgetech image. In this case we see a significant reduction in the number of false alarms for the 3D features compared to the Haar features. The improvement in performance on the Edgetech data could be due to the increased resolution of the sensor, however it could also be attributed to the difference in sea-floor types between the two data-sets. Further work would be required to draw any conclusions about the performance differences between sensor types.

Next we evaluate the performance on the truncated cone objects. In the following experiments we use the "Large High Res" feature set and a cascade trained with 30 stages.

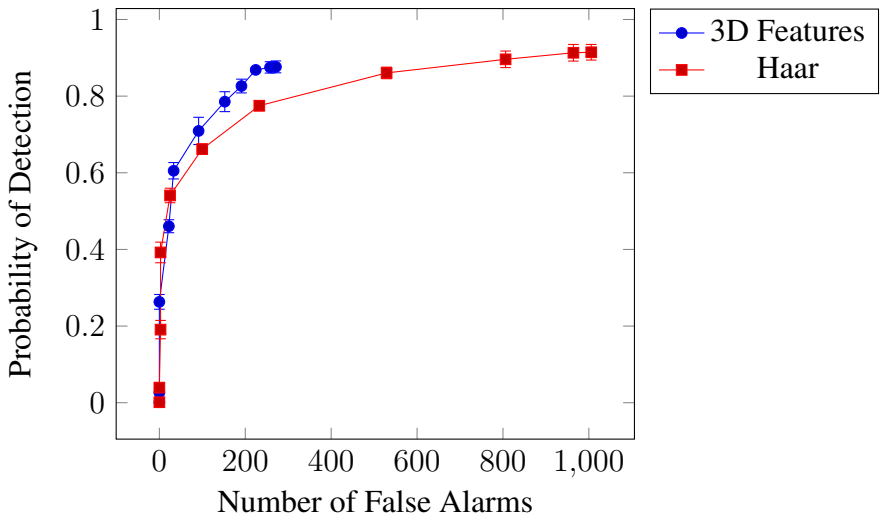


Figure 4.19: Comparison of the performance of the 3D features and the Haar features algorithm on cylindrical targets in Edgetech images.

Results for the Marine Sonic images are shown in figure 4.20. In this case the Haar features produce significantly fewer false alarms that the 3D features. This can be attributed to the fact that the truncated cone is symmetric and does not change its appearance with respect to rotation. By allowing the 3D features to test all possible orientations of an object we actually make it harder for the algorithm to learn the difference between the appearance of the object and the background.

Difficulty of Sea-floor

Next in figure 4.21 we compare the performance of the algorithm in flat and complex regions of the sea-floor. As in the previous experiments, in flat terrain the results from the 3D features are comparable to the Haar features. However, in complex terrain the 3D features demonstrate an increase in Probability of False Alarm (PFA). As in the previous section, this can be attributed to the fact that the 3D algorithm calculates features from the most target-like aspect of a false alarms. Therefore, the probability of a false alarm having target-like features is increased. We also note that there is a significant reduction in the PD between the two sea-floor types. We attribute this to the fact that the simulated mines often fail to render correctly in the complex regions and that mines in complex regions only form 9% of the training data.

Computation Performance

Finally we comment on the computational performance of the two approaches. This is tested on a single core of an Intel i7 1.60GHz CPU. We note that the training times that we report are significantly faster than the standard OpenCV cascade training algorithm. This is due to a number of optimisations that we have made to the training process. However,

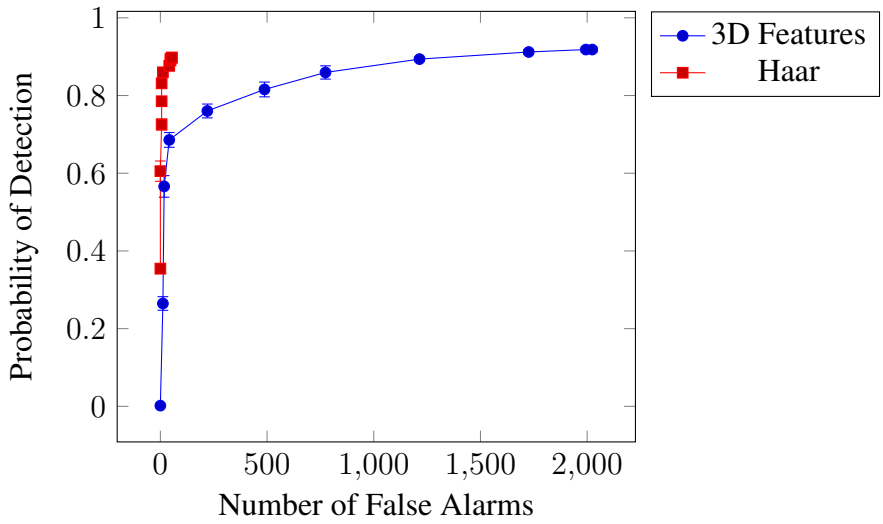


Figure 4.20: Comparison of the performance of the 3D features algorithm on truncated cone targets in Marine Sonic data. The data set is split into flat images and complex images with ripples and rocks

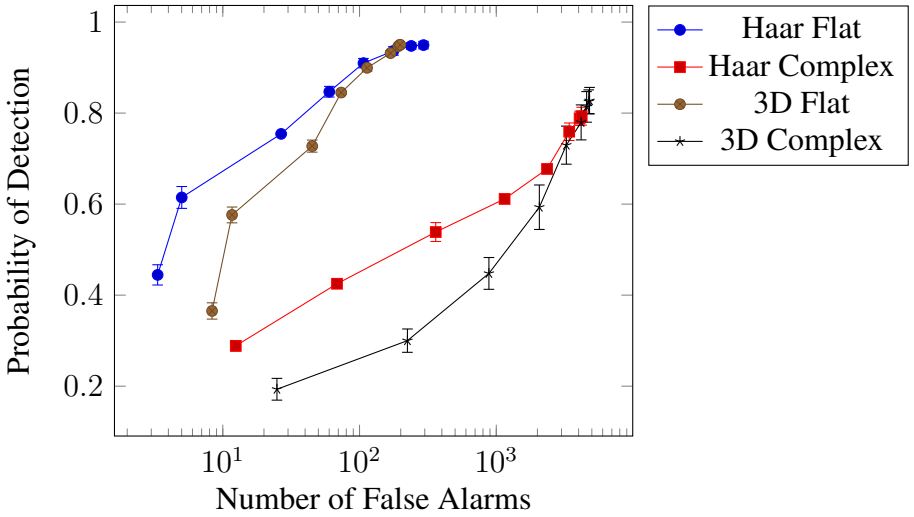


Figure 4.21: Comparison of the performance of the 3D features algorithm on cylindrical targets in Marine Sonic data. The data set is split into flat images and complex images with ripples and rocks

these are commercially sensitive and are not be included in this thesis. As both systems were trained with the same optimisations enabled the comparative results are still valid.

The training time for a 20 stage classifier with the Haar features on a 30x80 window is 35 ± 7 minutes. The equivalent training times with the "Small Low Res", "Small High Res" and "Large High Res" 3D feature set are 14 ± 3 minutes, 32 ± 5 minutes and 120 ± 20 minutes respectively. In the case of the "Large High Res" feature set the large increase in time is due to the calculated features exceeding the available memory cache allocated by the algorithm. This results in the features being recalculated several times during the training process. The runtime of the algorithm on the Marine Sonic images for the cylindrical targets is 80 ± 5 ms for the Haar features and 1650 ± 20 ms for the 3D features. The increase in run-time is a combination of the calculation of the additional integral images and the need to calculate 11 template responses to derive each feature value.

Summary

In this section we have presented the results from a number of experiments that compare the performance of the 3D features with the Haar features. We have shown that in general the performance of the 3D features is comparable, or slightly better, than the Haar features. Additionally the cascade trained on the 3D features requires half the number of training samples before the cascade becomes over-trained. We conclude that the 3D features approach is at least as good as current state of the art for object classification in SSS imagery. The time required to train the two algorithms is highly dependent on the exact parameters of the algorithm but is of a similar order of magnitude.

The most significant draw-back encountered with the 3D features is the 20 fold increase in runtime. However, due to the use of integral images and the attention focussing, in which the majority of the image is rejected by an early stage of the cascade, the algorithm still analyses the images faster than the time required to collect the data (about 30s for a Marine Sonic image). In the next section we demonstrate the key improvement offered by the 3D features algorithm. That is the ability to re-use an existing target database when the algorithm is trained for a new sonar.

4.11.3 Multi-Sonar Classification

In this section we consider the performance of the 3D features and the Haar features when a cascade classifier is trained on one model of sonar and used to detect targets in another model of sonar. In this section we are more interested in the maximum PD than the PFA. This is because the PFA can be reduced by training the algorithm with representative background data. This is relatively inexpensive to collect as there is no requirement to deploy and retrieve representative targets. For this reason existing background databases

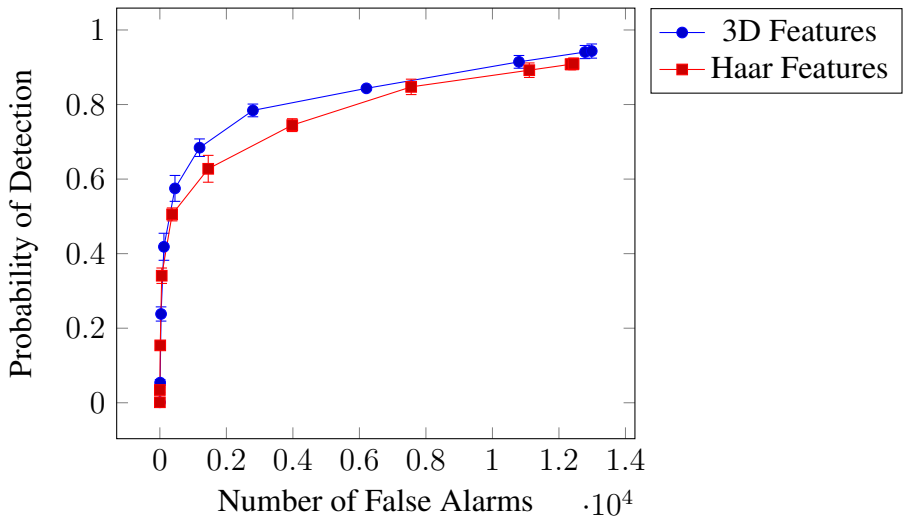


Figure 4.22: Comparison of the performance of the 3D features algorithm and the Haar algorithm trained on cylindrical targets in Marine Sonic data and tested on Edgetech data

could be used to retrain the cascade or alternatively additional stages could be added to an already trained cascade.

In figure 4.22 we compare the performance of the Haar features and the 3D features when they are trained on cylinders in Marine Sonic images and tested on Edgetech images. The first observation is that for both methods the false alarm density has increased by an order of magnitude over the results in figure 4.18. This can largely be attributed to the similarity between areas of the sea-floor in the test Edgetech images (figure 4.24) and the end on cylinders used to train the classifier. The second observation is that both the Haar features and the 3D features detect greater than 90% of the targets in the Edgetech images. This is actually slightly higher than the classifiers trained on the Edgetech data.

In chapter 4 we demonstrated that a Haar classifier trained to detect truncated cones in Marine Sonic data performs very poorly in Edgetech data. Therefore, it is surprising that for cylinders the Haar classifier performs almost as well as the 3D features. The key difference between the cylinder and the truncated cone is that asymmetry of the cylinder results in a large variation in object appearance in the training data set. As the Haar features are not invariant with respect to rotation each feature has a large range of possible values that are 'object like'. This explains both the increase in false alarms when compared to a symmetric target such as the truncated cone and also the good detection of cylinders when the classifier is tested on a different sonar type. However, this is not true when we run the experiment in the opposite direction.

The performance classifiers are now trained on cylinders in Edgetech images and tested on Marine Sonic images. In this case the Haar features demonstrate an improvement of between 20% - 40% in the false alarm density compared to the 3D features. However, the 3D features demonstrate a maximum PD of 82% compared to the Haar features which only detect 72% of the targets. The low PD for the 3D features algorithm

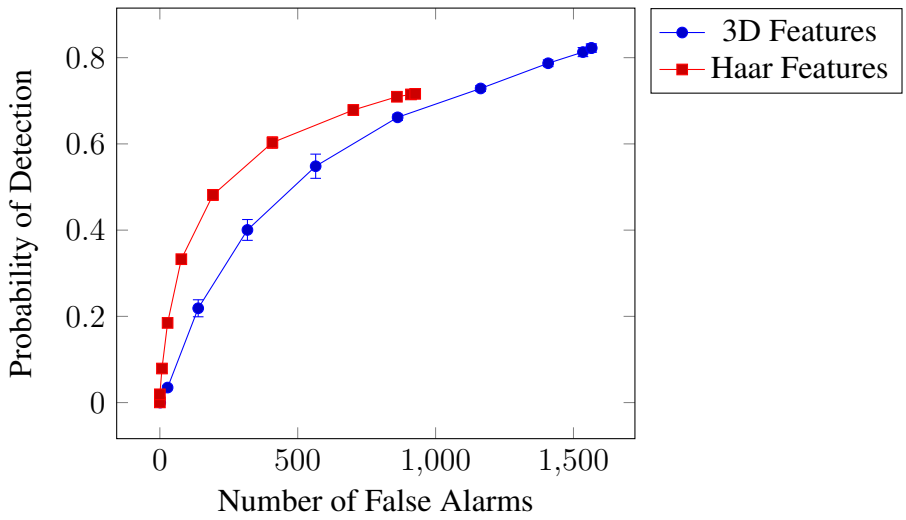


Figure 4.23: Comparison of the performance of the 3D features algorithm and the Haar algorithm trained on cylindrical targets in Edgetech data and tested on Marine Sonic data

can again be attributed to the difficult textures in figure 4.24. By rejecting this texture the classifier also rejects the majority of the end on targets in the Marine Sonic images.

Finally we replicated the experiment that we introduced in chapter 4 where a truncated cone classifier is trained on Marine Sonic images and tested on Edgetech images. However, this time we compare both feature types. The results are shown in figure 4.25. The false alarm density for 3D features agree to within error with the Haar features. However, the maximum PD is increased from 63% to 93%.

In this section we have compared the performance of the 3D features when they are trained using one sonar type and tested on another. We return to the question posed at the beginning of the chapter: can we design a classifier that allows us to re-use a target database when we move to a new sonar type? The first factor is the PD achievable with the 3D features when we change the sonar type. If the features are independent of the sonar parameters then this figure should be comparable. The second factor is the PFA on the new sonar type, ideally this should be independent of the sonar type.

In all cases we see an improvement in the maximum PD for the 3D features of between 12% and 30% compared to the Haar features. In 2 of the 3 experiments the maximum PD achieved was greater than 90%. This is comparable to the results where the classifier is trained and tested on the same sonar type. When the classifier was trained on Edgetech cylinders and tested on Marine Sonic data the PD is only 82%. However, this can be attributed to the low along-track resolution for both sonar types. End-on cylinders are only 1 pixel wide in the along track direction and a large proportion of textures look very similar to the end-on cylinders. As a result all of the end-on cylinders were rejected by the Edgetech classifier during training. The PD of 82% in the Marine Sonic images only represents a 6% decrease from the PD of 88% when the algorithm is tested on Edgetech data.

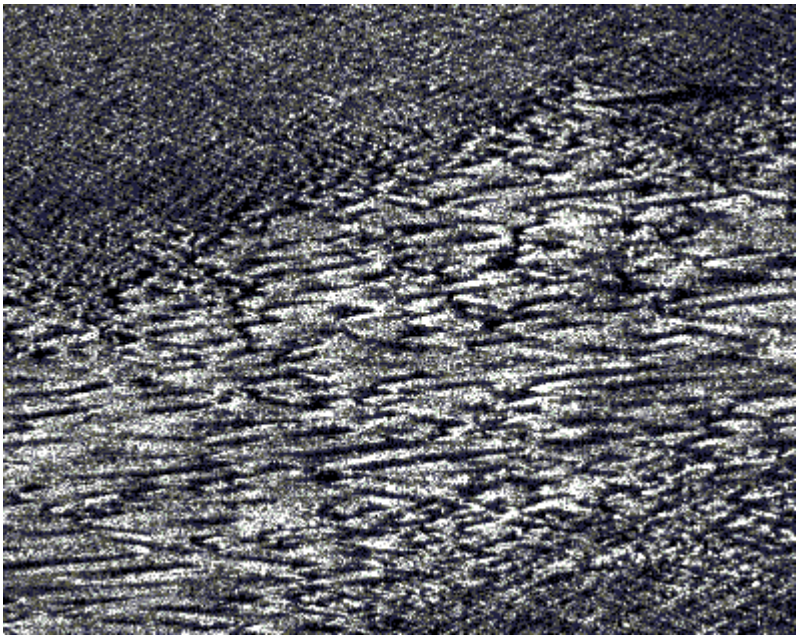


Figure 4.24: Example of a sea-floor region in an Edgetech image which mimics the appearance of end on cylinders

The PFA is comparable for the Haar features and the 3D features but is unpredictable between sonar types. For example a large number of false alarms were produced from a single texture that was only found in the Edgetech data set. However, this problem is caused by the change in the environment rather than the change in sonar type. There are several approaches to mitigating this problem, the first is to create a large database of background images for the purposes of training that include images from several different sonar types and environments. The second approach is to retrain the algorithm using specific background images from the mission of interest. We have performed a number of preliminary tests which suggest that we can achieve a large reduction in the number of false alarms simply by retraining the algorithm on the 5 most difficult images in the mission.

In figure 4.26 we demonstrate the first approach, where a cascade trained entirely on Marine Sonic data for 20 stages has an additional 10 stages trained with the same Marine Sonic targets and the addition of an EdgeTech background set. Clearly the classifier trained entirely on Edgetech data has significantly better performance with between 20% and 50% of the false alarms. However, we have reduced the number of false alarms by an order of magnitude with respect to the classifier training only on Marine Sonic data shown in figure 4.22. In the event that there were no Edgetech data available for training, the classifier trained on the combined data sets achieves reasonable performance.

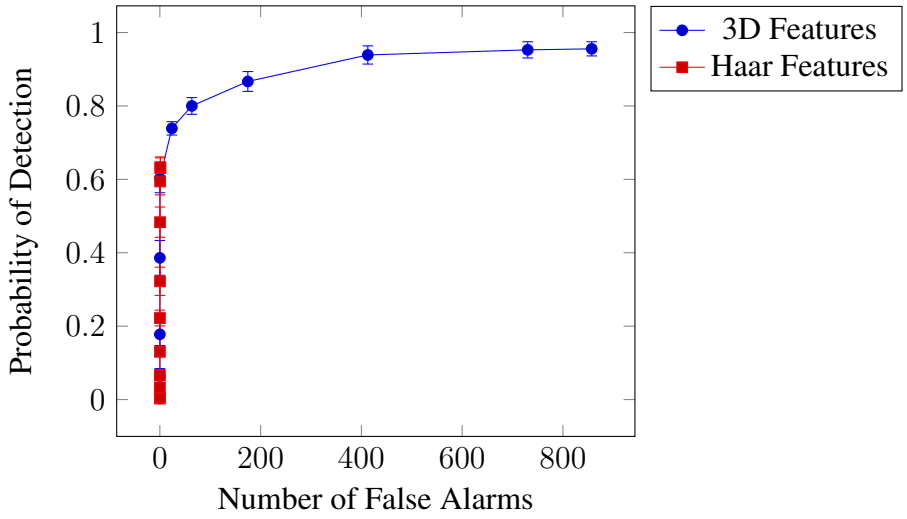


Figure 4.25: Comparison of the performance of the 3D features algorithm and the Haar algorithm trained on truncated cone targets in Marine Sonic data. The algorithms are tested on Edgetech data

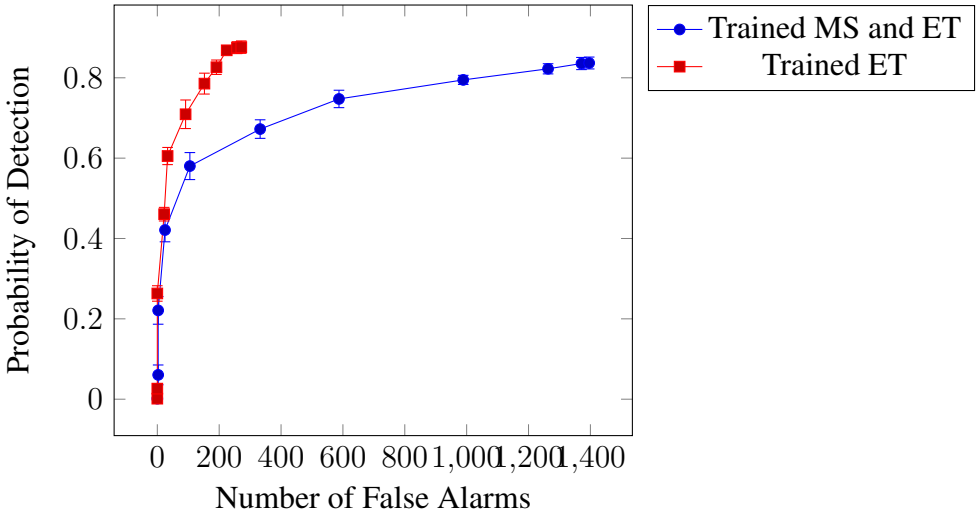


Figure 4.26: Comparison of the performance of the 3D features algorithm. The 'Trained MS and ET' algorithm was trained to 20 stages with only Marine Sonic data. An additional 10 stages were trained with Marine Sonic targets and Edgetech background.

4.12 Conclusion

In this chapter we have demonstrated a novel algorithm for mine-like object detection in SSS. In the introduction we described the problem of ATR algorithms which perform very well in certain conditions but are very sensitive to small changes in the EOCs. Specifically we introduced the problems caused by changing the model of sonar and the costs involved with collecting additional representative images of targets for training. In section 4.3 we demonstrated these problems for one of the current state of the art ATR algorithms in SSS images, the Haar-Cascade. In addition we highlighted the requirement for very large training databases and commented on the inability of the algorithm to model the change in appearance of objects under rotation.

We identified that one of the problems with the current generation of filter based ATR algorithms is that they look for 2D patterns in data generated from a 3D world. As a result the classifiers must learn the representation of the target object under different EOCs and rotations. Consequently, these classifier are complex and prone to over-training. More importantly, if the appearance of the target is significantly different in a new sonar type then the classifier will not recognise the object.

To address this problem we introduced an image formation model that we use to generate 2D filters from 3D objects. Filters are adapted to the local EOCs simply by re-projecting the 3D objects under the new set of conditions. Unlike template matching algorithms we do not have to know the appearance of the object a priori. Instead we test the response of the image to templates generated from a large set of simple cubic objects. One of our key contributions in this chapter was to develop the set of integer-skewed integral images to accelerate the calculation of these templates. In this way we can calculate the millions of template responses required to analyse an image faster than the data acquisition speed.

Next we performed a detailed comparison of the performance of the Haar features with the 3D features. First we demonstrated, in the idealised case, that the 3D features show a significant reduction in the variance of the feature response with respect to orientation. The trained cascades using the 3D features contained approximately half the number of stump decision-trees compared to the Haar features. This suggests that the information gain from the 3D features is greater than the Haar features. Secondly, the performance on the asymmetric object that we tested was significantly better than that of the Haar features. However, for the symmetric objects the performance of the 3D features was worse than the Haar features. This is because the rotational invariance of the 3D features does not have any advantage for the classification of an object that does not change appearance under rotation. Additionally, as the most target like response is always selected over all rotations there is a much greater probability that background will appear to be target-like when viewed from a certain direction.

In general, we concluded that the performance of the 3D features on training and testing data with similar EOCs was comparable. However, the significant increase in computation time for the 3D features (approximately 20 times) would most likely rule out their use in this scenario. There are some advantages in the number of training samples required to prevent over-training, we demonstrated that the 3D features required approximately half the number of target objects compare to the Haar-Cascade. Additionally, in some cases the small improvement in PD for asymmetric objects may also be useful. However, more significant gains in performance are observed when the algorithms are trained on one model of sonar and tested on another.

The important factor when changing the model of sonar is the maximum PD achieved by the algorithm. As we did not have multi-sonar data of the same region of sea-floor, the sea-floor textures are responsible for a large number of false alarms. However, background data is relatively inexpensive to acquire and could even be obtained as part of an Mine Counter-Measure (MCM) mission. Therefore, it is feasible to retrain an ATR algorithm to reduce the PFA simply by providing new background data. This brings us back to why PD is important. It is not possible to increase the PD without new target data for training. For both the objects we tested, and for both sonar types, the 3D features consistently achieved a higher PD than the Haar features. This was most notable for the truncated cone shape where the 3D features achieved a PD 30% higher than the Haar features. The key result in this section was that when we combined the background data-sets the 3D features cascade trained only on Marine Sonic targets achieved reasonable performance over the Edgetech testing set. In future work we aim to find out what proportion of new to old targets is required to match the performance of the cascade trained entirely on new data.

In conclusion, we have developed a novel ATR algorithm that matches the performance of state of the art ATR algorithms while also demonstrating an improved tolerance with respect to the EOCs. In this chapter we tested that performance with respect to the model of sonar. We showed that the algorithm allows us to train a classifier for a new model of sonar using background and target data collected with a different model of sonar. This result is important as it brings us one step closer to a practical ATR system that has state of the art performance and can be transitioned to a new model of sonar without the prohibitive costs associated with data collection.

Chapter 5

Sea-floor Characterisation

5.1 Introduction

In this chapter we introduce a new approach to sea-floor characterisation. This information is useful both for autonomous Mine Counter-Measure (MCM) missions and operator review of Sidescan Sonar (SSS) data. Information about the sea-floor environment is essential for rapid decision making in MCM missions. However, existing techniques are computationally intensive or require in-situ training by a skilled operator. In this chapter we identify characteristics of the sea-floor that are relevant to MCM operations. We then seek to find efficient computational representations of these characteristics that are suitable for low performance computers typically found on-board Autonomous Underwater Vehicles (AUVs).

We start by looking at the motivation for using sea-floor information in MCM missions. In general a technology is useful if it reduces the time required for an MCM mission or reduces the risk to assets passing through a cleared region. First we consider the applications for sea-floor information to assist operators and tactical decision makers in traditional MCM missions.

MCM operators take longer to process SSS data in cluttered and rippled areas of the sea-floor. With accurate sea-floor information it is possible to reduce the impact of these difficult regions. One of the simplest applications is to improve estimations of the time required to review a SSS data set. This information can be used to inform higher level tactical decisions. Alternatively, if the mission objective is to clear a path for assets to transit through an area then sea-floor information can be used to identify the path of least resistance. Combined with software such as SeeByte's SeeTrack 4 [2] operators can choose to only review the images in these regions. This significantly reduces the time required to review the data. At a higher level, the sea-floor information can contribute to tactical decision making. For example, the sea-floor information can be used to estimate the increased risk to assets from mines hidden in cluttered areas.

While sea-floor information is useful for traditional MCM missions it is even more powerful when the survey vehicles can react in real time to the sea-floor information. MCM surveys are typically conducted by a few vehicles equipped with SSS. In a standard operation, each vehicle is assigned a non-overlapping region of the sea-floor to survey. Following recovery of the vehicles, a human operator reviews the data to identify potential targets. Possible targets can then be reacquired by another vehicle to obtain more detailed video footage. However, the data analysis and reacquisition of targets is a time consuming process.

The Neptune system, developed at SeeByte Ltd., aims to increase the tempo of MCM missions using a fleet of cooperating AUVs. When a survey vehicle detects a target using the on-board Automatic Target Recognition (ATR) algorithms, it assigns another vehicle to perform a closer inspection using Forward Looking Sonar (FLS) or video. In flat sea-floor regions this saves time as both sonar and video imagery are available for the operator to review when the vehicles are retrieved. However, in rippled regions or areas with a high clutter density the vehicle can rapidly identify too many targets for reacquisition.

In these regions a number of behaviours are possible. In regions with regular ripples the vehicle can run a survey orthogonal to the ripple direction. This reduces the visibility of the ripples, preventing false alarms. In complex areas the simplest approach is to avoid the region entirely. However, the complexity of the regions also guarantees that there will be sufficient features to fuse multiple views of the same target. Therefore, a vehicle could be tasked to create a video, or FLS, mosaic of the entire region. Alternatively, the survey vehicle could run additional SSS survey legs to create data for multi-view fusion.

In this chapter we use a wavelet decomposition of the sea-floor to extract information about the sea-floor. Unlike existing techniques we calculate values from the wavelet coefficients that relate directly to the types of sea-floor that are relevant to MCM missions. We refer to these values as sea-floor characteristics. Using these sea-floor characteristics we can perform a fast and accurate segmentation of the sea-floor into regions with homogeneous characteristics. The classes of sea-floor that we aim to identify are flat, rippled, complex and cluttered. However, wavelet analysis does not give us an accurate estimation of the mine-like clutter density. Therefore, using the image segmentation work in chapter 2 we identify the density of mine-like objects within an area of the sea-floor.

The chapter is organised as follows. In section 5.2 we discuss the calculation of textural characteristics from sonar images. Three textural characteristics are introduced and the distribution of these features with respect to sea-floor types and sonar types is explored. In section 5.5 we develop a segmentation algorithm to group homogeneous regions of the sea-floor. Some regions are not well described by the textural characteristics. For example a flat region may contain a large number of Mine Like Objects (MLOs). Therefore, in section 5.6 we introduce a simple approach to estimating the clutter density. Finally in section 6.2 we present our key results and conclusions from the chapter.

5.2 Textural Characteristics

In this section we discuss the characterisation of sea-floor textures using wavelet analysis. Texture analysis is the process of extracting features that describe a texture and labelling homogeneous regions, based on the extracted features. It is complicated by the range of patterns that are considered to be textures. This problem is demonstrated, for sonar images, in figure 5.1. The figure shows a SSS image with textures that we have manually classified as Flat, Ripple, Complex and Mixed.

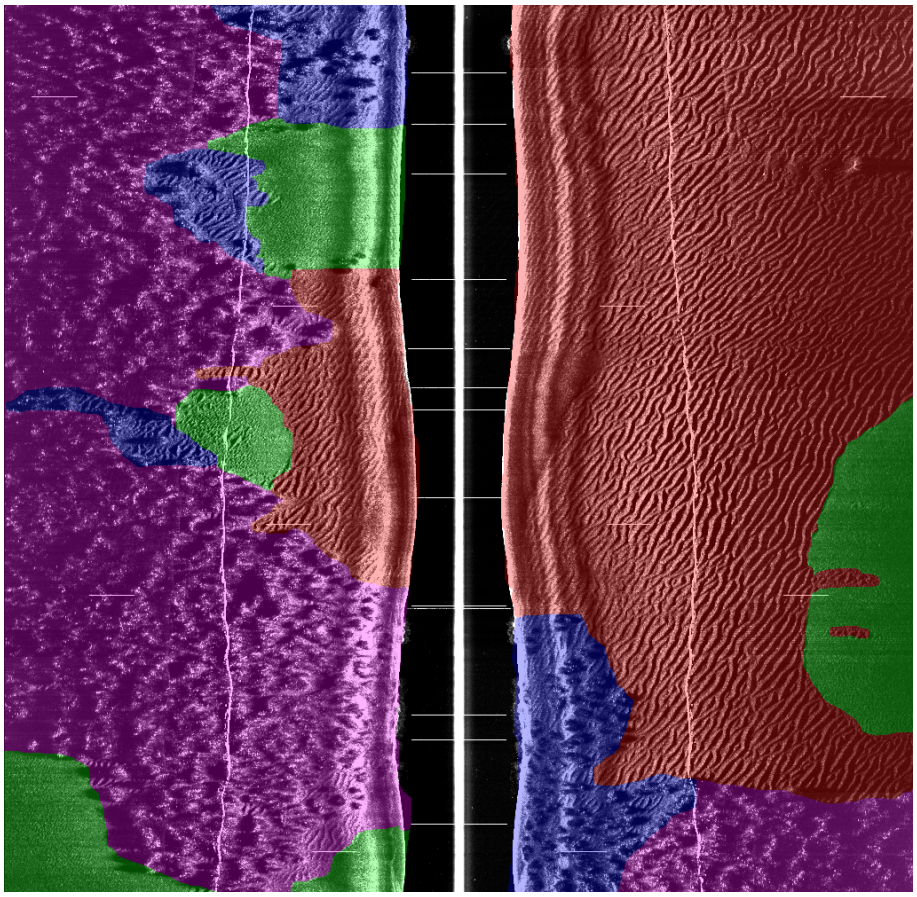


Figure 5.1: Manual segmentation of a SSS image. Flat = Green, Ripples = Red, Complex = Magenta, Mixed = Blue

The flat regions are described by a single statistical distribution, either a Rayleigh or more accurately a K-distribution. This is an example of a stochastic texture; the region can be described by a single statistical distribution [89]. The rippled and complex regions do not demonstrate a single underlying statistical distribution. Instead there are small regions, or cells, that demonstrate local homogeneity. The large scale distribution of cells can, however, be described by a statistical distribution. This type of texture is referred to as a structural texture and is described in an early paper by Haralick[90]. The two texture definitions that will be referred to in the rest of this chapter are

Definition 5.1. Structural texture - A texture that can be decomposed into primitives displaying some ordered spatial relationship

Definition 5.2. Stochastic texture - A texture that can be generated by some underlying statistical distribution

This section is organised as follows. In section 5.2.1 we introduce the topic of texture analysis and review the existing literature. The difficulty in finding a single characteristic to describe a sea-floor type is illustrated using the Gray Level Co-occurrence Matrix (GLCM). We then give a broad overview of a number of wavelet and power spectral approaches to analysing sea-floor textures.

In section 5.4 we extend the wavelet-based approaches discussed in section 5.2.1 and introduce three characteristics to identify flat, ripple and complex sea-floor types. In section 5.4.3 we analyse the performance of the textural characteristics with respect to different sea-floor and sonar types. This leads to the segmentation approach which is described in section 5.5. Finally we present the shortcomings of the approach which are addresses in section 5.6.

5.2.1 Textural Analysis in Sonar Images

The aim of the work in this chapter is to find a robust, computationally efficient approach to characterise and segment rippled and complex regions of the sea-floor. The current approach to labelling different sea-floor types is to use a supervised sea-floor classification algorithm. Historically, these algorithms have been used to identify sea-floor types for geological mapping and surveys [91, 92, 93, 94]. Early algorithms were severely limited by computational power and the poor resolution of the sonar. However, more recent approaches with high resolution sonar [95, 96, 97] demonstrate excellent classification accuracy. There are several examples of these algorithms being used to analyse MCM missions [23, 98, 99]. However, they have all been used for post-processing rather than on-board processing with an AUV.

There are a number of challenges integrating existing sea-floor classification algorithms with the decision making algorithms running on-board an AUV. Supervised approaches such as [23] train a classifier using a 'large bag of features'. These classifiers are excellent at classifying a large mission, given a limited training sample from within the same mission. However, specific sea-floor classes such as ripples and rocks vary in appearance between different regions. Indeed, they often vary with different views of the same region. Therefore, a supervised classifier must be trained on a representative sample of the sea-floor types. For MCM missions this is not always practical as there is not time to survey the area to provide sea-floor training samples. Unsupervised algorithms such as [69] can segment previously unseen textural regions, however a semantic label still needs

to be applied to the region. Additionally, the computational complexity of unsupervised algorithms can be a problem for the limited computational power on-board an AUV.

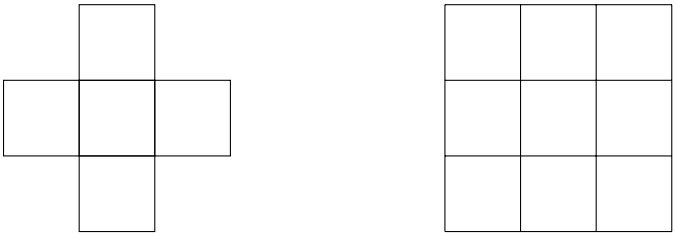
Classification of stochastic textures, such as sediment, in sonar images is a well studied problem. Sea-floor regions are classified according to the reflectivity and statistical properties of the sediment [95, 96, 100]. If the sea-floor type is known then sediment in sonar images can be described very accurately by a K-distribution[96]. The K-distribution is a probability distribution particular to radar and sonar imagery. It is derived from the Rayleigh distribution, which describes the sonar speckle, and a distribution representing the sonar cross section. If the original sonar image, before pre-processing, is available then the parameters of the K-distribution can be estimated[101] and compared to a database of known sediment types. In practice the original sonar data is not always available and manual classification may be required. In this chapter we do not consider sediment analysis. However, it would be relatively simple to include this information in the future.

Textural analysis in sonar images has largely focussed on two classes of textural descriptors. The GLCM [92, 97, 102] described in section 5.2.1 and wavelet domain and fractal approaches [18, 23, 69, 98, 103, 104, 105, 106, 107] described in section 5.2.1. Both approaches are computationally inexpensive and good segmentation results can be achieved using a supervised classifier. In this section we explore these approaches in detail and evaluate their suitability for the applications presented in the introduction to this chapter.

Grey Level Co-occurrence Matrix

The GLCM, introduced by Haralick [108], is a simple example of a stochastic texture measure. The GLCM characterises the first order statistics of a texture with respect to the grey level of neighbouring pixels. The GLCM is used for the classification of SSS and multibeam sonar by Blondel et al. [92, 97] and for multibeam sonar by Lubniewski et al. [102]. In this section we describe the calculation of the GLCM and use it to calculate feature distributions for several sea-floor textures.

First a connected neighbourhood is defined; typically this will be a 4 or 8 connected neighbourhood (figure 5.2). A GLCM is generated for each possible direction defined by the neighbourhood. For the 4 connected neighbourhood this will be north, east, south and west. The GLCM is a 2D histogram of the frequency with which neighbouring pairs of grey level values occur over the image. A co-occurrence matrix P is defined over an $n \times m$ image I, parametrised by an offset $(\Delta x, \Delta y)$, as:



(a) 4-Connected (b) 8-Connected

Figure 5.2: 4-connected and 8-connected neighbourhood for a pixel

$$P_{\Delta x,\Delta y}(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, & \text{if } I(p,q)=i \text{ and } I(p+\Delta x, q+\Delta y)=j\\ 0, & \text{otherwise}\end{cases} \tag{5.1}$$

For the standard 256 grey levels the matrix will be sparse and relatively uninformative. Therefore, the number of grey levels is typically quantised to a smaller value dependent upon the problem space. Values of 8 and 16 grey levels are commonly used. The GLCM approach assumes that a texture can be uniquely described by the intensity differences between neighbouring pixels. Textures are rarely homogeneous in this respect and artefacts will be introduced by the windowing process. Therefore, a number of more general textural characteristics are derived. In the original paper there are 14 such measures, here 4 of the most commonly used features are discussed.

We first define the normalised co-occurrence matrix . A normalised co-occurrence matrix p is defined by dividing P by the number of neighbouring pixels R in the window.

$$p(i,j) = P(i,j)/R \tag{5.2}$$

Two marginal probability matrices are also defined by summing the rows and columns of the GLCM

$$p_x(i) = \sum_{j=1} p(i,j) \tag{5.3}$$

$$p_y(i) = \sum_{i=1} p(i,j) \tag{5.4}$$

These matrices have associated mean ν_x, ν_y and standard deviation σ_x, σ_y.

Definition 5.3. Contrast - The contrast of the GLCM is a measure of the deviation of the $p(i,j)$ from a diagonal matrix. An image with large intensity changes between neighbouring pixels will have a high contrast.

$$\sum_{i,j} |i-j|^2 p(i,j) \tag{5.5}$$

Definition 5.4. Correlation - The correlation of the GLCM is a measure of how well correlated a pixel is to it's neighbours over the image

$$\sum_{i,j} \frac{ij \times p(i,j) - \nu_x \nu_y}{\sigma_x \sigma_y} \tag{5.6}$$

Definition 5.5. Energy - The energy, or uniformity is a measure of the sum squared elements of the GLCM

$$\sum_{i,j} p(i,j)^2 \tag{5.7}$$

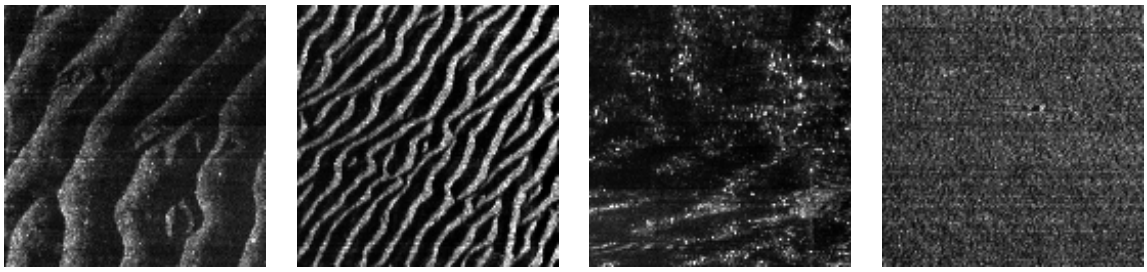
Definition 5.6. Homogeneity - The homogeneity is a measure of the closeness of the elements of the GLCM to the diagonal

$$\sum_{i,j} \frac{p(i,j)}{1+|i-j|} \tag{5.8}$$

An example of these features is shown for four different textures sampled from a Marine Sonic SSS survey (5.3). In this example only the horizontal features are calculated $\Delta x = 1, \Delta y = 0$. The GLCM is calculated with 4 gray levels over a 64 by 64 window. The features are calculated for all window positions in the texture samples. The distribution of the feature values is approximated as a normal distribution and the results are shown in figure 5.4.

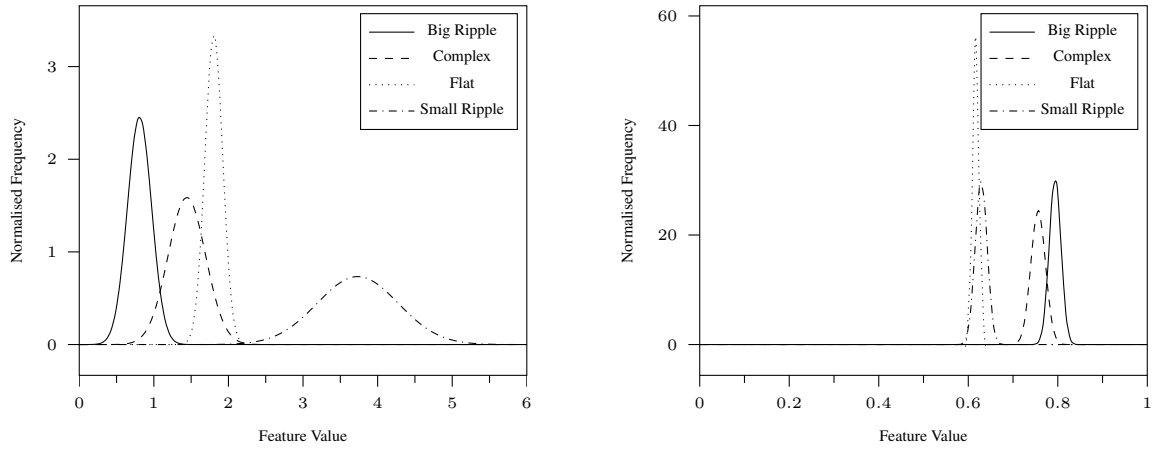
While there is clearly enough information in the GLCM features to separate the example textures, real world systems require a larger number of features and analysis at several scales. In practice the GLCM is calculated at a $0°, 45°, 90°, 135°$ angle and all 14 features are calculated. If a multi-scale approach is used then it is easy to generate hundreds of features. The complexity is typically reduced by applying Principle Component Analysis (PCA) to the data before classification. Therefore, while good separation of classes is achieved with the GLCM it is difficult to relate the features to the properties of the classes that would be recognised by a human operator. This is demonstrated in the example texture set where the distribution of features for big ripples and small ripples only overlaps in the Correlation feature space.

In the next three sections we review wavelet analysis, fractional dimension analysis and other wavelet based approaches. With the notable exception of the fractal dimension approach of Nelson et al. [44] and the wavelet based ripple detection of Williams et al. [109] all of the approaches suffer from a similar problem. While the bag of features is informative, individual features do not correlate to a particular sea-floor type. In the next section we review wavelet analysis.

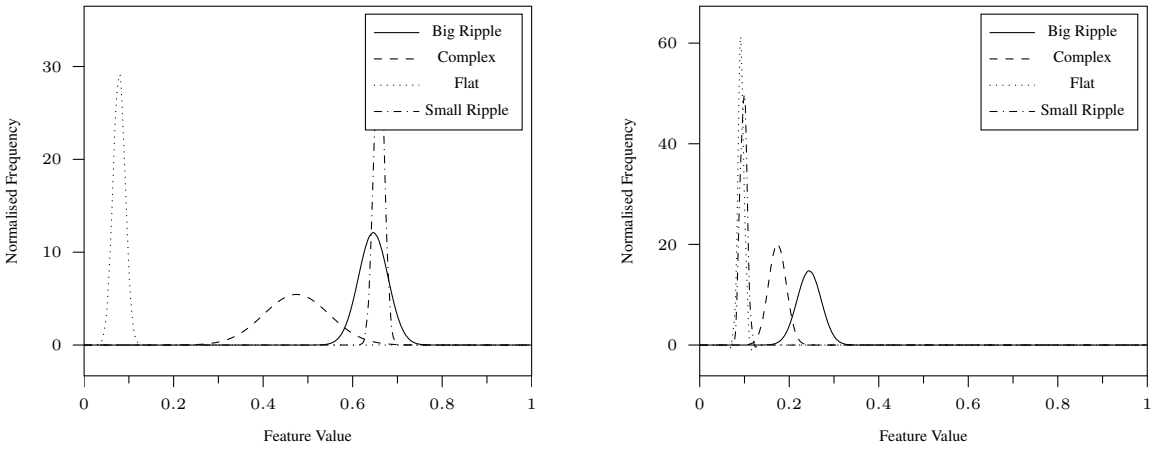


(a) Large Ripples (b) Small Ripples (c) Complex (d) Flat

Figure 5.3: Examples of several different sea-floor textures, recorded with a Marine Sonic SSS



(a) Contrast (b) Homogeneity



(c) Correlation (d) Energy

Figure 5.4: Features calculated from an GLCM over 4 different sea-floor types

Wavelet analysis

In wavelet analysis, an image is decomposed into the sum of oscillating signals with a specific frequency, direction and spatial extent. These 'brief oscillations' are referred to as wavelets. Typically a number of wavelets, with different frequencies and directions, are convolved with an image. As such, wavelet analysis provides information about the image at multiple frequencies and directions. An introduction to wavelet theory can be found in appendix 5.3. In this section we explain how textural features can be obtained from a wavelet decomposition and review a number of papers using this approach.

The 2D Discrete Wavelet Transform (DWT) measures the properties of a texture at a position $\mathbf{x} \in \mathbb{R}^2$ angle $\phi \in \{\phi_1, \phi_2, ..., \phi_d\}$ and scale $j \in \{1, 2, ..., s\}$. Each pixel in the image is therefore described by a feature vector consisting of $d \times s$ features. Typically the image is split into tiles and a single feature vector is composited from all of the feature vectors within a tile. The main challenge in wavelet analysis of textures is therefore one of feature reduction. This is addressed for unsupervised segmentation in a number of papers [18, 69, 98].

Standard clustering approaches, such as k-means, fail due to the sparsity of the data in high-dimensional spaces. A number of different techniques are applied to reduce the dimensionality of the data. Williams et. al [98] apply spectral clustering to the problem. Karoui et al. [69] use the KullbackLeibler divergence between feature vectors as an input to an Maximum Marginal Probability (MMP) and a level set segmentation algorithm. These techniques are discussed in more detail in the section 5.5.

These approaches will always segment the image into k-classes even if the segments are visually similar. Therefore, a segment and merge scheme would be required to create a fully autonomous system. Unlike the other two approaches Mignotte et al. [18] reduce the dimensionality of the wavelet features using a supervised classifier. A Markov Random Field (MRF) based fusion scheme is used to smooth the resulting classification. This approach is unsuitable for autonomous sea-floor classification as the high dimensionality of the feature vector makes it very likely that the algorithm will specialise on a single data set. For this reason the algorithm is re-trained for each mission. In the next section we see how multiple wavelet coefficients can be combined to calculate a single fractal dimension.

Fractal dimension

The fractal dimension of a texture can be interpreted as measure of how the energy is distributed with respect to scale. The fractal dimension is based on the assumption of self-similarity in a texture. That is $f : \mathbb{R}^2 \mapsto \mathbb{R}$, such that

$$\mathbb{E}[f(\gamma x)f(\gamma \zeta)] = \gamma^{2H}\mathbb{E}[f(x)f(\zeta)] \tag{5.9}$$

where $H \in [0,1]$ for some $\gamma \in \mathbb{R}$. Self-similarity implies that the spectral energy is a function of the scale at which the texture is observed. This property is true for fractional Brownian surfaces which have been used to model the sea-floor [110].

In [44] the Dual Tree Complex Wavelet Transform (DT$\mathbb{C}$WT) is used to estimate the fractal dimension of rippled and flat sea-floor regions. The fractal dimension can be calculated using the wavelet transform Wf of an image $f : \mathbb{R}^2 \mapsto \mathbb{R}$ at spatial location $\boldsymbol{x}$:

$$(Wf)(k, m; \boldsymbol{x}) = 2^{-k} \int_{\mathbb{R}^2} f(\boldsymbol{\epsilon}\overline{\psi_m(2^{-k}(\boldsymbol{x} - \boldsymbol{\epsilon}))}) d\boldsymbol{\epsilon} \tag{5.10}$$

where ψ_m is a mother wavelet with orientation m and scale k.

Nelson and Kingsbury [111] show that the following relationship holds for statistically self similar processes:

$$\log_2 \mathbb{E}[|(Wf)(k, m; .)|^2] = 2k(H + 1) + C_{\psi_m, f} \tag{5.11}$$

where $C_{\psi_m,f}$ is independent of scale k and $2k(H + 1)$ is a linear function of k. Therefore, the exponent H, which approximates the fractal dimension, can be calculated from the average slope of all the directional sub-bands.

Using this relationship, rippled regions are detected by observing a departure from the power spectra observed for the flat regions. In [112] it is shown that sea-floor images can be classified from their multi-fractal dimension. The idea is to approximate a power spectra about a number of frequencies and calculate the fractal dimension at each point in the spectrum. While this gives an informative texture measure it also suffers from the curse of dimensionality.

Other Approaches

There are a number of other approaches to texture classification in SSS images. Several are based on variations of the wavelet transform. The contourlet transform used in [113] introduces greater directional selectivity to the wavelet transform through the use of multiple directional filter banks. Javidan et al. apply the non subsampled variant of this approach [114]. This results in a shift-invariant, multi-scale, multi-resolution view of the sea-floor. The mean energy and variance of the contourlet transform is calculated for each scale and direction over tiles in the image. Tiles are then classified using the euclidean distance from a training sample. The performance of the contourlet based segmentation is compared using the same framework to the standard 2D wavelet transform. In their analysis they conclude that the contourlet transform is superior to the wavelet transform. However, the wavelet transform is only applied over 2 scales and the sample size is limited to a single image.

A further variation on the wavelet transform, scattering operators, is applied to Syn-

thetic Aperture Sonar (SAS) images by Valeyrie et al. [115]. Scattering operators are used to improve the stability of the wavelet transform with respect to local deformations. Wavelet transforms are stable with respect to local deformations which are much smaller than the scale of the wavelet. Consequently high frequency information in the signal is particularly susceptible to deformations. The scattering operators work by transforming high frequency information into the low frequency domain and then analysing this signal to generate a feature vector. The full mathematical approach is presented in [116], however a summary is provided in the next 2 paragraphs for interested readers.

Scattering operators are implemented as a cascade of wavelet filters and modulus operators. Consider a wavelet filter in the frequency domain. If we take the Shannon wavelet then the wavelet will act as a rectangular band pass filter or alternatively a delta function convolved with a window. Consider the effect that this has on a signal. All frequencies outside of the band pass filter will be killed and the delta function will shift the remaining frequencies about the zero frequency axis. Taking the modulus of the wavelet response removes the position dependent phase and leads to shift invariance in the output signal. The result of the first bandpass filter and modulus operator is referred to as the first order interferences of the function.

In standard wavelet filtering the first order interferences would be the square root of the energy of the function. Deformation invariance could then be achieved by low pass filtering the wavelet energy over a window. However, the low pass filter effectively removes all of the information held in the high frequency components. Scattering operators retrieve this information by cascading wavelet filters and modulus operators. The second wavelet filter acts on the first order interferences. The first order interferences are high frequencies which have been shifted to the low frequency domain. As the second wavelet filter in the cascade is operating on low frequencies the second order interferences are much less sensitive to deformations of the signal.

The properties of shift invariance and deformation invariances are useful for analysing real textures, where textures are often corrupted by a deformation. Additionally, the retention of second order interferences allows the classification of textures that have identical first order interferences; for example Gaussian and Bernoulli white noise. State of the art performance has been demonstrated for scattering operators over a range of classification tasks. However, for the purpose of characterising the sea-floor the extra information provided by the scattering transform is offset by the increase in the size of the feature space and the additional computational complexity of the algorithm.

Summary

In this section we have reviewed GLCMs, wavelet filters, fractal dimensions, countourlets and scattering operators. Of these techniques only the GLCM operates in the spatial do-

main. The other techniques operate in the frequency domain and are based on wavelet theory. Interestingly, the GLCM can be reformulated using Haar wavelets [117]. Therefore, analogues of the GLCM features could be derived using any wavelet basis.

To date there has not been a comparative review of the different textural descriptors on sea-floor textures. For classification accuracy on general texture samples the doctoral thesis of Bruna [116] suggests that scattering operators outperform general wavelet filters over a large range of classification tasks. Karoui et al. report that the GLCM provides more informative features that then wavelet transform. However, this can be attributed to the preprocessing of the GLCM. The GLCM is preprocessed to generate features such as the homogeneity and contrast while the energy of the wavelet transform is used directly as a feature.

The general approach is to calculate a large number of features and apply clustering or supervised learning to reduce the dimensionality of the feature set. Clustering allows us to segment the data without any user input but says nothing about the characteristics of each segment, beyond the distribution of features. Supervised classification must be retrained for each data set due to variations in the survey altitude and the sea-floor. Only one of the papers reviewed [69] has considered the effect of operational parameters on texture analysis. Here, the grazing angle is compensated for by splitting the images into angular ranges, over which the appearance of a texture is homogeneous. The similarity measure between two texture samples is relaxed for samples that occur in different angular ranges.

While there is a large body of literature on sea-floor classification, there are relatively few papers which combine both sea-floor classification and ATR. Given that the sea-floor type has such a large impact on risk in MCM missions this is a rather large oversight. The problem has been partially addressed by Mignotte et al. [18] and Valeyrie et.al in conference proceedings based on earlier work in [115]. In [18] augmented reality targets are inserted into classified sea-floor regions allowing a performance estimate to be made for the ATR in each region. This approach is a powerful post-processing tool but requires operator input for the sea-floor classification and is computationally intensive. Valeyrie et al. have demonstrated a completely autonomous solution in which ATR detections are evaluated by their similarity to the surrounding sea-floor. However, this approach does not address the risk of missed detections.

For unsupervised characterisation we require a set of features that are invariant to operational parameters and whose output can be predicted for previously unseen textures. In the author's opinion the most promising approach to characterising the sea-floor has been developed by Nelson et al. [44]. Wavelets provide a computationally efficient algorithm to extract information about the image at multiple scales and directions. The development of wavelet based features that are scale, direction and contrast independent is the major contribution of this chapter. In the next section we describe how these features are calculated.

5.3 Dual Tree Complex Wavelet Transform

Wavelet analysis is used in image processing to extract information about the spatial frequencies in an image. There are many parallels between wavelet and Fourier signal analysis. In-fact we show later that a Fourier transform is simply a special case of the wavelet transform. In this section we assume that the reader is familiar with Fourier transform. We start by exploring the limitations of the Fourier transform, before introducing wavelet theory.

The motivation for using a wavelet transform is to identify the position at which spatial frequencies occur in the image. Consider a 1D signal $f(t)$ consisting of a sin wave with frequency ν multiplied by a Gaussian window with mean 0 and standard deviation σ

$$f(x) = sin(2\pi x/\nu) \times gauss(0, \sigma^2) \tag{5.12}$$

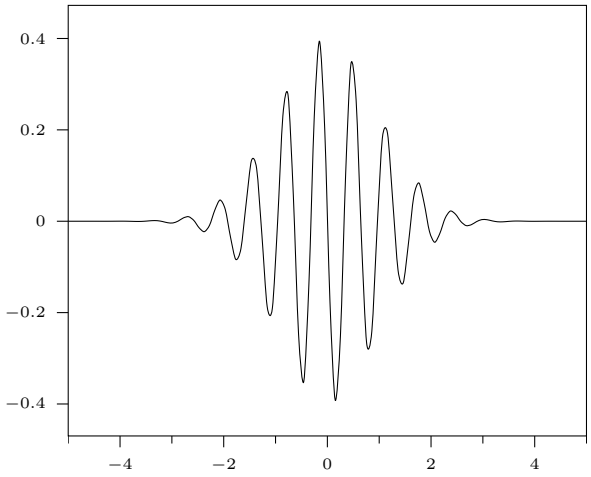


Figure 5.5: $f(x) = sin(2\pi x/\nu) \times gauss(0, \sigma^2)$

The Fourier transform of this function $\hat{f}(\epsilon)$ is a delta function at $\delta(\epsilon \pm \nu)$ convolved with a Gaussian with mean 0 and standard deviation $1/sigma$

$$f(\epsilon) = \delta(\epsilon \pm \nu) * gauss(0, 1/\sigma^2) \tag{5.13}$$

Convolution with a delta function defines a shift on the x-axis therefore $f(\epsilon)$ simplifies to

$$f(\epsilon) = gauss(\pm\nu, 1/\sigma^2) \tag{5.14}$$

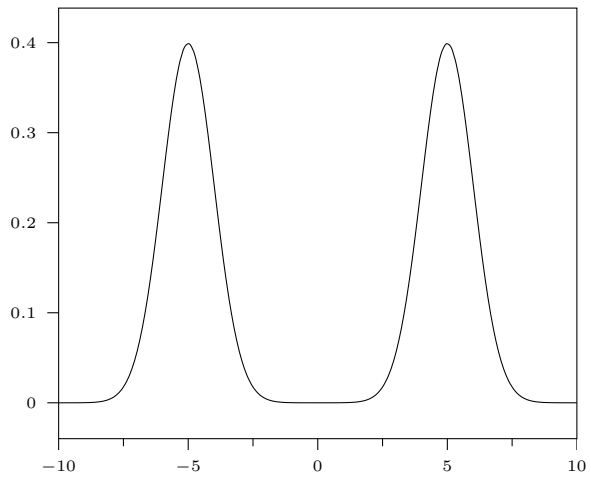


Figure 5.6: $f(\epsilon) = gauss(\pm\nu, 1/\sigma^2)$

The Fourier transform identifies the fundamental frequency of the sine wave but does not contain any information on its spatial distribution within the signal (Observant readers may have noticed that in this example the position of the Gaussian could be reconstructed from the imaginary part of the Fourier transform. However, in general the phase of the input signal would not be known and this information would be lost).

The spatial information can be retrieved by applying the short-time Fourier transform. In the short-time Fourier transform the signal is multiplied by a window function which is non-zero for a short period of time. Or in the case of image analysis, for a short distance. The Fourier transform is applied sequentially as the window is moved along the x-axis. For this example a Gaussian windows with spatial mean mu_w and standard deviation σ_w is used to window the signal. Applying the convolution theorem for two Gaussians the Fourier transform will be

$$f(\epsilon, \mu_w)_{REAL} = gauss(\pm\nu, 1/\sigma^2 + 1/\sigma_w^2) \tag{5.15}$$

It is interesting to see how this affects the uncertainty in the estimate of the frequency ν. The uncertainty in the frequency of the sine wave in the signal $f(x)$ is inversely related to the standard deviation of the Gaussian. As the standard deviation of the Gaussian tends to infinity the uncertainty in the frequency of the sine wave tends to zero. By applying a window to the signal the uncertainty in the frequency ν is increased by $1/\sigma_w^2$. The spatial uncertainty is defined by the standard deviation of the window $sigma_w$. It is therefore not possible to know both the spatial position and frequency with an arbitrary precision. If the spatial precision is increased by using a smaller window then the uncertainty in the frequency is increased. This principle is known as the uncertainty principle of signal processing, which states

$$\Delta x * \Delta\omega \geq \frac{1}{2} \tag{5.16}$$

where x represents distance and ω is the angular frequency ($\omega = 2\pi x$)

In the short time Fourier transform the window size is chosen such that there is a compromise between the uncertainty in the spacial position and spacial frequency. However, the short-time Fourier transform introduces a constant error in spatial frequency over all frequencies. This is not optimal as high frequency components will have a small fractional error while low frequency components will have a large fractional error. The advantage of the wavelet transform over the short-time Fourier transform is that the support of the window scales with the frequency that is being measured. The following paragraph introduces a formal definition of the wavelet and wavelet transform.

A wavelet is a function with some characteristic frequency multiplied by a window with a scale and shift on the x-axis. An orthonormal set of wavelets is generated from a child wavelet $\psi_{j,k}$.

$$\psi_{j,k}(x) = \frac{1}{\sqrt{j}}\psi\left(\frac{x-k}{j}\right) \tag{5.17}$$

where j is positive and defines the scale and k is any real number and defines the shift. The wavelet basis is orthonormal if it satisfies

$$\langle \psi_{jk}, \psi_{lm} \rangle = \delta_{jl}\delta_{km} \tag{5.18}$$

Completeness is satisfied if every function $f \in L^2(\mathbb{R})$ can be expanded in the basis as

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{jk}\psi_{jk}(x) \tag{5.19}$$

where c_{jk} are the wavelet coefficients.

The integral wavelet transform $X(j,k)$ is defined the inner product of a function $f(x)$ with a wavelet ψ_{jk}

$$X(j,k) = \frac{1}{\sqrt{|j|}}\int_{-\infty}^{\infty} f(x)\psi_{jk}^{*}\left(\frac{x-k}{j}\right)\,dx \tag{5.20}$$

We can relate the wavelet transform to the Fourier transform via an appropriate choice of wavelet. If ψ_{jk} is chosen such that $\psi_{jk} = e^{\frac{2\pi i}{k}}$ then the wavelet transform is simply the standard Fourier transform. In practice the continuous wavelet transform is not often used in image analysis, instead the DWT is used.

5.3.1 Discrete Wavelet Transform

The DWT is applicable to signals that are discretely sampled. Typically the discrete wavelet transform is implemented as series of quadrature high pass and low pass filters. The output from the low pass filter provides the input to the next stage of the filter bank.

The DWT for a signal $x[n]$ is implemented as a cascade of high pass $h[n]$ and low pass filters $g[n]$:

$$y_{low}[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k] \tag{5.21}$$

$$y_{high}[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{5.22}$$

The high pass and low pass filter must form a quadrature mirror filter. A quadrature mirror filter has the property that the magnitude of the frequency response of the two filters is symmetric about 1/4 of the sampling frequency (or half the maximum observable frequency). This has the effect of splitting the observable frequency range into a high and low frequency component. As half of the frequency components have been discarded, according to Nyquist's rule, the filter response can be down sampled by a factor of 2. This is represented by the sub sampling operator $\downarrow$.

5.3.2 Cascade of Filter Banks

The DWT can be implemented very efficiently as a cascade of high and low pass filters. The high pass filter contains the detail coefficients (or wavelet response) and the low pass filter removes the high frequencies so that the signal can be down sampled without aliasing. This gives the algorithm an $O(n)$ time complexity. This compares favourably to the short time Fourier transform which has a $O(nlogn)$ time complexity. However, this is not a fundamental property of the DWT but is instead due to the choice of a logarithmic frequency scale. Figure 5.7 shows a 3 level cascade. The associated frequency response of the coefficients is shown in figure 5.8.

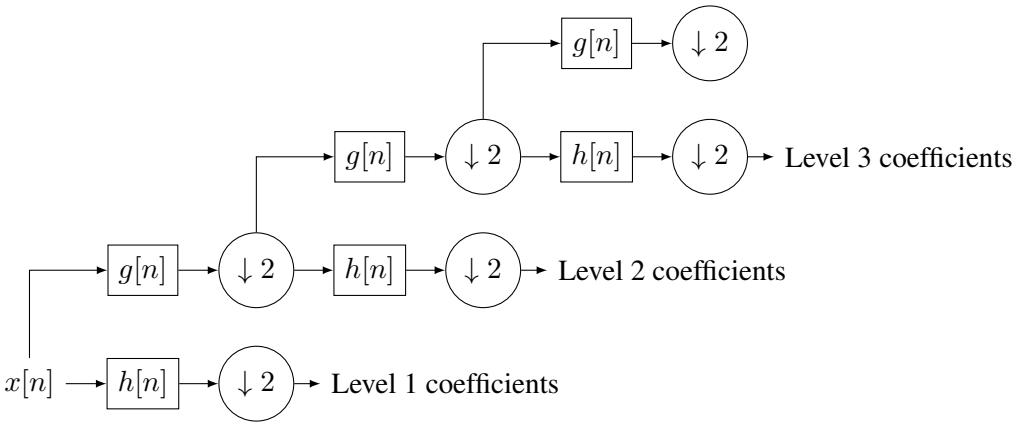


Figure 5.7: Implementation of the DWT.

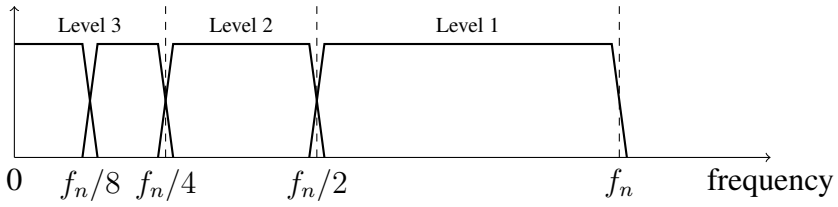


Figure 5.8: Frequency domain representation of the DWT.

5.3.3 2D Discrete Wavelet Transform

The DWT is easily extended to 2 dimensions by applying separable row and column filters. Figure 5.9 shows a single level of the standard 2D DWT. First high $h[n]$ and low pass $g[n]$ filters are applied along the columns of the image and the image is downsampled by a factor of 2. This is then repeated along the rows of the image. The 4 bandpass sub bands are denoted $AB(n)$ where A represents the row filter and B represents the column filter and can be any combination of high pass H or low pass G filters.

The 2D separable DWT results in 2 filters with clearly defined direction and a third filter which forms a chequerboard pattern and has no clear direction (Figure 5.10). The poor directional selectivity of the standard 2D seperable DWT is one of the motivations for using the dual tree wavelet approach.

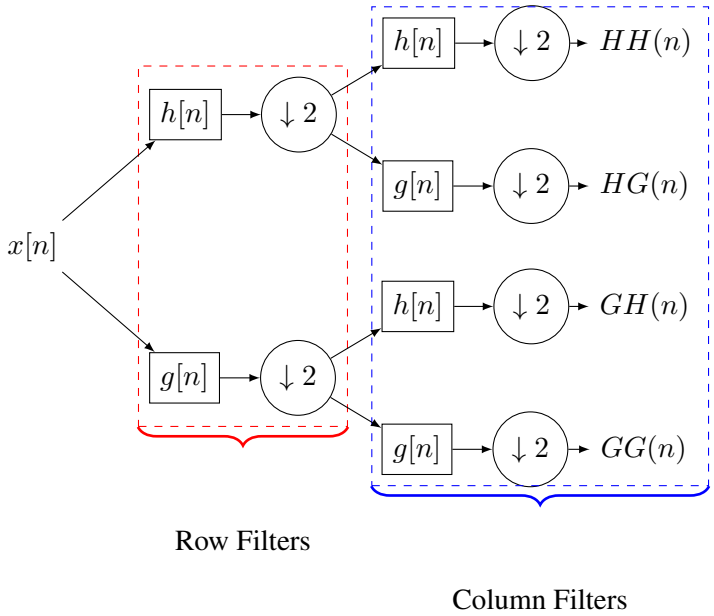


Figure 5.9: Implementation of the 2D DWT.

5.3.4 Aliasing and Shift Variance

Consider the wavelet filters in figure 5.8, the filters are not perfect band pass filters, there is some overlap between the low pass and high pass filter. If the sampling frequency of the

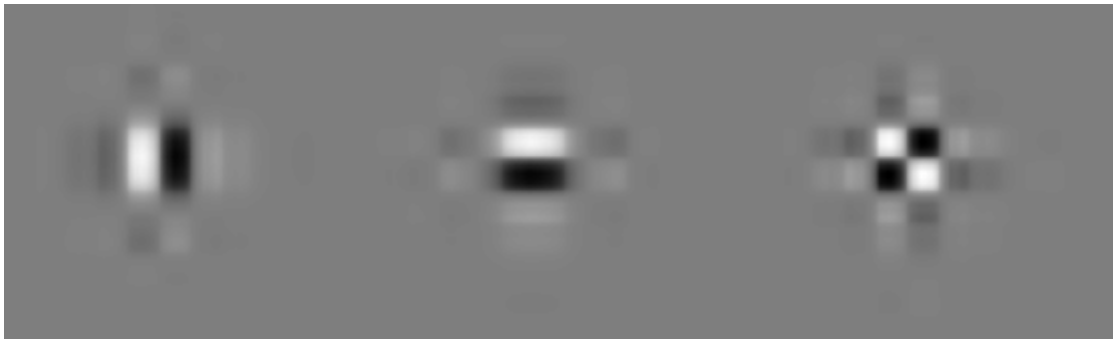


Figure 5.10: Orientation of the 2D wavelet filters.

original signal is F_s then when the signal is down sampled Nyquist's rule will be violated and there will be some aliasing of the signal.

Alternatively we can consider the same problem in the spatial domain, which we refer to as shift variance. Shift variance describes a large change in wavelet co-coefficients with a small change in signal position. It is caused by the critical down-sampling of the original signal. This is demonstrated for the Haar wavelet in figure 5.11. In this example a delta function is convolved with the scaling function (low pas filter) from a Haar wavelet. When the output signal is down-sampled there will be twice as much energy in the odd co-coefficients compared to the even co-coefficients.

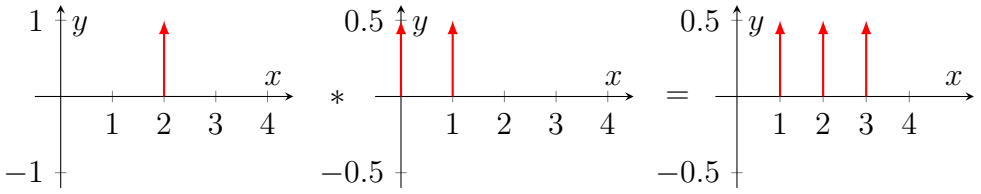


Figure 5.11: Shift variance is caused by the critical down-sampling of the original signal. In this example a delta function is convolved with the scaling function from a Haar wavelet. When the output signal is down-sampled there will be twice as much energy in the odd coefficients compared to the even coefficients.

A simple approach to generating shift invariant discrete wavelet transforms is summarised in [118]. Instead of down-sampling the signal at each level of the DWT the filters are up-sampled by inserting zeros between each wavelet coefficient. The sampling frequency is now n_0 and single pixel shifts of the input signal will not cause shift variance. This representation is highly redundant however, therefore another approach is required to efficiently represent a signal.

5.3.5 Oscillation About Singularities

Another problem with discrete wavelet transforms is oscillation of the wavelet coefficients about edges and singularities. Consider a Daubechies wavelet convolved with a delta function 5.12. The wavelet produces both small and large coefficients close to the

singularity. As we will see in the next section the coefficients of the Complex Wavelet Transform (ℂWT) are proportional to their distance from the singularity.

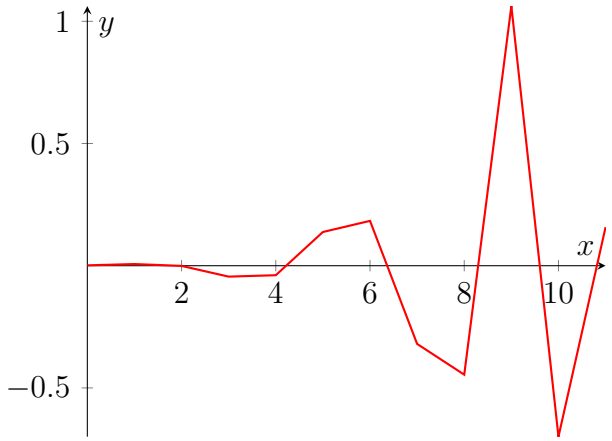


Figure 5.12: Daubechies wavelet.

5.3.6 Complex Wavelet Transform

In this section we introduce the ℂWT. We begin by noting that the Discrete Fourier Transform is not affected by shift variance or aliasing. In section 5.2.1 it was shown that Fourier transform is a special case of the wavelet transform for $\psi_{jk} = cos(2\pi(x - k)/j) + isin(cos(2\pi(x - k)/j))$. The Fourier transform has two interesting properties that are missing from the more general wavelet transform. Firstly the Fourier transform has infinite support, therefore perfect stop bands can be achieved and aliasing is not introduced by the filtering. Secondly there is an imaginary component that is $90°$ out of phase with the real component. Therefore the magnitude of the Fourier transform does not oscillate about singularities.

The problem of oscillation about singularities is solved by using a ℂWT. Approximate shift invariance and greater direction selectivity is introduced in the next section using the DTℂWT. A complex wavelet $\psi_c(x)$ can be defined as:

$$\psi_c(x) = \psi_r(x) + i\psi_i(x) \tag{5.23}$$

Where $\psi_r(x)$ is even and real and $i\psi_i(x)$ is imaginary and odd. Additionally if $\psi_r(x)$ and $\psi_i(x)$ are $90°$ out of phase with each other then $\psi_c(x)$ is an analytic signal. As with the standard wavelet transform a dyadic scaling $2^{j/2}\psi_c(x)(2^j x - k)$ is applied to produce wavelet coefficients

$$d_c(j,k) = d_r(j,k) + id_i(j,k) \tag{5.24}$$

with magnitude

$$|d_c(j,k)| = \sqrt{|d_r(j,k)|^2 + |id_i(j,k)|^2} \tag{5.25}$$

and phase

$$\angle d_c(j,k) = arctan\left(\frac{d_i(j,k)}{d_r(j,k)}\right) \tag{5.26}$$

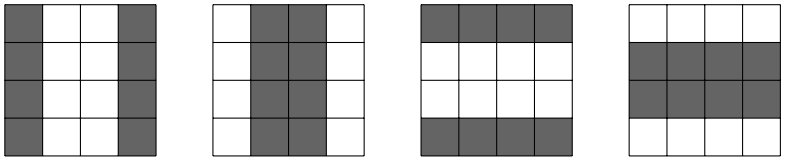
In the next section we introduce a computational approach to calculating the $\mathbb{C}$WT.

5.3.7 Dual Tree Complex Wavelet Transform

The implementation of the DT$\mathbb{C}$WT is surprisingly simple. From equation 5.23 it can be seen that the $\mathbb{C}$WT can be decomposed into two real DWTs with one tree calculating the real coefficients and the second tree calculating the complex coefficients. There is no complex arithmetic involved in the implementation, as long as the two wavelet filters form an approximate Hilbert transform pair (they are 90°out of phase with each other) the filter will be analytic. While the topic of filter design is outside the scope of this thesis it is interesting to consider the design of the low pass filters. The low pass filters for each tree are shifted versions of each other with a 1 pixel shift. In the 1D case while each tree is critically sampled the combination of the results from the two trees is two times expansive. The additional samples combined with the one pixel shift between the two trees result in a wavelet magnitude which is approximately shift invariant.

5.3.8 Oriented Wavelets

While shift invariance and stability about singularities is very useful for texture analysis, the main attraction of the DT$\mathbb{C}$WT is its directional selectivity. The 2D DT$\mathbb{C}$WT is implemented as a row and column filter as for the standard 2D DWT. The interesting result for image analysis is the way in which the two trees are combined to produce directionally selective wavelets. An intuitive understanding can be obtained using the simple graphical notation presented in an IEEE Signal Processing magazine article by Selesnick et. al. [119]. The support of a wavelet is represented in Fourier domain by figure 5.13. The high pass filter along the rows of the image has support in the upper half and lower half of the positive and negative spectrum. The low pass filter along the rows of the image has support in the lower half of the frequency domain.



(a) High Pass Row Filter (b) Low Pass Row Filter (c) High Pass Column Filter (d) Low Pass Column Filter

Figure 5.13: DWT filters in the Fourier Domain.

The standard 2D DWT filters can be represented in the Fourier domain by a multiplication of the column and row filters. For the standard 2D DWT there are three detail coefficients. Because the 2D DWT is not analytic the wavelets have support in the positive and negative frequency domain. This results in the chequer-board artefact that can be observed in the HH wavelet.

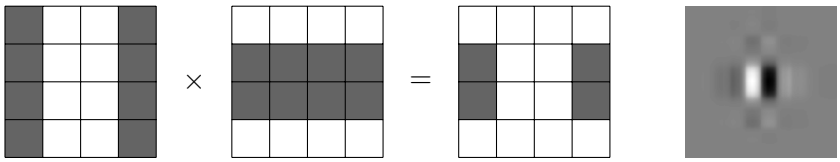


Figure 5.14: $\psi_1(x, y) = g(x)h(y)$ LH wavelet

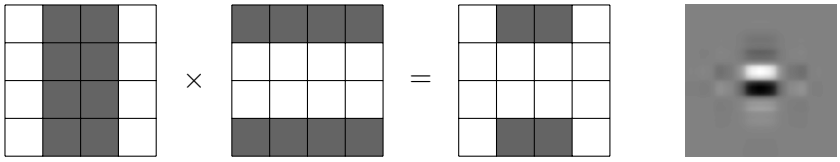


Figure 5.15: $\psi_2(x, y) = h(x)g(y)$ HL wavelet

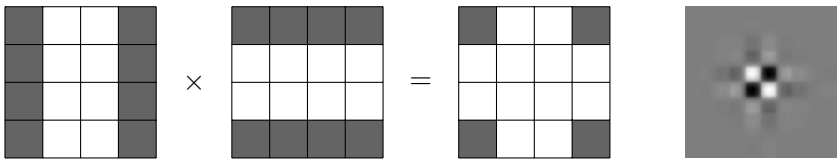


Figure 5.16: $\psi_3(x, y) = h(x)h(y)$ HH wavelet

The approximately analytic filters of the DT$\mathbb{C}$WT only have support in one half of the frequency domain. We can exploit this to build a set of oriented wavelets. Consider the wavelet produced by the row and column complex high pass filter $HH_c(x, y) = h_c(x)h_c(y)$. $HH_c(x, y)$ is complex and $h_c(x) = h_r(x) + ih_i(x)$. This can again be represented diagrammatically as

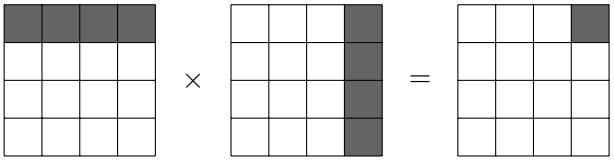
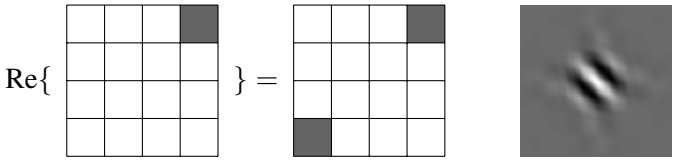


Figure 5.17: $HH_c(x, y) = h_c(x)h_c(y)$

If we take the real part of $HH_c(x, y)$ then the function must be symmetric with respect to the origin. Here we can see that we have obtained the 45°oriented real wavelet.

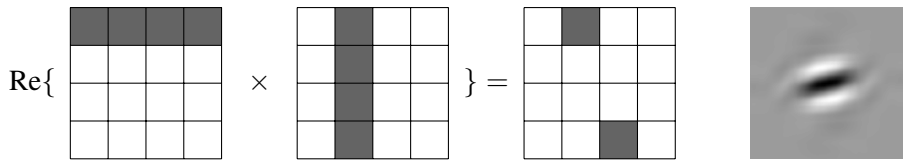


It is not immediately obvious how to obtain the other directional filters. The first step is to realise that $\text{Re}\{\psi_c(x, y)\}$ can be written as the sum of two separable 2D DWT filters.

$$\begin{aligned} HH_c(x, y) &= [h_r(x) + ih_i(x)] \times [h_r(y) + ih_i(y)] \\ &= h_r(x)h_r(y) - h_i(x)h_i(y) \\ &+ i[h_r(x)h_i(y) + h_i(x)h_r(y)] \end{aligned} \tag{5.27}$$

$$\begin{aligned} \text{Re}\{HH(x, y)\} &= h_r(x)h_r(y) - h_i(x)h_i(y) \\ &= HH_r(x, y) - HH_i(x, y) \end{aligned} \tag{5.28}$$

Here, the 45°filter is simply the difference of the HH filters from the separable real and imaginary 2D DWT trees. A similar exercise can be complete for the other combinations of high and low pass filters. The final step is given by observing that the complex conjugate of a complex wavelet flips the support about the origin in the frequency domain. An example of the high pass filter multiplied by the complex conjugate of the low pass filter is shown below. The other filters are left as an exercise for the reader.



$$
\begin{aligned}
H\bar{L}_c(x,y) &= [h_r(x) + ih_i(x)] \times [g_r(y) - ig_i(y)] \\
&= h_r(x)g_r(y) + h_i(x)g_i(y) \\
&+ i[h_i(x)g_r(y) - h_r(x)g_i(y)]
\end{aligned} \tag{5.29}
$$

$$
\begin{aligned}
\text{Re}\{H\bar{L}_c(x,y)\} &= h_r(x)g_r(y) + h_i(x)g_i(y) \\
&= HL_r(x,y) + HL_i(x,y)
\end{aligned} \tag{5.30}
$$

By exploiting the analytic properties of complex wavelets we have produced a set of 6 oriented real wavelets. The calculation of the real oriented wavelets is two times expansive. By an appropriate choice of filters we can generate another set of real wavelets that form a Hilbert transform pair with the first set of oriented wavelets. In this way we produce the full complex DT$\mathbb{C}$WT. These wavelets are four times expansive and are oriented, approximately analytic and shift invariant.

5.4 Method

In this section we introduce the DT$\mathbb{C}$WT and derive three textural characteristics. The textural characteristics that we introduce are the complexity, scale and anisotropy of the sea-floor. The complexity describes a scale and orientation independent measure of the contrast and density of features on the sea-floor. The scale describes the characteristic size of the features on the sea-floor. The anisotropy describes how ordered the texture of the sea-floor is. The complexity and scale are both useful for identifying difficult regions of the sea-floor (clutter, complex and ripple), while the anisotropy can be used to separate rippled regions from other clutter.

In previous work we introduced measures for the complexity and anisotropy of a texture [120] and showed that the number of false alarms from an ATR algorithm was much higher in complex and anisotropic regions. In this work we compared both the Gabor wavelet and the Haar wavelet. While the Haar wavelet was significantly faster to compute the additional directional selectivity of the Gabor wavelet resulted in greater stability with respect to the orientation of the textures. This was also noted by Fakris et al. [121] who extended the Haar wavelet approach using integer rotated Haar wavelets[85]. The wavelet features used in [120] are formulated in this chapter using the Dual Tree Wavelet transform. The Dual Tree Wavelet has good directional selectivity and is computationally efficient. Therefore, it provides an excellent compromise between the directional selectivity of the Gabor and Haar wavelets. For this reason it supersedes the work in [120], hence the work is not repeated here.

In this chapter we use the DT$\mathbb{C}$WT. The DT$\mathbb{C}$WT has 6 strongly oriented directional wavelets and is computationally efficient as it can be calculated using the standard DWT. For interested readers a more complete introduction to the properties of the DT$\mathbb{C}$WT can be found in Appendix 5.3. After we have introduced the DT$\mathbb{C}$WT we next introduce the measures of complexity, scale and anisotropy.

5.4.1 Textural Characteristics

Textural characteristics are calculated using the DT$\mathbb{C}$WT [119]. A DT$\mathbb{C}$WT decomposition of an image $I(x,y) \in \mathbb{R}, x \in \{0,...,w-1\}, y \in \{0,...,h-1\}$ results in a number of sub-band images $\psi_{s,\theta} \in \mathbb{C}$ indexed by scale $s = 1,...,m$ and orientation $\theta = 15°, 45°, 75°, 105°, 165°$. The oriented wavelets are shown in figure 5.18. The angular sub-bands are indexed by defining an angular resolution $\Delta\theta$ and an index k such that the angle $\theta = k \times \Delta\theta$. Each wavelet sub-band has size $w/2^s \times h/2^s$. The sub-bands are up-sampled using bilinear interpolation to $w/2 \times h/2$. The notation $\psi_{s,k}(x,y)$ denotes a position in the up-sampled sub-band. The wavelet energy $E_{s,k}$ is defined as

$$E_{s,k} = |\psi_{s,k}(x,y)|^2 \tag{5.31}$$

In practice the energy of features at the lower scales dominates over the energy of features at the higher scales. The power spectrum of the wavelet energy can be attributed to the statistical characteristics of the sea-floor [44]. A normalised energy $N_{s,\theta}$ is defined by weighting wavelet responses by α and β. The values of $\alpha = 0.25$ and $\beta = 0.75$ were determined experimentally.

$$N_{s,k} = \alpha^s E_{s,k}^{\beta} \tag{5.32}$$

The notation N_s represents the mean of the wavelet energy over the 6 directional sub-bands.

$$N_s = \underset{k}{\text{mean}}(E_S) \tag{5.33}$$

We now define three texture measures, Complexity, Scale and Anisotropy.

Complexity

The complexity C is defined by the mean of the normalised wavelet energy over all scales. It is a measure of the contrast in the image. Therefore, regions which have a high object density will have a high complexity. Similarly large ripples will appear more complex than shallow ripples.

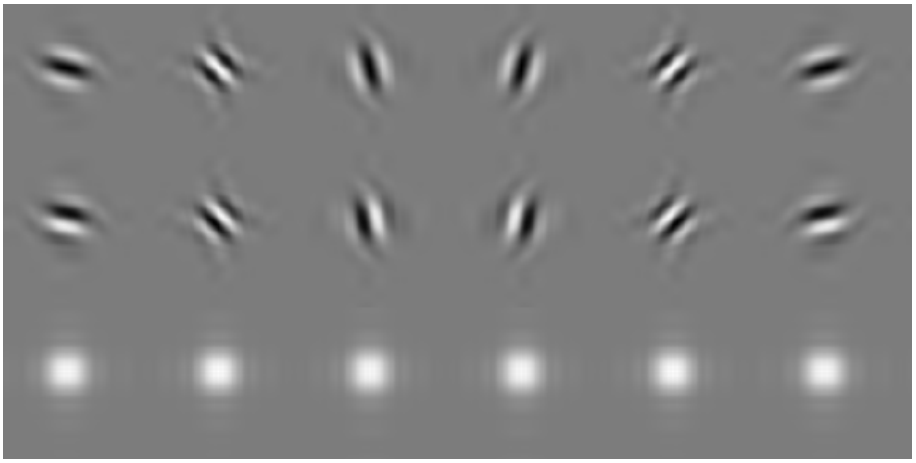


Figure 5.18: The oriented real (top) and complex (middle)wavelet filters and the wavelet energy profile (bottom) [1]

$$C = \underset{S}{\text{mean}}(N_s) \tag{5.34}$$

Scale

The DT$\mathbb{C}$WT measures the energy in the signal over a set of frequency sub-bands or scales. We define the Scale S of a texture as the sub-band that exhibits the maximum normalised energy.

$$S = \underset{s}{\text{argmax}}(N_s) \tag{5.35}$$

Anisotropy

The anisotropy of a texture is a measure of its directional homogeneity. Textures such as ripples have a maximum wavelet response in the directional sub-band parallel to the ripples and a minimum response orthogonal to the ripples. Anisotropic textures therefore have a high standard deviation when compared with isotropic textures. The anisotropy A_s is normalised by the mean wavelet response (or complexity), such that it is not dependent on the image contrast.

$$A_s = \sum_{s=2}^{n} \frac{\text{stddev}_k(N_{s,k})}{N_s} \tag{5.36}$$

The smallest scale $s = 1$ is not included in the calculation of anisotropy as the response is dominated by speckle noise.

5.4.2 Description of Data Set

Our data set consists of several survey missions. Each mission is conducted using a different sonar type. We use two different types of SSS Marine Sonic, and Klein and the experimental MUSCLE SAS system from Centre for Maritime Research and Experimentation (CMRE). From each data set we select 5 images that display interesting textures or clutter and manually classify the sea-floor types in the image. We resize all of the images to the same resolution of 5cm per pixel in both the along track and cross-track resolution.

5.4.3 Results and Analysis

In this section we present the results for the sea-floor characterisation of a set of SSS and SAS images. We first present examples of the sea-floor characteristics for three sonar types. Next we plot the distribution of each of the characteristics with respect to the manual classification of the sea-floor. Finally, we discuss the variation of the characteristics and with respect to different models of sonar.

Figure 5.19: Jet colour map. The notation Jet(a,b) is used to represent values scaled between a and b

Figure 5.20 shows the scale, complexity and anisotropy for a MUSCLE SAS image. First we consider the scale. From the scale image we can immediately identify the flat, and non-flat regions. The flat regions typically have a scale less than 2 while the rippled regions have a scale between 2 and 3. The complex regions have a much less regular scale of between 2 and 4.

The cluttered region to the right hand side of the image is also clearly identifiable in the complexity image. However, the ripples have a similar complexity to the flat regions. This is because there is a lower contrast between the highlight and shadow in the rippled region. Additionally, the flat regions have significant structure at scales 0-1, resulting in a high complexity score.

In the MUSCLE SAS image the scale is clearly better for separating flat and non-flat regions. For example, the flat region to the left side of the image shows regular structure which is identified as complex. However, the structure is below the scale of a target object and therefore would not interfere with a MCM mission. In figures 5.21-5.22 there is no

structure to the flat regions and they consist entirely of speckle noise. Therefore, either the complexity or the scale could be used to separate flat and rippled areas.

The rippled regions can be separated from the other complex areas using the anisotropy. In the MUSCLE SAS images the long regular ripples have a high anisotropy score. However, in the Klein and Marine Sonic images the ripples are less regular and there is less difference between the anisotropy of rippled and cluttered regions.

The results for the Klein image in figure 5.21 and the Marine Sonic image in figure 5.22 are similar to the MUSCLE image. However, there are a number of significant difference. First we consider the difference due to the sonar types. The Marine Sonic image has significant distortion due to the beam pattern in the first 5m of the image. This can be seen in the characteristic images as a region with large scale, high anisotropy and complexity. Effectively, no useful information can be gained from the first 5m of the image. This is reflected later in the poor segmentation scores obtained from the Marine Sonic images. Additionally the surface return that can be seen in the Marine Sonic image is visible in the scale, anisotropy and complexity images. However, it should be noted that these visual artefacts could be removed using the pre-processing methods describe in chapter 2.

From the Marine Sonic image we can also observe the difference in the behaviour of the scale and the complexity with respect to contrast. The complexity reduces towards the right hand side of the image at far range, where there is also a reduction in contrast. This is partially due to the physical attenuation of the sonar energy and partially due to a poor normalisation of the image. However, importantly we do not observe any change in the scale of the image at far range. In the next section we look at the distribution of features with respect to the manual segmented images. Examples of the manually segmented images can be seen in section 5.5.

Anisotropy

Figures 5.23 - 5.29 show histograms of the distribution of the characteristics with respect to different sea-floor types. The histograms for each sea-floor type are normalised such that the area under the graph is 1. The histograms are then stacked to produce the plots shown.

Figure 5.23 shows histograms of the anisotropy values for all sea-floor classes and all sonar types. There is some overlap between the flat and ripple classes and between the complex and ripple classes. There are a number of reasons for this. Objects on the sea-floor show a characteristic highlight shadow pattern. The shadow from an object produces a strongly oriented, anisotropic pattern on the sea-floor. Therefore, cluttered regions will have regions with high anisotropy. Similarly if a large object is placed in a ripple field then the object's shadow will be the dominant feature (figure 5.21) and the anisotropy of the

ripples is reduced. Finally if the ripples are not regular (figure 5.22) then the anisotropy will be lower.

This is apparent if we compare the histograms for the Klein sonar in figure 5.25 and the histogram for the MUSCLE sonar in figure 5.24. In the MUSCLE data set the ripples are very regular and typically do not have clutter on the ripples. In the Klein images the ripples are more irregular and often contain clutter. While the complex and flat regions are similar, the rippled regions show a greater degree of separation from the other classes in the MUSCLE images. Given the limitations of our data set it is not possible to make any conclusions about the stability of the anisotropy between different sonar types. In future work we intend to image the same region with all three sonar types.

Complexity

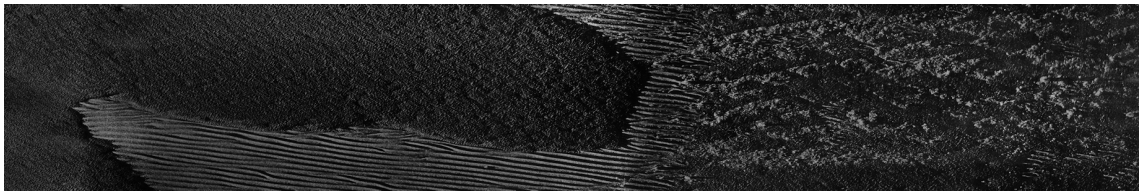
Figure 5.26 shows the histograms of the complexity values for all sonar types and all classes. As expected both the ripple and flat regions show a higher complexity than the flat regions, however there is significant overlap between the flat and complex regions. If we take the complexity values for a single sonar type, for example the Klein sonar in figure 5.27, there is much better separation between classes. As we have already discussed the complexity is dependent on the image contrast. Therefore differences in the sonar systems and the pre-processing applied result in a different range of complexity values. Therefore, while it would be possible to segment images from a single sonar type using a threshold on the complexity value, it is not possible to find a single complexity threshold for all sonar types.

Figure 5.28 shows the distribution of complexity values for complex regions separated by sonar type. From this graph we can see that as well as a significant difference between different sonar types there is also considerable variation in the complexity values for a single sonar type. Fortunately, as we will see in the next section, the texture scale gives a stable characteristic to segment complex regions.

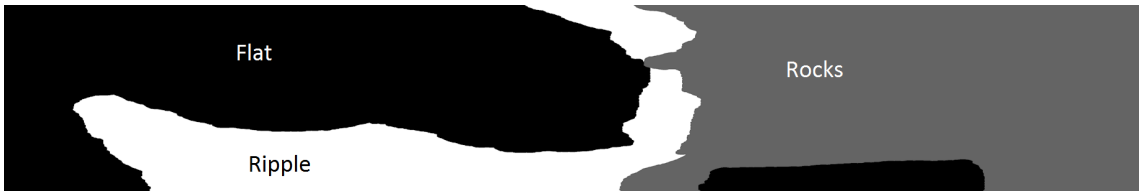
Scale

Figure 5.29 shows the histograms for the distribution of scale values with respect to sea-floor class for all sonar types. It is clear from this graph that the visual threshold for flat and non-flat classes is between scale 2 and 3. Although we see significant overlap between the histograms for each class at scale 2 this can be explained by the fact that some flat regions contain objects, just as some complex regions contain small flat regions. As we show in section 5.5 by smoothing the scale image we can obtain almost perfect separation of flat and non-flat regions. Finally, in figure 5.30 we consider the distribution of scale values for flat regions with the histograms separated by sonar type. The distribution of scale values for the Marine Sonic and the Klein sonar are similar with majority of flat

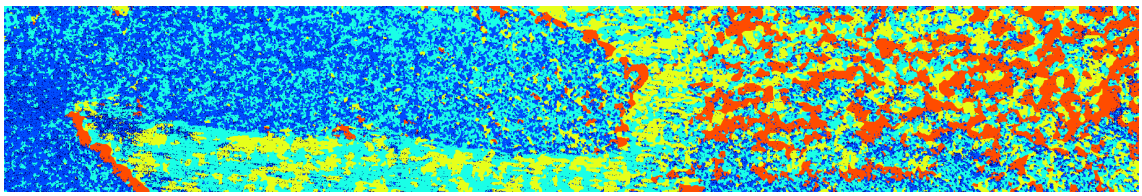
sea-floor having a scale of 0-1. The majority of the flat sea-floor in the MUSCLE sonar has a scale of between 1-2. Visually the flat sea-floor in the MUSCLE images is not as smooth as the the other two sonar types, therefore it is reasonable to assume that the scale is independent of the sonar type.



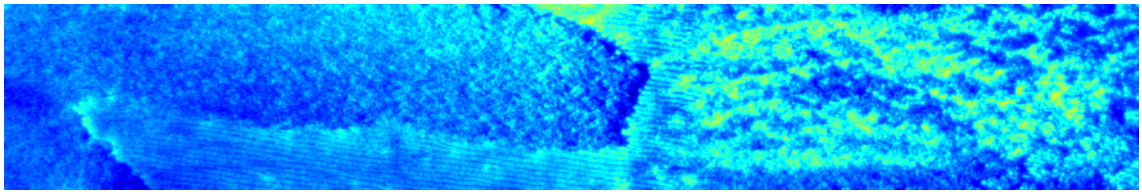
(a) Original MUSCLE SAS image



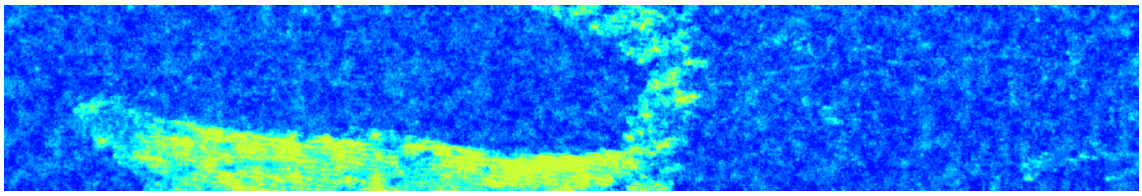
(b) Ground truth image



(c) Scale

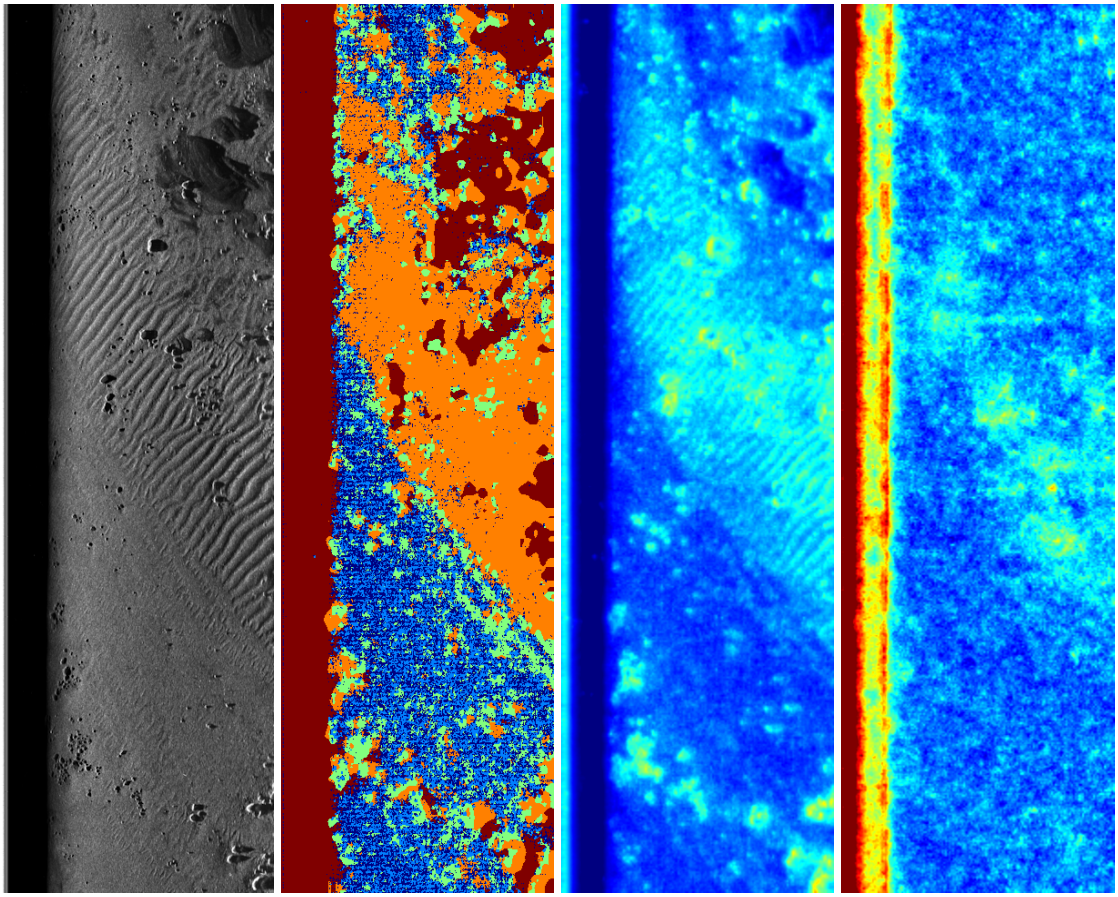


(d) Complexity



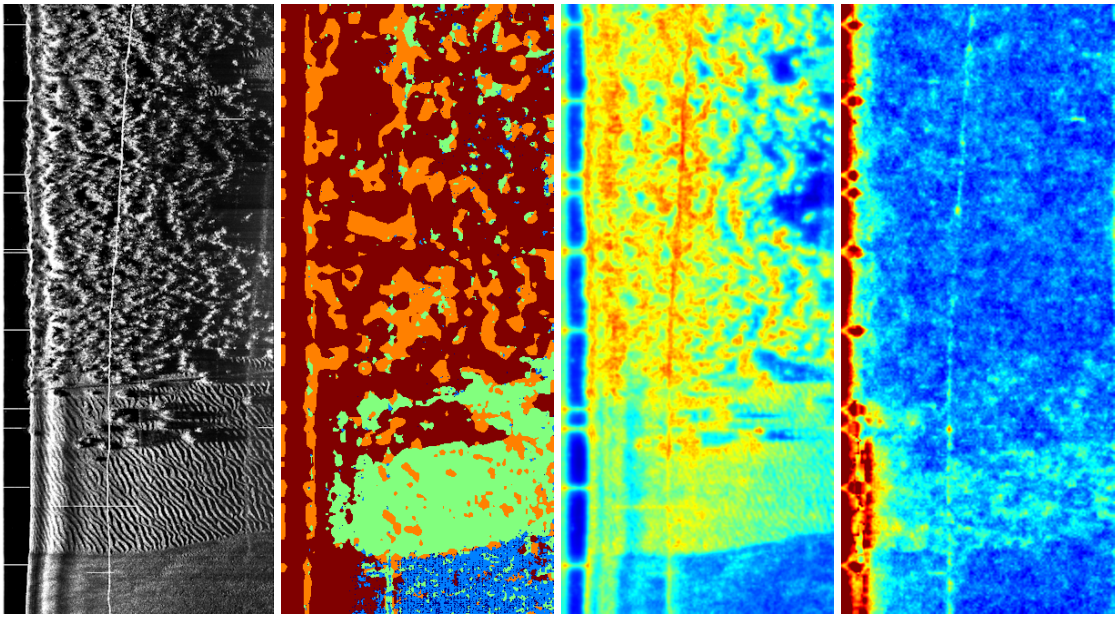
(e) Anisotropy

Figure 5.20: Example of the Scale, Complexity and Anisotropy values for a MUSCLE SAS image. The Scale is shown using the colour map Jet(0,4), Complexity and Anisotropy are shown using the colour map Jet(0.1,0.8). (Key shown in figure 5.19)



(a) Original Klein image (b) Scale (c) Complexity (d) Anisotropy

Figure 5.21: Example of the Scale, Complexity and Anisotropy values for a Klein SSS image. The Scale is shown using the colour map Jet(0,4), Complexity and Anisotropy are shown using the colour map Jet(0.1,0.8). (Key shown in figure 5.19)



(a) Original Marine Sonic image (b) Scale (c) Complexity (d) Anisotropy

Figure 5.22: Example of the Scale, Complexity and Anisotropy values for a Marine Sonic SSS image. The Scale is shown using the colour map Jet(0,4), Complexity and Anisotropy are shown using the colour map Jet(0.1,0.8). (Key shown in figure 5.19)

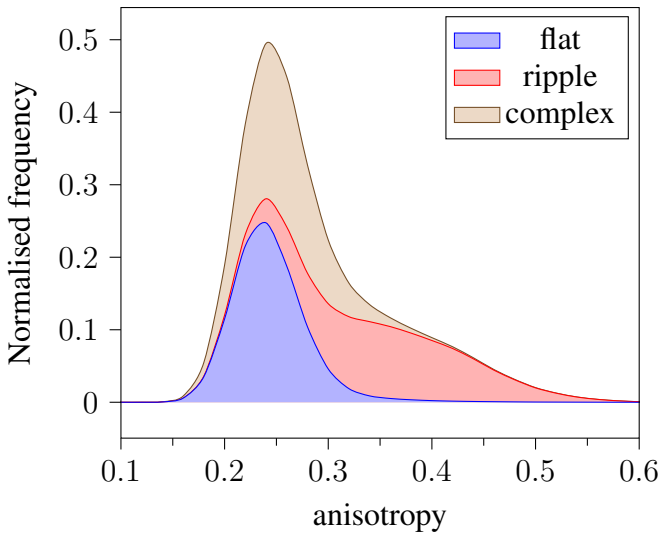


Figure 5.23: Distribution of anisotropy for all sonar types

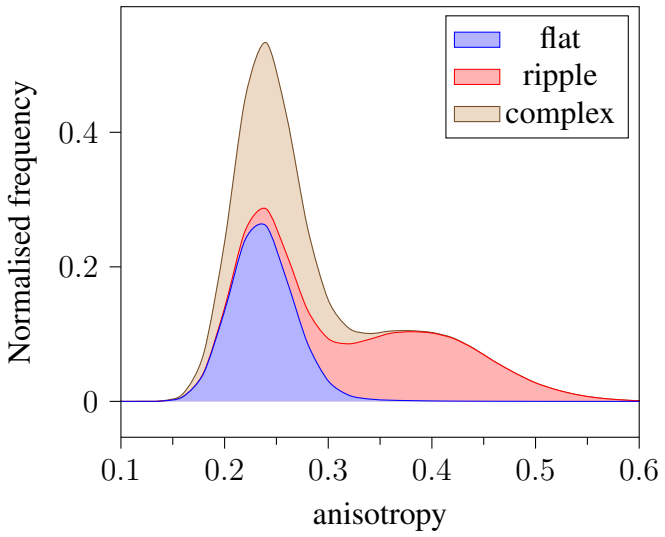


Figure 5.24: Distribution of anisotropy for the MUSCLE sonar

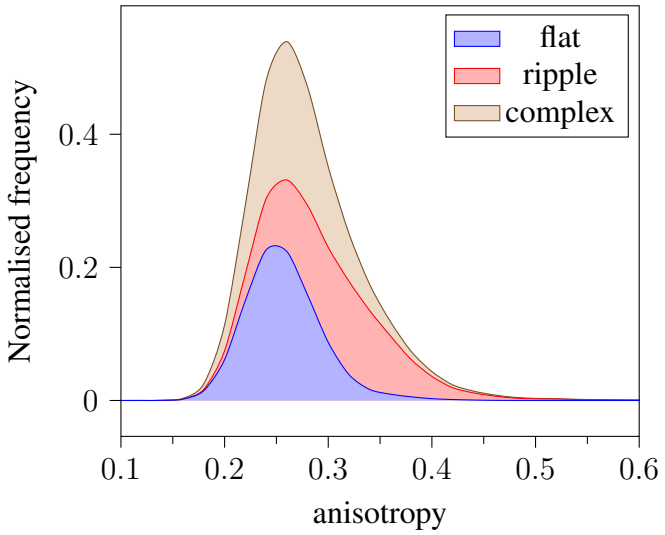


Figure 5.25: Distribution of anisotropy for the Klein sonar

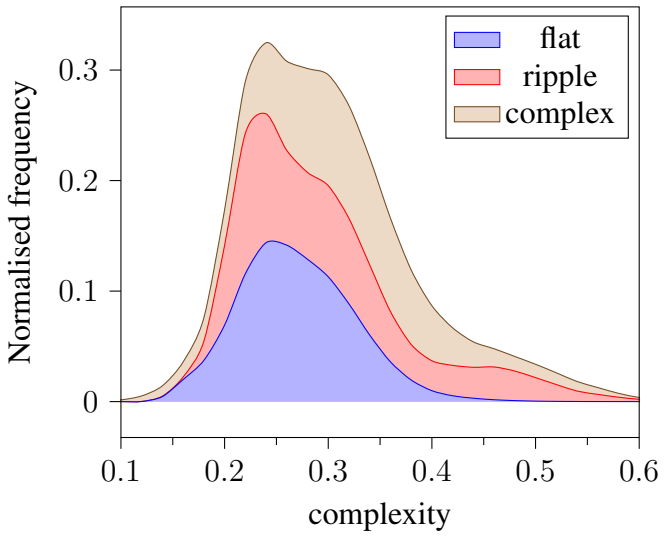


Figure 5.26: Distribution of complexity for all sonar types

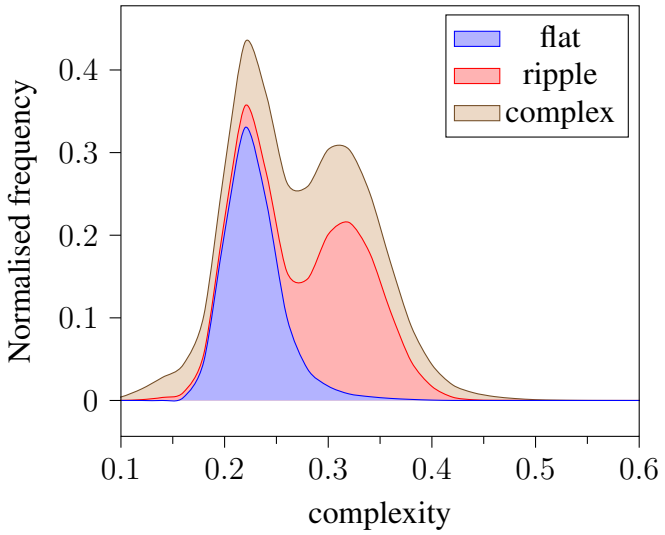


Figure 5.27: Distribution of complexity for the Klein sonar

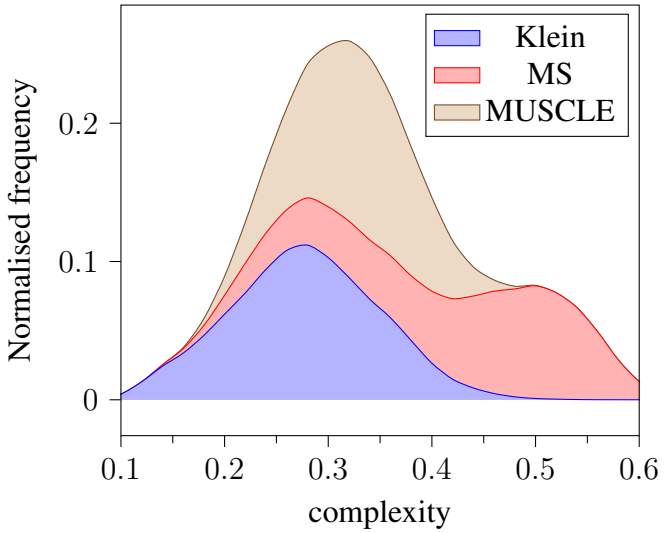


Figure 5.28: Comparison of the complexity of Complex regions for different sonar types

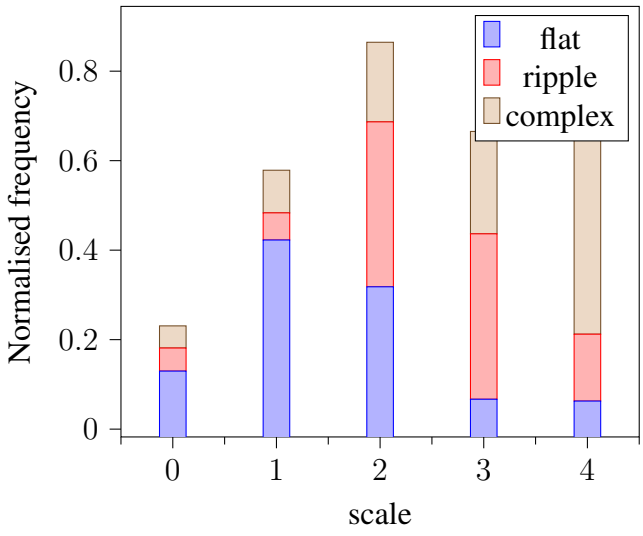


Figure 5.29: Distribution of scale for all sonar types

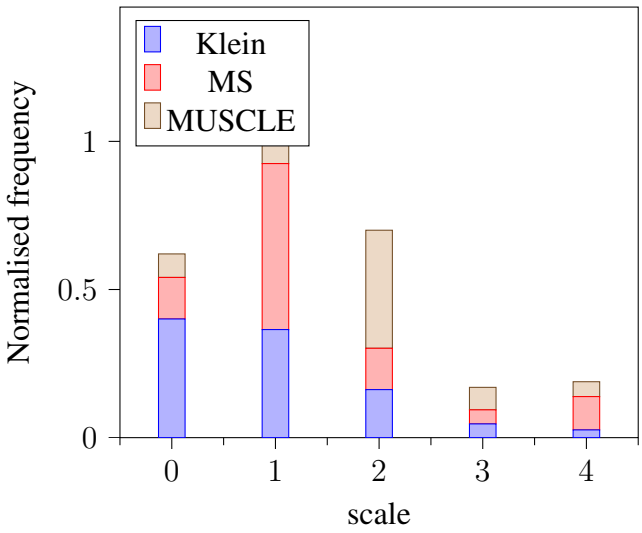


Figure 5.30: Comparison of the scale of flat regions for different sonar types

5.4.4 Discussion

In this section we have developed a new set of sea-floor characteristics for identifying flat, rippled and cluttered regions in SSS and SAS images. We first identified the shortcomings in the features that are currently used to classify sea-floor images. The key conclusion from the literature survey is that, for the most part, the interpretation of the features is performed by a machine learning stage. Therefore, while the features are descriptive, there is no direct correlation with a particular sea-floor type. The two exceptions are the fractal dimension approach to identifying sand ripples by Nelson et al. [44] and the wavelet based approach to identifying sand ripples by Williams et al. [109]. We therefore, followed a similar approach of identifying the characteristics of interest in the sea-floor and finding combinations of features that measure these characteristics.

We introduced three texture measures based on a wavelet decomposition of the image; Complexity, Anisotropy and Scale. The aim was to identify characteristics that were independent of rotation and to minimise the impact of preprocessing techniques and differences in the signal processing between sonar types. As we were not able to image the same region with different sonar types we restrict our conclusions to general comments based on observations over the data set.

First we consider the Anisotropy. The histograms for rippled and non-rippled regions of the sea-floor were well separated. While the complex and flat regions had similar histograms for all regions, the anisotropy of the rippled regions showed considerable variation. We attribute this to differences in the regularity of the ripples and the fact that some of the ripples had multiple objects in the ripple area. However, identifying cluttered and irregular ripples as complex regions is not a problem for MCM operations. Orienting the survey orthogonal to the ripple direction in these regions would have little effect on the appearance of the clutter.

While the Complexity is not independent of contrast, previous work [120, 121] has shown that it is a useful characteristic for identifying cluttered regions of the image. In this section we showed that for a single sonar type, and mission, the complexity was useful for identifying non-flat regions of the sea-floor. However, due to the contrast dependence, a different threshold is required to identify cluttered regions for each sonar type and preprocessing method. Additionally, in some of the Marine Sonic image we observed a drop in the Complexity at far range where there is a loss of contrast. Finally as we calculate the complexity over all scales (apart from scale 0) very small scale patterns can contribute considerably to the complexity.

The problems with the complexity are solved by using the Scale to identify complex regions of the sea-floor. The Scale characteristic exploits the properties of the speckle noise in sonar images. In the absence of any objects or texture the maximum normalised wavelet energy is at scale 0. Objects or regular textures are easily identified by a change

in the Scale characteristic. As the scale is only dependent on the relative energy at each scale it is not effected by changes in contrast. Therefore, it is very stable with respect to pre-processing, loss of contrast and sonar type. Additionally, unlike the Complexity and Anisotropy, the Scale can be used to accurately identify the edges of the complex regions.

We conclude that we can identify rippled, complex and flat regions in sonar images with just two characteristics; these are the Anisotropy and Scale. In the next section we demonstrate how these two characteristics can be used to segment the image.

5.5 Segmentation of Sidescan Sonar Images

5.5.1 Introduction

In this section we consider the segmentation and classification of homogeneous regions of the sea-floor. Segmenting the sea-floor allows useful information, such as the bounding contours of a region, to be calculated. This both reduces the amount of information that needs to be stored and simplifies further processing of the data. For example, an AUV can use the bounding contour and average ripple direction to plot the best survey pattern to cover a region of ripples.

The sea-floor is a complex environment and regions are rarely completely homogeneous. For example flat sea-floor may contain small objects, and cluttered areas contain flat regions. Below a certain size it is not useful to segment these regions. However, ignoring these regions while finding accurate bounding contours can be difficult. Therefore, segmentation approaches typically require constructing and solving an optimisation problem. In order to use the segmentation approach on-board AUVs we need to find an approach that is both computationally efficient and produces accurate contours.

In this section we first review the literature on segmentation in SSS and SAS images. We then introduce and compare two segmentation approaches. The first approach is a mean filter followed by a threshold. This is fast but does not produce a good segmentation. The second approach utilises graph-cut, which we introduced in chapter 2. This approach is slower but produces a better quality segmentation. This is demonstrated by comparing the segmentation quality of the two methods against an expert segmentation of the images.

5.5.2 Segmentation in Sidescan Sonar image

Sea-floor classification is typically a two stage process. For each pixel, or tile, in an image a descriptor is calculated in the chosen feature space. This is referred to as the feature vector. The classification stage assigns a label or a class probability to each feature vector. Finally the segmentation algorithms groups pixels which belong to the same class. We start by discussing supervised classification algorithms before introducing the

segmentation algorithms

Supervised Classification

Supervised classification is arguably the simplest and most robust approach to classifying and segmenting sea-floor textures [23, 24, 104, 105, 107, 115, 122, 123]. Supervised classifiers infer a classification function from a labelled training set. The classification function can then be used to estimate the class likelihoods for a textural region. Assigning a label based only on maximum-likelihood results in an irregular segmentation. Techniques such as MRFs are often used to smooth the segmentation. In this section we focus on Fuzzy Logic [104] and Dempster-Shafer belief theory [24]. Both techniques fit very well with the MRF framework which we describe in section 5.5.

Fuzzy Logic is a simple method for making logical decisions based on imperfect decision boundaries. It is similar to a rule-based approach with the hard thresholds replaced by continuous functions. The functions represent the belief that a particular feature value belongs to a given class label. It is best suited to low dimensional feature vectors with simple features that are well separated between different classes.

Taking the example of Mignotte et al. [104], they calculate the length of continuous shadows in a region. Figure 5.31 shows how a set of fuzzy logic belief functions could be constructed for this feature. Regions with no, or small, shadows are likely flat. Larger shadows belong to small objects and clutter and long continuous shadows are most likely ripples. Fuzzy Logic is particularly powerful when belief functions for several variables are combined with MRFs. In this framework the labels assigned to neighbouring textures can influence the final classification.

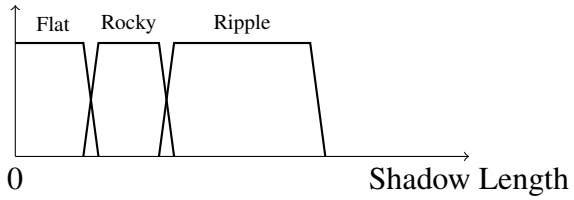


Figure 5.31: Example of Fuzzy Logic belief functions for the length of continuous shadows in a region

Unlike Fuzzy Logic Dempster-Shafer belief theory is not constrained to low dimensional feature sets. Dempster-Shafer belief theory combines multiple,single feature, Gaussian classifiers. Evidence supporting a hypothesis is assigned a normalised mass, such that the sum of all masses is equal to 1. Mass functions are combined to define concepts of belief and plausibility. The belief in a hypothesis is the sum of all masses in the set enclosed by the hypothesis. The plausibility is the sum of all the masses in the set not enclosed by the hypothesis. In other words the belief is the sum of the evidence supporting the hypoth-

esis while the plausibility is the extent to which the evidence contradicting the hypothesis leaves room for belief in the hypothesis.

For example, consider the two class sea-floor classifier (C)lutter and (F)lat shown in table 5.1. Applying the absolute decision rule which states that the belief in a hypothesis d must be greater than or equal to the the plausibility of the other hypotheses not enclosed by d ($\sim$d) we find that $Bel(C) \geq Pl(F)$ therefore the sample would be classified as clutter. The advantage of Dempster-Shafer theory over the Naive Bayes classifier is not in the classification accuracy (if joint probabilities are considered in the Naive Bayes classifier then the two approaches are mathematically equivalent) but in the ability to model ignorance. In the example provided ignorance is introduced by specifying a non-zero probability for the class hypothesis C or F. The consequence is that in some cases the model will not provide a classification. It will be shown later that this is a very useful property when the classification is smoothed using a MRF.

	Mass	Belief	Plausibility
C	0.8	0.8	0.85
F	0.15	0.15	0.2
C or F	0.05	0.05	1.0

Table 5.1: Example of belief and plausibility for a two class sea-floor classifier (C)lutter and (F)lat

Segmentation

Texture segmentation introduces constraints on the location of a texture into the classification problem, this is commonly referred to as a smoothness constraint. Intuitively a tile, or pixel, which has a similar probability of being cluttered or rippled is more likely to be rippled if all of the surrounding tiles are also rippled. To simplify the notation tiles and pixels will be referred to as nodes, while connections between neighbouring nodes will be referred to as vertices.

The balance between smoothness and classification accuracy is typically modelled by minimising a cost or energy function. The energy function can be global, as for the MRF, or local, for variational techniques such as active contours. The advantage of global techniques is that the segmentation can avoid convergence to a local minimum. However, for reasons that will be explored later, global energy constraints must be written in the form of potentials between pixels. Therefore, it is not possible to model constraints on the shape of the regions or the smoothness of borders. These additional constraints can be modelled by local techniques. However, if the model is not initialised close to the true solution it is likely that the result will be a local minima. Techniques such as simulated annealing which can avoid these problems are computationally costly.

The difference between global and local energy minimisation can be seen by comparing the energy functions from the MRF approach in [24] and the active contour approach in [68]. Before introducing these functions, a basic understanding of the Potts interaction model is required. The Potts model is a model of interacting spins on a crystal lattice. It is interesting not because of it accuracy but because it is exactly solvable. Additionally,fast computational methods exist for solving the system. The Potts Hamiltonian consists of an interaction term between neighbouring particles and a global term from an external magnetic field. The analogue in computer vision is a smoothness term $V_{p,q}$ which models the interaction between pixels and a data term $D_p(.)$ which forces a pixel towards it's most likely class. The typical energy function E is therefore

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{(p,q) \in N} V_{p,q}(L_p, L, q) \tag{5.37}$$

where $L = L_p | p \in P$ is a labelling of an area P.

Reed et al. [24] set $D_p(.) = -log[Bel(L_p)]$ and $V_{p,q} = B[1 - \delta(L_p, L_q)]$, where $Bel(L_p)$ is the Dempster-Shafer belief for label L_p and B is a weighting term for the smoothness term over the data term (Note that the treatment of unlabelled pixels has been ignored).

The active contour model used by Lianantonakis et al. [68] is a simplified version of the Mumford-Shah functional. Consider the textural feature space of an image as a vector field I. For two class segmentation $c1, c2$ bounded by a curve C the average of I inside C ($c1$) is defined as $m_{in}(C)$ and the average of I outside C ($c2$) is defined as $m_{out}(C)$. The simplified Mumford-Shah energy functional is then defined as:

$$\begin{aligned} E(c1, c2, C) =& \lambda \int_{int(C)} |I(x,y) - m_{in}(C)|^2 dxdy \\ &+ \lambda \int_{ext(C)} |I(x,y) - m_{out}(C)|^2 dxdy + \mu Length(C) \end{aligned} \tag{5.38}$$

The first two terms minimise the sum of squares between the image model and the textural feature space within the boundaries. This is equivalent to the data term in equation 5.37 with a euclidean distance classifier. The second term minimises the length of the boundary. This terms does not have an analogue in the Potts energy model as it is calculated from local information that can't be modelled by a MRF. Karoui et al. [69] apply a similar active contour model to segment sonar images, however they extend the scheme to multiple classes and use a log-likelihood membership function based on the Kullback-leibler divergence described in chapter 3.

From the two energy equations, the global Potts model and the local Mumford-Shah functional, it can be seen that there are several trade-offs between the two methods. The

MRF model does not need to be initialised close to the true solution for it to converge and is computationally more efficient. The active contour model can include priors on the shape of the boundary but can converge to local minima.

Conclusion

The choice of classification and segmentation algorithm is largely dependent on the properties of the textural features. We first consider the case where there are tens or hundreds of features that are not invariant to parameters such as grazing angle, rotation of textures or changes in the scale. A supervised classifier will obtain good classification results using these features over a single data set. However, moving to a new sea-floor area, or changing the direction of the survey, will result in poor classification results.

The opposite case is a small feature set, where all or some of the features are largely invariant over a single class of texture. In this case a simple rule-based classification scheme can be applied to robustly classify textures between data sets. In the previous section we demonstrated that the Scale and Anisotropy of different sea-floor texture were invariant with respect to the type of sonar. We show, in the next section, that consequently we can apply a simple supervised classification algorithm to classify the sea-floor. Next, the smoothness problem is addressed using the Potts model described in the previous section. Additionally, we improve the computational efficiency and accuracy by using the graph-cuts [40] approach to solve the energy optimisation problem.

5.5.3 Method

In this section we describe two segmentation approaches. The first is a simple rule-based segmentation approach based using a local average of the characteristics. We compare this to a second approach using MRFs and graph cuts.

Local Average

The simplest segmentation approach is to calculate a local average for each characteristic and apply a threshold. As for the more complex approach in the next section, we first segment flat from non-flat terrain. A Gaussian blur with a window size of 51x51 pixels and a standard deviation of 5 pixels is applied to the characteristic images. Non-flat terrain is segmented by applying a threshold at $Scale > 1.5$ to the Scale characteristic. Ripples are then segmented from non-ripple regions by applying a threshold of $Anisotropy > 0.32$ to the Anisotropy characteristic. This allows us to segment the image into the three classes, flat, complex and rippled. In section 5.5.4 we demonstrate that a reasonable segmentation can be obtained using this approach. However, this can be improved by using the graph-cuts approach presented in the next section.

Graph-Cuts

Graph cuts [40] is a computationally efficient method for solving binary segmentation problems on a MRF. We first introduced this approach in chapter 2 and it is explained in more detail in appendix 2.7.1. As in the previous section we divide the segmentation problem into two binary segmentations. First flat and non-flat regions are segmented based on scale and then ripple and non-rippled regions are segmented using the anisotropy. First we introduce the notation, then we describe the data and neighbourhood terms.

We model the MRF as an undirected flow graph $G =< V, E >$ where a set of vertices $v \in V$ are connected by edges $(u, v) \in E$. A 4-connected neighbourhood about each vertex ne(v) is defined. A capacity $c_{(u,v)}$ is associated with each edge. Each vertex is also connected to a source vertex s and a sink vertex t $s, t \in V$.

The graph is solved to push the maximum flow from the source to the sink. The flow may not exceed the capacity such that $f_{u,v} \leq c_{u,v}$ and flow must be conserved $\sum_{u:\ (u,v)\in E} f_{uv} = \sum_{u:\ (v,u)\in E} f_{vu}$ for each $v \in V \setminus \{s, t\}$. The value of the flow is the sum of the flow coming from the source.

$$|f| = \sum_{v:(s,v)\in E} f_{s,v} \tag{5.39}$$

The flow graph is solved to find the routing that is associated with the maximum flow being pushed from the source to the sink $f_{max} = \text{argmax}|f|$.

The primal-dual problem to calculating the maximum flow is calculating the minimum s-t cut on the graph. An s-t cut $C = (S, T)$ is a partition of V such that $s \in S$ and $t \in T$. The capacity of an s-t cut is defined by the sum of the capacities of all the edges that connect the source and sink sets. Only edges that go from the source set to the sink set are include in the capacity.

$$c(S, T) = \sum_{(u,v)\in S\times T} c_{uv} \tag{5.40}$$

In appendix 2.7.1 we showed that the optimal segmentation of a MRF is equivalent to minimising an energy equation of the form.

$$U(L|x, \text{ne}(L)) = \sum_{p\in V} U(x_p|L_p) + \sum_{p\in V, v\in \text{ne}(p)} U(L_p|L_v) \tag{5.41}$$

where $x \in X$ is a feature vector, $L \in \{0, 1\}$ is a binary class label, ne(L) is the 4 connected neighbourhood about L. The data term $U(x_p|L_p)$ assigns an energy cost to a feature vector x for adopting a label L. The neighbourhood term assigns an energy cost $U(L_p|L_v)$ for adopting a label L_p given a neighbouring label L_v.

We model our belief that a characteristic describes a particular sea-floor class using a

sigmoid log-likelihood data term. We first define a likelihood function $\mathcal{L}$:

$$\mathcal{L}(x; x_{min}, x_{max}, x_{inf}, \alpha) = \frac{\tanh\left(\frac{\alpha(x - x_{min} - x_{inf})}{x_{max} - x_{min}}\right) + 1}{2} \tag{5.42}$$

where x_{min} and x_{max} are the minimum and maximum values expected for the characteristic x and x_{inf} is the inflection point for the sigmoid function and α controls the steepness of the curve.

We show the likelihood function $\mathcal{L}(x_{scale}; 0, 5, 1.5, 5)$ for complex terrain in figure 5.32 . The inflection point is determined from the histograms in figure 5.30. α is chosen to approximate the spread of the scale values observed. Similarly we define a likelihood function for rippled regions based on the anisotropy $\mathcal{L}(x_{anisotropy}; 0.1, 0.5, 0.32, 5)$. This is shown in figure 5.33.

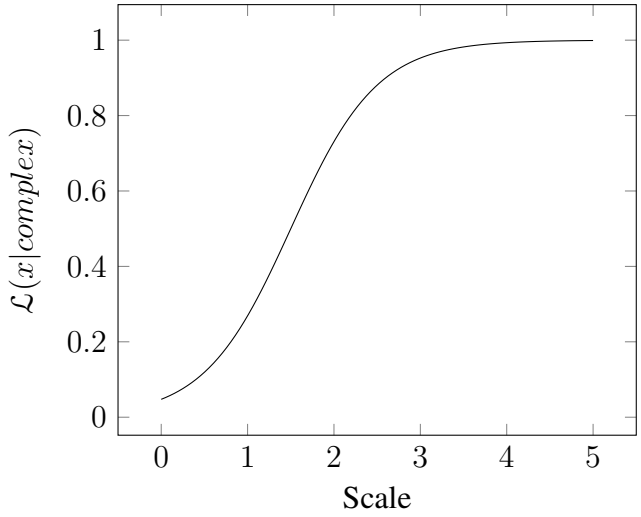


Figure 5.32: Likelihood function for complex terrain based on the scale characteristic. $\mathcal{L}(x_{scale}; 0, 5, 1.5, 5)$

Next we define the negative log-likelihood function $S(x)$:

$$S(x) = -\log(\mathcal{L}(x)) \tag{5.43}$$

and the inverse function S_{inv}:

$$S_{inv}(x; x_{min}, x_{max}, x_{inf}) = -\log(1 - \mathcal{L}(x)) \tag{5.44}$$

The data terms for flat and complex sea-floor are then defined as :

$$U(x_{scale}|L_{Flat}) = S(x_{scale}, 0, 5, 1.5, 5) \tag{5.45}$$

$$U(x_{scale}|L_{Complex}) = S_{inv}(x_{scale}, 0, 5, 1.5, 5) \tag{5.46}$$

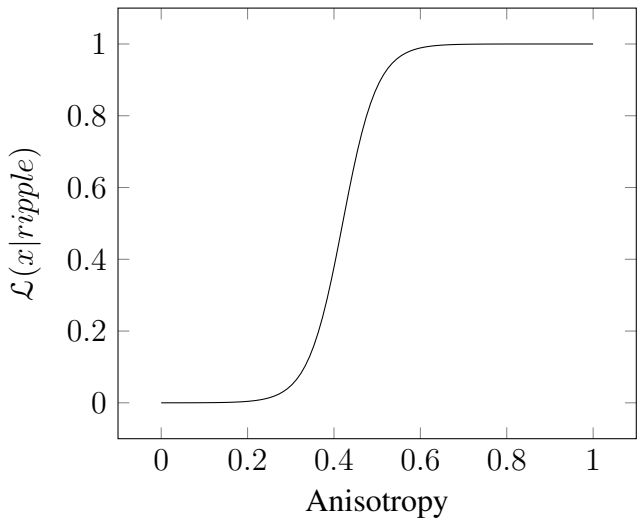


Figure 5.33: Likelihood function for rippled terrain based on the anisotropy characteristic. $\mathcal{L}(x_{anisotropy}; 0.1, 0.5, 0.32, 5)$

The data terms to differentiate between complex and rippled sea-floor are defined as:

$$U(x_{anisotropy}|L_{Complex}) = S(x_{anisotropy}, 0.1, 0.5, 0.32, 5) \tag{5.47}$$

$$U(x_{anisotropy}|L_{Ripple}) = S_{inv}(x_{anisotropy}, 0.1, 0.5, 0.32, 5) \tag{5.48}$$

Finally we discuss the neighbourhood or smoothness term. From the characteristic images in section 5.2 it can be seen that the edges in the Scale image are a good match to the boundaries between sea-floor types. Therefore, the neighbourhood term $U(L_p|L_v)$ assigns a cost to assigning the same class label to pixels either side of an edge in the scale image. $U(L_p|L_v)$ is defined as:

$$U(L_p|L_v) = \beta e^{|Scale_p - Scale_v|} \tag{5.49}$$

where β controls the weighting between the neighbourhood term and the data term.

Analysis Framework

The quality of the segmentation is evaluated by calculating the Jaccard Index with respect to the expert segmentation. For each class the Jaccard Index is defined as size of the intersection divided by the size of the union of the sample sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.50}$$

While the Jaccard index provides a quantitative measure of the segmentation quality it has a number of shortcomings. The key shortcoming is that we are assuming that our expert segmentation is in some sense correct. Furthermore the Jaccard index does not pe-

nalise unwanted features such as small gaps in an otherwise uniform segmentation or poor segmentation of the boundaries between different textural reasons. Unsupervised evaluation of image segmentation is a research field in its own right [124], therefore given the small size of our data set we present a subjective analysis of the two image segmentation approaches.

5.5.4 Results and Analysis

In this section we present a comparison of the local average and graph segmentation approaches. We then demonstrate the effect of the smoothness parameter on the graph segmentation. Figures 5.34 - 5.37 show examples of the original sonar image, an expert segmentation of the image and a comparison of the two segmentation approaches. In these example a smoothness value of $\beta = 15$ was used for the graph segmentation. The segmentation quality is compared using the Jaccard Index is table 5.2.

The first interesting observation is that the local average approach provides a good segmentation of the different sea-floor types. This is not surprising given that the characteristics were designed to detect these sea-floor types. However, these results prove that a single threshold can be applied to the scale and anisotropy characteristics over all three sonar types that we tested. The local average segmentation approach is also very efficient taking just 0.3s to process a 1833x299 pixel image on a single core of a 1.6 GHz Intel i7 Q 720 processor. Even allowing for the slower processors typically found on-board AUVs this means that we can perform the sea-floor characterisation with little impact on the computational load of the AUV.

The graph segmentation approach produces visually similar results to the local average approach. However, in figures 5.35 and 5.37 there is a clear improvement in the quality of the segmentation of the ripples. The question is why we see comparable results for the complex regions but an improvement in the ripple regions. In the local average approach the ripples are identified from just the Anisotropy image. As we saw in section 5.2 the Anisotropy image is susceptible to objects in the ripple field and regions of irregular ripples. Additionally, the local averaging tends to underestimate the anisotropy at the boundary between flat and rippled regions. As a result the vertical strip of ripples in the center of figure 5.34 is not well segmented. The graph segmentation approach only uses the scale image to calculate the smoothness term. As a result the class boundaries are likely to map to large changes in scale. We can see that in figure 5.37 the local averaging approach results in an irregular segmentation of the ripple region. However, we see in figure 5.22 that the ripple field has a constant scale, therefore the graph segmentation achieves a much more regular segmentation. This difference is only noticeable in small or irregular regions of ripples which are close to the Anisotropy threshold. The graph segmentation approach, while slower than the local average, is also very efficient taking

Sea-floor Type	Local Average	Graph (β=15)	Graph (β=30)	Graph (β=60)
Flat	0.50 ± 0.09	0.46 ± 0.09	0.54 ± 0.09	0.61 ± 0.07
Complex	0.69 ± 0.06	0.68 ± 0.06	0.70 ± 0.06	0.70 ± 0.06
Ripple	0.48 ± 0.08	0.49 ± 0.08	0.49 ± 0.08	0.50 ± 0.08

Table 5.2: Comparison of the segmentation quality for the local average and the graph segmentation. The smoothness parameters is shown for each example of the graph segmentation

Seafloor Type	Local Average	Graph (β=15)	Graph (β=30)	Graph (β=60)
Flat	0.5 ± 0.2	0.52 ± 0.2	0.5 ± 0.2	0.6 ± 0.2
Complex	0.65 ± 0.09	0.62 ± 0.9	0.69 ± 0.08	0.7 ± 0.08
Ripple	0.76 ± 0.08	0.7 ± 0.2	0.7 ± 0.2	0.81 ± 0.03

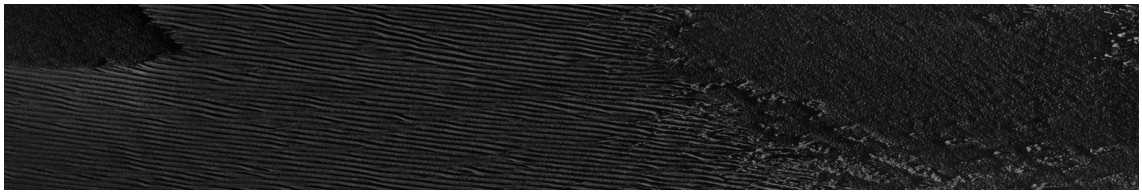
Table 5.3: Comparison of the segmentation quality for the local average and the graph segmentation using only the MUSCLE SAS images. The smoothness parameters is shown for each example of the graph segmentation

approximately 1.2s to process a 1833x299 pixel image on a single core of a 1.6 GHz Intel i7 Q 720 processor.

While in some borderline cases the graph segmentation can outperform the local averaging method, for the majority of images tested the results are comparable. In table 5.2 we can see that all of the methods tested agree to within one standard error. Therefore, to see if there is any significant difference between the segmentation approaches we consider the segmentation results for just the MUSCLE images. Table 5.3 shows the segmentation quality for just the MUSCLE SAS images. Again there is no significant difference between the segmentation approaches. However, in figure 5.38, where we compare the effect of the smoothness parameter on the segmentation, we can see that there is a clear difference in the quality of the segmentation of three images. As we increase the smoothness parameter the small segments are removed from the image. However importantly we do not loose the definition in the edges as we would if we increased the size of the filter in the local averaging method.

5.5.5 Summary

In this section we used the sea-floor characteristics, introduced in section 5.2, to segment SSS and SAS images into regions with homogeneous textural characteristics. We introduced two approaches, the first approach was simply to find a local average of the characteristics and apply a threshold to segment the different regions. The second approach used a MRF model with graph cuts to segment the regions. Both of the segmentation approaches used two binary segmentation stages. First flat sea-floor was segmented from non-flat sea-floor using the Scale characteristic then rippled regions were segmented from complex regions using the Anisotropy characteristic.



(a) Original MUSCLE SAS image

(b) Expert segmentation

(c) Local Average

(d) Graph Cuts

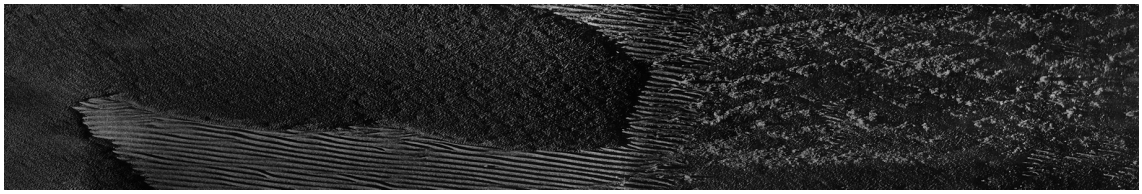
Figure 5.34: Example of image segmentation for a MUSCLE SAS image. In the Expert segmentation Black = Flat, White = Ripple, Grey = Complex. In the other images Blue = Flat, Red = Ripple, Green = Complex

While the overall segmentation accuracy agreed to within one standard error for both approaches, graph cuts achieved better segmentation of the edges of rippled regions. This can be attributed to the fact that in the graph-cuts approach we only used the Scale characteristic to optimise the position of the edges. As the ripple fields in our test images were typically at a single scale this resulted in very clear edges between regions.

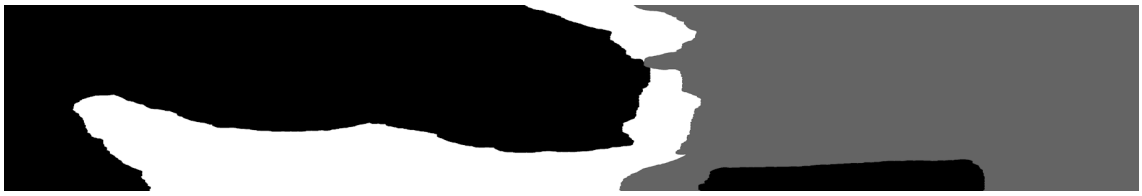
There are a number of other techniques for smoothing images such as Morphological or Bilinear filtering. These have not been considered in this chapters as the results achieved using the mean filtering address the needs for fast segmentation while the Graph Cuts segmentation provides a good quality segmentation at the expense of increased computational time.

Both of the segmentation approaches were able to operate using a single threshold on the Scale and Anisotropy for all sonar types and regions. This suggests that our hypothesis from section 5.2 that the anisotropy and scale are independent of the sonar type is correct.

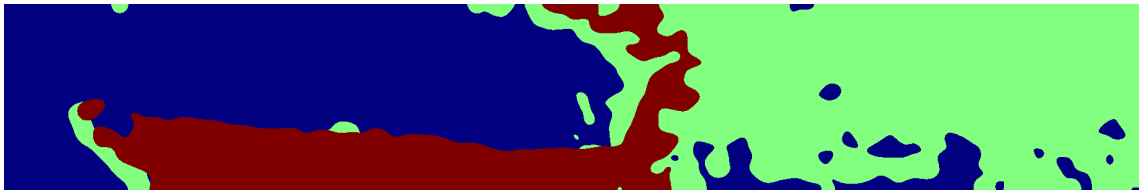
While we have shown that we can characterise and segment sonar images for MCM



(a) Original MUSCLE SAS image



(b) Expert segmentation

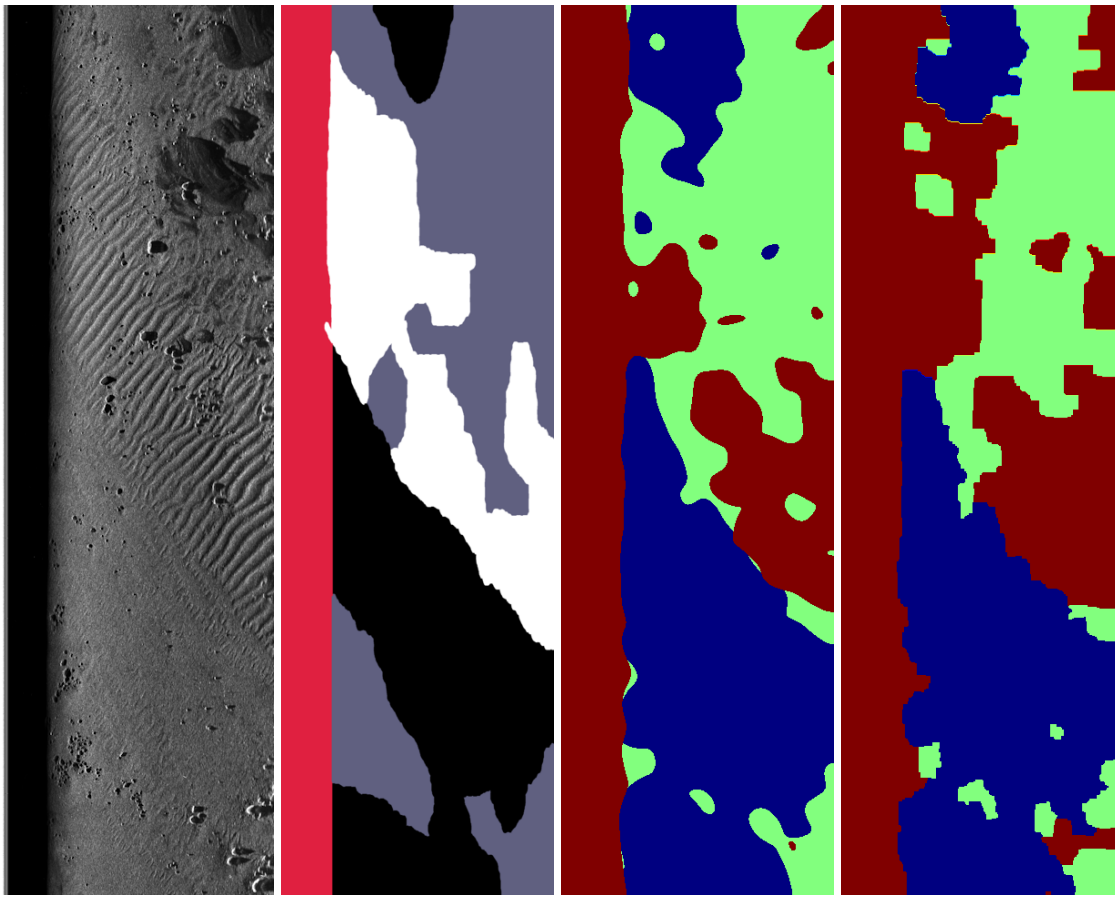


(c) Local Average

(d) Graph Cuts

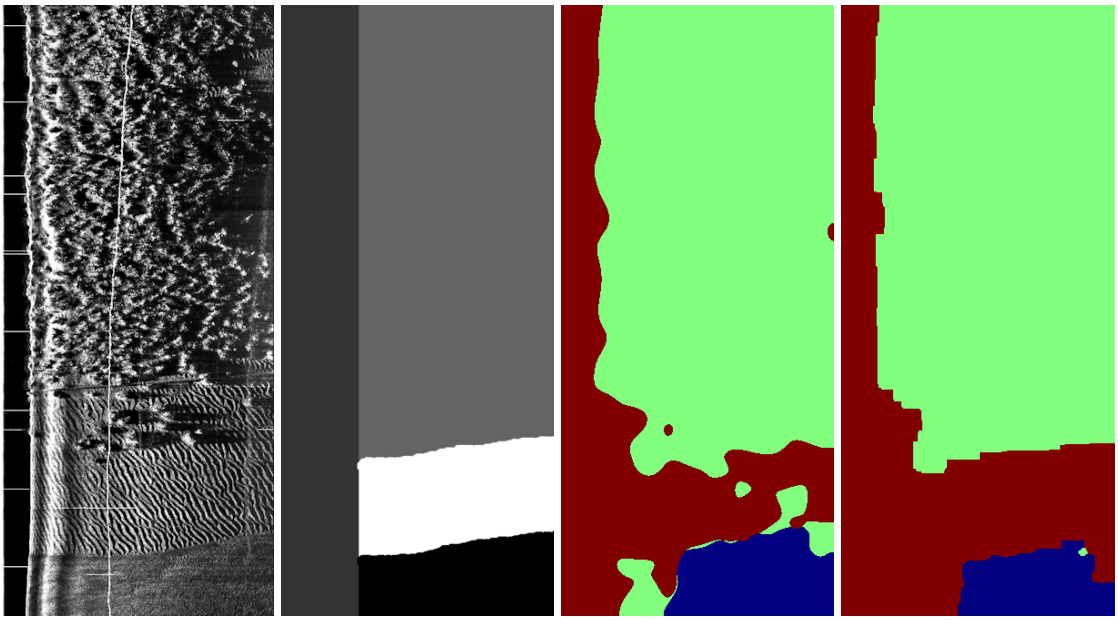
Figure 5.35: Example of image segmentation for a MUSCLE SAS image. In the Expert segmentation Black = Flat,White = Ripple,Grey = Complex. In the other images Blue = Flat, Red = Ripple, Green = Complex

operations using the textural characteristics we have not addressed the appearance of objects within these regions. For example a region identified as flat by characterising the texture could still contain tens of MLOs per image. In the next section we consider how to characterise the appearance of objects in the sonar image.



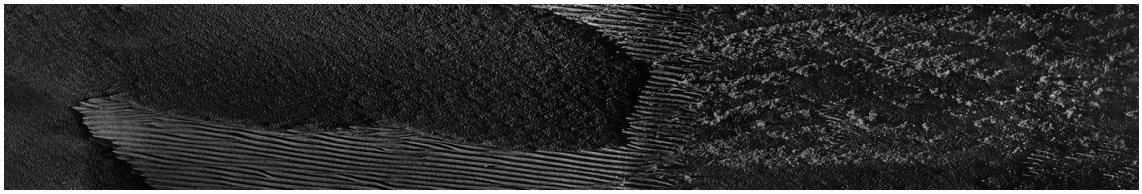
(a) Original Klein image (b) Expert Segmentation (c) Local Average (d) Graph-Cuts

Figure 5.36: Example of image segmentation for a MUSCLE SAS image. In the Expert segmentation Black = Flat,White = Ripple,Grey = Complex. The water column is shown in red and is not used in the analysis. In the other images Blue = Flat, Red = Ripple, Green = Complex



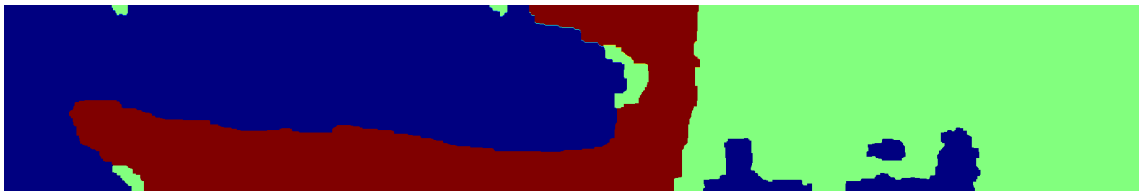
(a) Original MS image (b) Expert Segmentation (c) Local Average (d) Graph-Cuts

Figure 5.37: Example of image segmentation for a MUSCLE SAS image. In the Expert segmentation Black = Flat,White = Ripple,Grey = Complex. The water column is shown in red and is not used in the analysis. In the other images Blue = Flat, Red = Ripple, Green = Complex

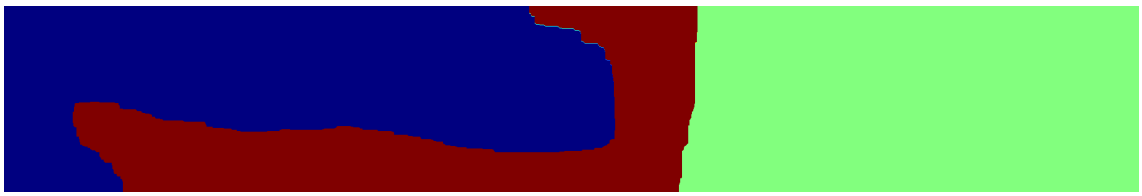


(a) Original image

(b) S = 15



(c) S = 30



(d) S = 60

Figure 5.38: Example of graph segmentation for a MUSCLE SAS image. In each image the smoothness parameter is increased. Blue = Flat, Red = Ripple, Green = Complex

5.6 Clutter Characteristics

In the previous sections we considered how to characterise and segment textures. However, inter-class variability within these regions can still affect MCM operations. In this section we consider the appearance of mine-like clutter within a region. ATR algorithms are more likely to produce false alarms for objects that are of a similar shape and size to the target object. These objects are referred to as MLOs. The textural descriptors developed in section 5.2 are not suitable for describing a region consisting of a large number of discreet objects, rather than a homogeneous texture. In this section a simple approach for estimating the density of MLOs in a region is developed.

A standard technique for extracting object information from SSS images is to segment the image into three classes $C \in$(Highlight, Background and Shadow). We use the approach described in chapter 2 to segment the image. Next we calculate a height map for all of the objects in the image and finally we calculate the density of objects with a similar height to the target object. In the next section we introduce the calculation of the height-map.

Height Map

The height of an object in a SSS image can be estimated from the length of the shadow that it casts on the sea-floor. Algorithm 2 attempts to match each highlight pixel to a set of connected shadow pixels. A set of connected shadow pixels can be assumed to be generated from a highlight pixel if they have at most n_{gap} background pixels in-between them.

Once the shadow length has been calculated, the height of each object can be calculated from the sonar altitude $a(y)$, the sonar resolution Res_x and the range of the first column of the image R_0.

$$h(x,y) = \frac{s(x,y)(s(x,y) + x + R_0)}{a(y)} \tag{5.51}$$

A number of features can be calculated from the height maps. The average height can be found by taking the mean of all non-zero pixels within a region. However, for mine hunting it is more important to know the density of objects that are of a similar height and size to a mine. Individual objects are separated from the height map by taking a Gaussian blur of the image with a standard deviation similar to the size of the target object. Local maxima within the blurred image represent individual objects. The MLO density can be found by counting the number of objects that have a height within $\pm 20\%$ of the target object height and dividing by the total search area.

Figure 5.39 shows an example of the height-map calculated for a MUSCLE SAS image. The quality of the estimated hight-map is entirely dependent on the quality of

Data: A segmented SSS image indexed by x in order of increasing slant range and y. Each pixel has class $c(x, y) \in C$ where $C \in \{$Highlight,Background and Shadow$\}$

Result: A map of shadow lengths $s(x, y) \in \mathbb{R}$ where $s(x, y)$ is the estimated shadow length in pixels of each highlight pixel $c(x, y)$

```
for y ← 0 to height do
    for x ← 0 to width do
        shift ← 1; if c(x,y) == Highlight then
            while c(x + shift,y) != Shadow do
                shift ← shift + 1;
            end
            if shift < n_gap then
                shadowlength ← 0; while c(x + shift,y) == Shadow do
                    shadowlength ← shadowlength + 1;
                end
                s(x, y) ← shadowlength;
            end
        end
    end
end
```

Algorithm 2: Algorithm for calculating the length of the associated shadow for each highlight pixel in the image

the segmentation. As we have already evaluated the image segmentation approach in chapter 2 we do not perform a quantitative analysis of the results. The application of features calculated using the height-map are discussed in more detail in the next chapter on adaptive ATR.

5.7 Conclusion

In this chapter we introduced a simple approach to characterising the sea-floor for MCM missions using an AUV. The work addressed the problem of robustly segmenting and labelling the sea-floor using low performance hardware suitable for on-board processing, and without any human input. Historically, this has not been a problem as processing of sonar data has been performed post mission [18, 96]. For this reason previous research has focussed on the accuracy of the segmentation and classification. Current approaches tend towards large features [97] sets and complex machine learning or clustering methods[98] and are not suitable for use on current embedded systems. The problem of character-siing the sea-floor for MCM operations is simpler than that of, for example, geological mapping[92]. Therefore, in this chapter we have simplified the problem to identifying flat, complex and rippled areas of the sea-floor.

We started, in section 5.2, by deriving textural characteristics to identify the different sea-floor regions. We derived three characteristics, the Anisotropy, Complexity and Scale.

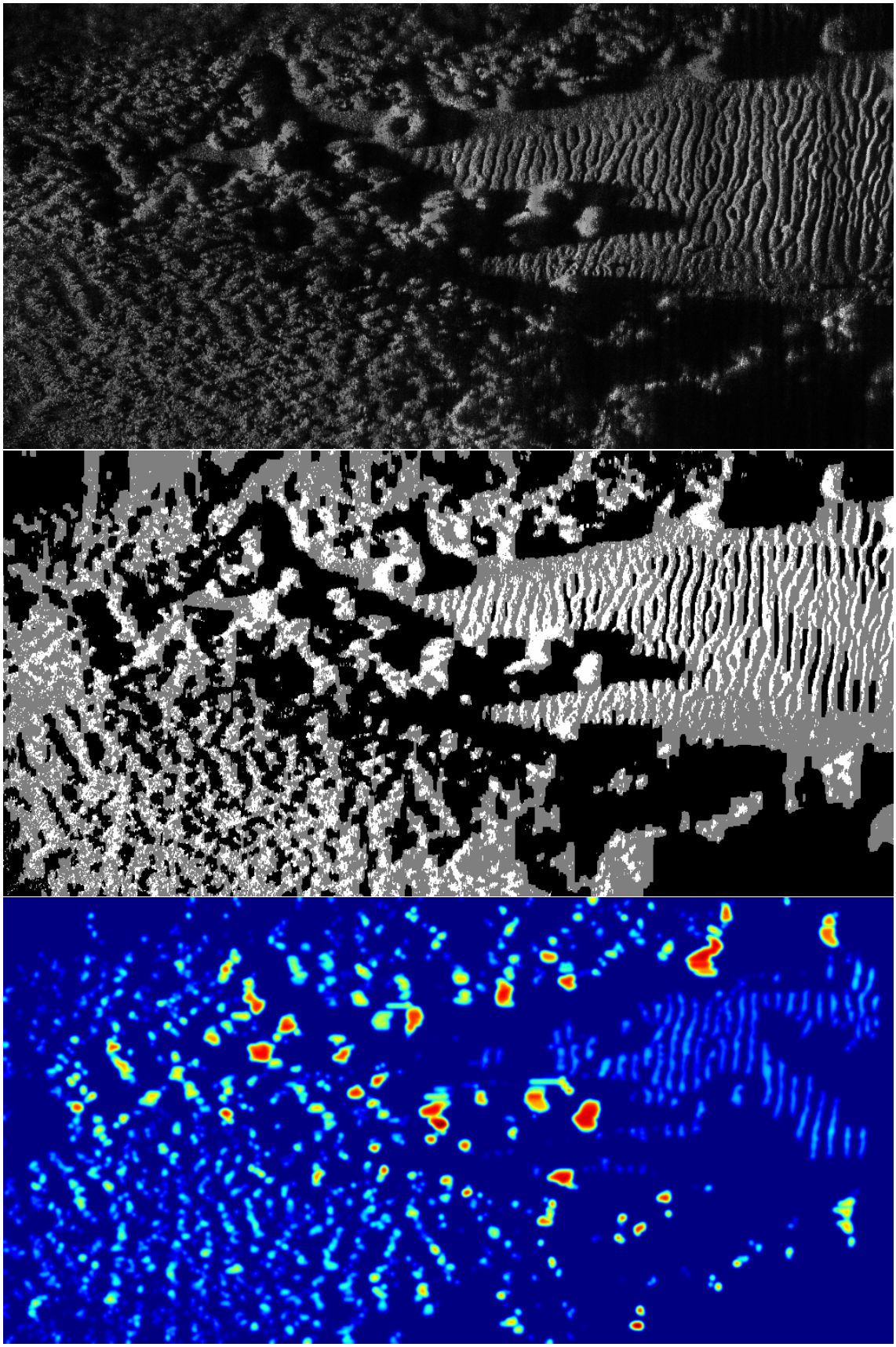


Figure 5.39: Example of the height map for a MUSCLE SAS image. Original image (Top), Segmented image (middle), Height map (bottom) displayed using Jet(0m,3m)

Using the DT$\mathbb{C}$WT we can measure the distribution of the energy, in the sonar image, with respect to spatial scale and direction. The Anisotropy measures the degree of order in the texture by measuring the standard deviation of the energy, at each scale, with respect to direction. The Complexity measures the sum of the mean energy of the texture at target-like scales. The Scale measures the spatial scale which contains the maximum signal energy.

Both the Anisotropy and Scale are designed to be independent of the sonar type and common pre-processing methods such as contrast stretching. The Anisotropy is normalised by the mean local energy, therefore a change in contrast does not change the ratio of the standard deviation to mean energy. The Scale is only dependent on the relative energy between scales, therefore a change in contrast has no effect on the Scale. The Complexity is dependent on the contrast. The results in section 5.2 show that while the Complexity can be used to identify complex regions from a single sonar type, the Complexity is not stable between different sonar types. Both the Anisotropy and Scale have relatively stable distributions with respect to different sonar types. As we were not able to survey the same region with each sonar type we were unable to make any statistically significant conclusions from the characteristic histograms. However, in section 5.5 we showed that a single threshold on the Scale and Anisotropy was optimal for all sonar types.

In section 5.5 we showed how the sea-floor characteristics could be used to segment a sonar image. We compared two computationally efficient approaches. First we applied a threshold to the local average of the sea-floor characteristics. Next, we introduced a more complex method using graph-cuts. Due to the stability of the characteristics between sonar types, we were able to identify rippled and complex regions with just two thresholds on the Scale and Anisotropy. Similarly in the graph-based approach we used just two functions to calculate the data term.

The quality of the segmentation using the local average agreed to within 1 standard error with the quality of the graph-cuts segmentation. However, we noticed a significant improvements in the quality of the segmentation of the edges using the graph-cuts approach. This was due to the fact that the smoothness term in the graph-cuts was only dependent on the Scale characteristics. From section 5.2 we can observe that the edges in the Scale image are much sharper than the Anisotropy image.

Finally we considered how to characterise the appearance of objects within a textural region. The height of objects can be calculated by segmenting the image into highlight, background and shadow regions and measuring the length of the shadow. Applying a Gaussian blur to the height-map and finding the local maxima allowed us to count the number of objects with a similar height to a MLO.

In summary we have developed a novel method for characterising the sea-floor. We are able to identify flat, rippled and complex regions, segment these regions into poly-

gons and measure the clutter density of MLOs in these regions. The characteristics are computationally efficient and using the simplest approach we are able to characterise and segment a MUSCLE SAS image in 0.3s. As a result we can implement our algorithm on current generation AUVs without using a significant proportion of the available processing power. In the next chapter we show how these characteristics can be used to reduce false alarms in ATR algorithms. In future work we intend to investigate the application of the characteristics to a fast supervised sea-floor classification algorithm. Additionally we intend to explore the use of the characteristics for image registration and Simultaneous Localisation and Mapping (SLAM).

Chapter 6

Environmentally Adaptive Automatic Target Recognition

6.1 Introduction

In the previous chapter we introduced algorithms to express the characteristics of the sea-floor in Sidescan Sonar (SSS) and Synthetic Aperture Sonar (SAS) images. We discussed various applications, such as driving contextual behaviours for Autonomous Underwater Vehicles (AUVs) and limiting false alarms from Automatic Target Recognition (ATR) algorithms in textured regions. In this chapter we consider the second application in more detail and introduce a novel approach to filter the detections from an ATR algorithm using the sea-floor information.

In chapter 4.6 we demonstrated that the performance of ATR algorithms is dependent on the environment in which they are deployed. Mine Like Textures (MLTs) and regions with high clutter density cause ATR algorithms to produce large numbers of false alarms [4]. We know from previous chapters that ATR algorithms are capable of obtaining close to 100% detection in flat regions with negligible false alarm rates. However, this requires a very low decision threshold and admits a large number of false alarms in cluttered and rippled regions. Without additional processing, the high Probability of False Alarm (PFA) would eliminate the time saving benefits from ATR aided Mine Counter-Measure (MCM) missions.

A number of approaches have been suggested to solve this problem. Perhaps the most promising, is training algorithms on augmented reality data [18]. This allows the ATR algorithm to specialize on a specific type of sea-floor. In theory, the algorithm could be trained in-situ to adapt to the specific sea-floor types that are most problematic. However, to date the augmented reality approach has only been used for off-line training and performance analysis.

Other problematic sea-floor types, such as ripples, exhibit regular patterns which can

be predicted [109] and removed from the image. This can be achieved either with adaptive mission planning [3] or with image processing [44]. Nelson et al.[44] report a significant reduction in false alarms when wavelet thresholding is used to filter ripples from a SAS image. An alternative approach, investigated by Valeyrie et. al [115], looks at filtering the contacts returned by an ATR algorithm. Each target is compared to its surroundings to produce a measure of the saliency of the target. A threshold on the saliency is applied to reject targets in difficult regions.

The use of background context for classification has been explored more recently in other domains such multi-spectral imagery[125, 126, 127]. Williams et al. [7] have shown, using our previous work on Anisotropy and Complexity [120], that sea-floor characteristics[121] can be used to train a context dependent ATR algorithm. As described in the previous chapter, our work in [120] is superseded by the work in this thesis and is not repeated here. The approach of Williams et al. is similar to that for multi-spectral imagery, except the contextual features have been replaced with sea-floor characteristics. In this approach an ensemble of classifiers are trained with each classifiers specialised for a specific sea-floor type. When the algorithm is applied to an image the classifiers are weighted according to the similarity of their training data to the target environment. Unfortunately, we require an algorithm that will work with closed source commercial ATR algorithms that may have been trained by a 3rd party.

In this chapter, we assume that there exists a trained ATR system, and that we have no access to the underlying algorithm. As such the technique can be applied in conjunction with any proprietary ATR algorithms that may already exist. The only requirement is that the ATR algorithm returns a confidence value for each contact that it has detected. We start in section by summarising the sea-floor characteristics introduce in the previous chapter. In section 6.3 we demonstrate that the sea-floor characteristics can be used to discriminate between regions where the ATR will perform well and those where there will be a large number of false alarms. Next, in section 6.4 we introduce a novel approach using a neural network to re-weight the ATR confidence value according to the local characteristics. In section 6.5 the Receiver Operator Characteristics (ROC) curves for the ATR algorithm are compared to the ROC curves for the ATR algorithm using the local sea-floor information. In section 6.6 we present the conclusions and plans for future work.

6.2 Summary of Sea-floor Characteristics

In this section we summarise the sea-floor characteristics that were introduced in the previous chapter. First we introduce the scale and anisotropy. These characteristics are calculated from the normalised wavelet energy $N_{s,k}$ where s is the wavelet scale and k is the wavelet direction. N_s is the average wavelet energy over all directions for scale S. Next we introduce the height-map for objects in the SAS image and finally the clutter

density.

6.2.1 Scale

Objects in SSS images produce a highlight-shadow pattern. For textures such as ripples and homogeneous clutter this pattern repeats with a characteristic frequency or scale S. (Fig 6.1c)

$$S = \underset{s}{\operatorname{argmax}}(N_s) \tag{6.1}$$

6.2.2 Anisotropy

The anisotropy of a texture is a measure of its directional homogeneity. Textures such as ripples have a maximum wavelet response in the directional sub-band parallel to the ripples and a minimum response orthogonal to the ripples. Anisotropic textures therefore have a high standard deviation when compared with non-anisotropic textures. The anisotropy A_s is normalised by the mean wavelet response, such that it is not dependent on the image contrast. (Fig 6.1b)

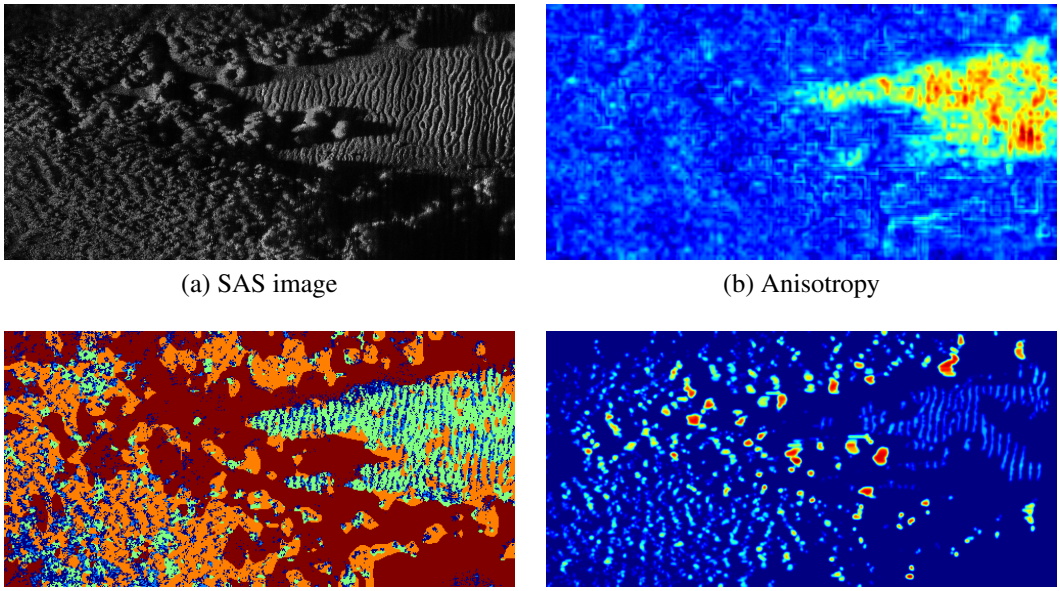
$$A_s = \sum_{s=2}^{n} \frac{\operatorname{stddev}_k(N_{s,k})}{N_s} \tag{6.2}$$

The smallest scale $s = 1$ is not included in the calculation of anisotropy as the response is dominated by speckle noise.

6.2.3 Height Map and Clutter Density

The height of an object in a SSS image can be estimated from the length of the shadow that it casts on the sea-floor. Each highlight pixel is matched to a set of connected shadow pixels. A set of connected shadow pixels can be assumed to be generated from a highlight pixel if they have at most 10 background pixels in-between them.

The height of an object can be calculated from equation . The height map is used directly to measure the height of each contact, this is used as a feature for the neural network. The density of objects that are of a similar height and size to a mine is also calculated. Individual objects are separated from the height map by applying a Gaussian blur to the image with a standard deviation similar to the size of the target object. Local maxima within the blurred image represent individual objects. The Mine Like Object (MLO) density can be found by counting the number of objects that have a height within $\pm 20\%$ of the target object height.



(a) SAS image (b) Anisotropy

(c) Scale (d) Height map

Figure 6.1: Example of the sea-floor characteristics. The characteristics are scaled between their minimum and maximum values and displayed using the standard 'Jet' colour map.

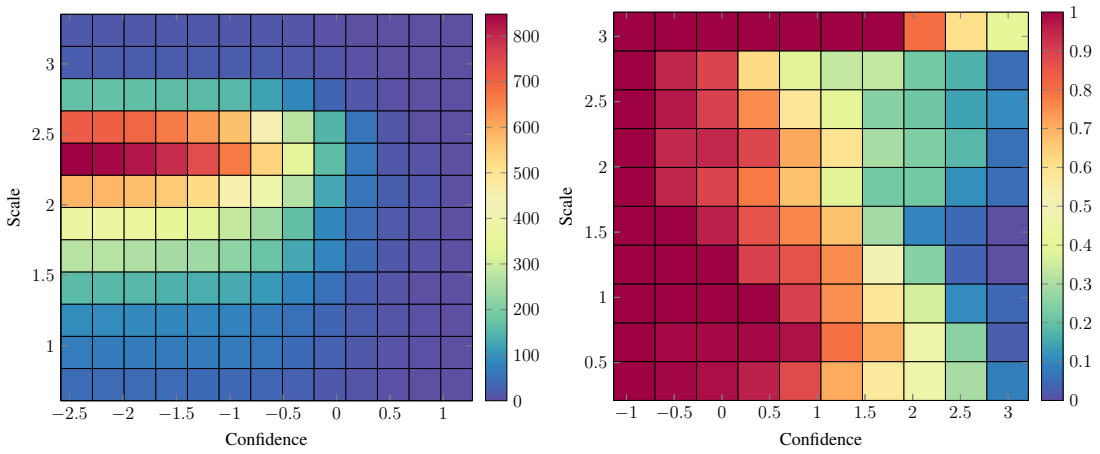
6.3 ATR Performance

Before we consider algorithms to reduce the false alarm rate from an ATR algorithm we first need to show that there is some dependence of the ATR results on the sea-floor type. In this section we investigate the relationship between the ATR performance and the local sea-floor characteristics. This investigation builds on the work that we presented in [120].

The Haar-cascade algorithm introduced in chapter 4.6 is applied to the MUSCLE SAS data set described in the previous chapter. This data set contains cylindrical, wedge and truncated cone targets deployed in flat, complex and rippled areas of the sea-floor. For each detection the classifier assigns a confidence value using the score cascade approach described in chapter 4.6. For the classifier used in this test the typical confidence values range between ± 3.

For each classifier we also calculate the average sea-floor characteristics in a 256×256 pixel window about the target. To prevent the target influencing the local sea-floor values we do not include the central 64×64 pixel window centred on the target. Therefore, for each detection we know the ATR Confidence, the Scale, Anisotropy, Clutter Density and Class of the sea-floor. Additionally we know the height of the detected object from the height-map described in the previous section.

Visualisation of the ATR performance with respect to the sea-floor characteristics is complicated by the fact that the number of false alarms detected and objects detected depends on the confidence threshold. In figures 6.2 - 6.6 we display 2D histograms of



(a) False Alarms

(b) Targets

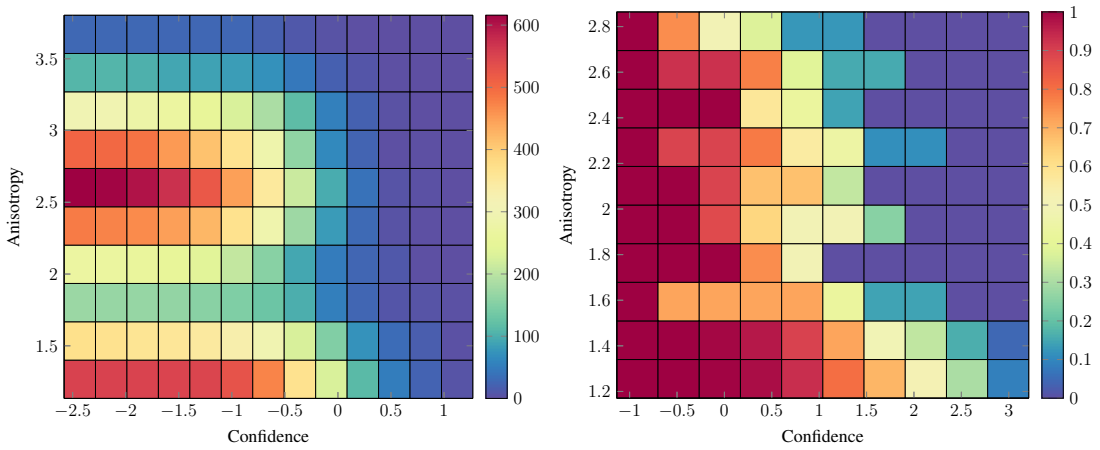
Figure 6.2: Histogram of ATR confidence values for false alarms (a) and targets (b) with respect to the **Scale** of the sea-floor. The histogram is cumulative from right to left on the x-axis and represents the number of false alarms and fraction of detections that would be retained for a given confidence threshold.

the ATR confidence with respect to the sea-floor characteristics. Detections are split into equally sized bins with respect to the sea-floor characteristic displayed on the y-axis. The histogram is cumulative from right to left on the x-axis. Each cell shows the number of detections that would be retained for the confidence threshold displayed on the left hand side of the cell. For the target histograms the results are normalised such that each row shows the fraction of targets detected for a given confidence threshold (x-axis) and feature value (y-axis).

For example if we consider figure 6.2(a), the figure shows the distribution of false alarms with respect to sea-floor Scale. As the threshold on the confidence value is increased the number of false alarms detected is reduced. On the Scale axis we can see that the greatest false-alarm density is found at scales higher than 1.5. From the previous chapter we know that sea-floor scales of 0-1 represents flat sea-floor. Therefore, the distribution of false-alarms agrees with our intuitive understanding that there are more false alarms in non-flat regions of the sea-floor.

Figure 6.2(b) shows the distribution of target confidence values with respect to scale. As the number of targets in each type of sea-floor is entirely dependent on the deployment pattern the fraction of target retained in each bin is shown, rather than the absolute number. Here, the most interesting result is that the classifier assigns a greater confidence to targets on flat sea-floor compared to those that are part of a textured region of the sea-floor. The anomalous result for scale = 3 can be attributed to the fact that there are limited target statistics for this row.

Comparing the histograms for target and background detections, we observe that the confidence value is actually doing a very good job of eliminating false alarms from tex-



(a) False Alarms

(b) Targets

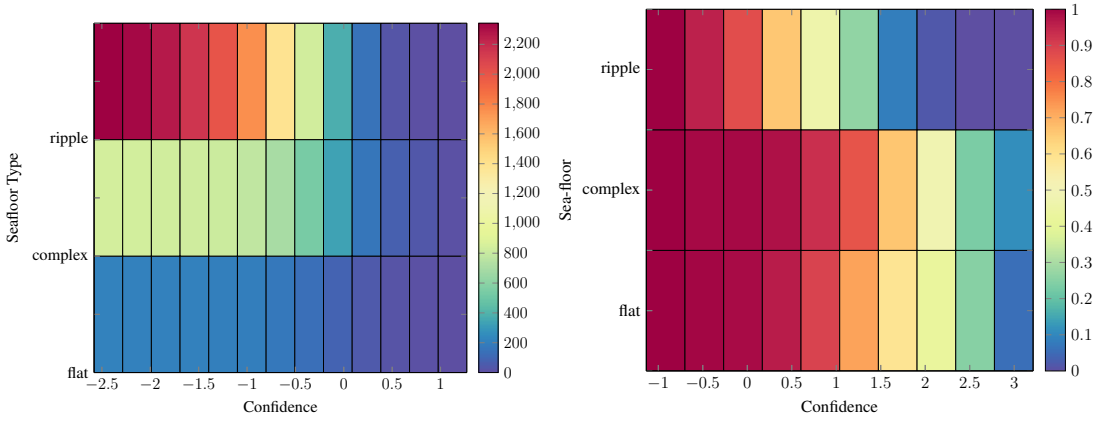
Figure 6.3: Histogram of ATR confidence values for false alarms (a) and targets (b) with respect to the **Anisotropy** sea-floor characteristic. The histogram is cumulative from right to left on the x-axis and represents the number of false alarms and fraction of detections that would be retained for a given confidence threshold.

tured regions. The majority of targets in flat regions have a confidence value greater that 0 while the majority of false alarms have a confidence of less than 0. Therefore, while we could remove the majority of false alarms in complex regions using the scale characteristic, we can already achieve a similar result using a threshold on the confidence value. Either approach will remove the majority of false-alarms and detections in complex regions.

Interestingly, other authors in the field have reached a similar conclusion. Valeyrie et al. identified false alarms by comparing the textural similarity of the detected patch to the surroundings with a texture measure based on [115]. While this approach successfully identified false alarms it duplicated the information provided by the confidence value and did not significantly improve the results.

It appears, on the basis of a single characteristic, that we are not able to remove false alarms in complex areas without also removing the real targets. However, we could improve performance in flat regions by setting adaptive confidence thresholds based on the sea-floor type. For example for scales less that 1 we could set a confidence threshold of -1 without significantly affecting the over-all false alarm rate. Before we discuss approaches to adaptively change the confidence threshold we first examine the confidence distribution with respect to the other sea-floor characteristics.

Figure 6.3 shows the distribution of confidence values with respect to the Anisotropy of the sea-floor. The two main concentrations of false alarms are in highly ordered regions such as ripples and cluttered irregular regions such as rocky areas. When we separate the detections by sea-floor class as in figure 6.4 it is clear that the majority of the false alarms are produced from rippled regions of the sea-floor. We also observe a reduction in the



(a) False Alarms

(b) Targets

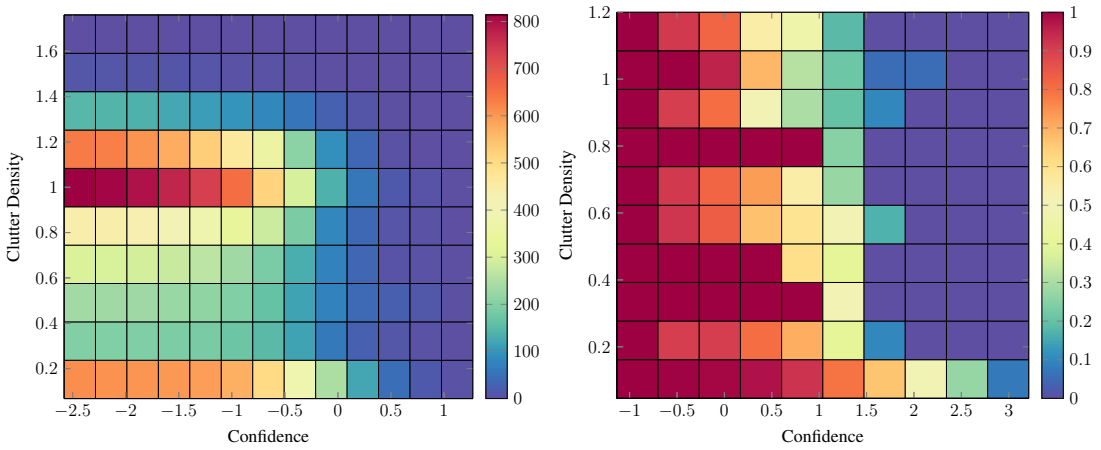
Figure 6.4: Histogram of ATR confidence values for false alarms (a) and targets (b) with respect to the **sea-floor type**. The histogram is cumulative from right to left on the x-axis and represents the number of false alarms and fraction of detections that would be retained for a given confidence threshold.

confidence values for real targets in these areas.

In figure 6.5 the effect of clutter density on the false alarm rate is explored. Unsurprisingly the majority of false alarms are in cluttered regions. Finally in figure 6.6 we compare the height of the ATR detections. Interestingly the majority of the false alarms are not of a similar height to the targets. This suggests that the particular classifier we are using does not use the height, or shadow length, as a discriminant feature. This would tend to agree with our conclusions in chapter 4.6. The Haar cascade does not use the shadow length as a feature, i.e. it does not select features which measure shadow length, as it can't model the change in shadow length with respect to range. Therefore, it appears for this particular classifier we may achieve some increase in performance by using the shadow length as a feature.

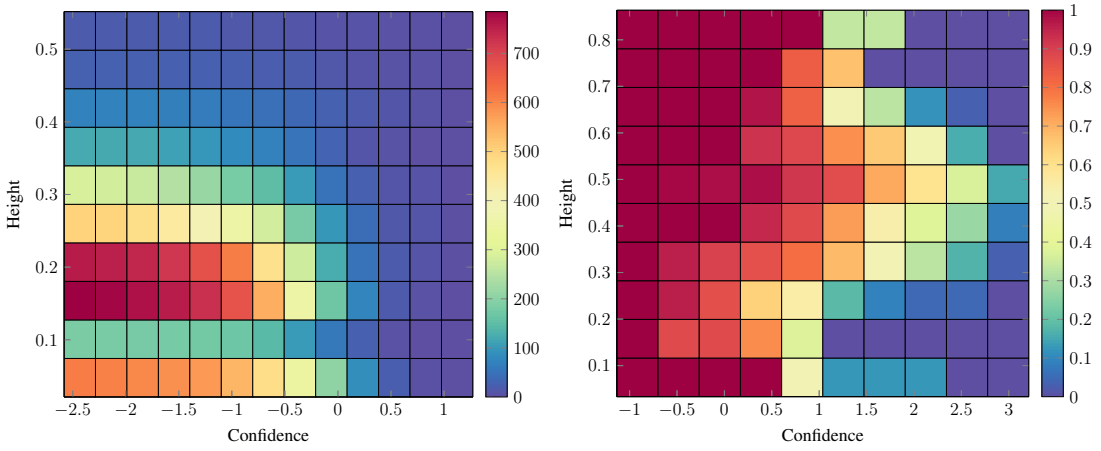
In this section we have demonstrated that the number of false alarms is related to the sea-floor characteristics. However, it is not clear if this information can be used to improve the overall classification performance. While many of the false alarms could be rejected on the basis of a single sea-floor characteristic they can also be rejected on the basis of the ATR confidence. The most promising observation from this section is that there are certain types of sea-floor where the number of false alarms is very low. For example flat, ordered sea-floor. If we can vary the confidence threshold independently for each region then it is possible to detect the difficult (low confidence) targets in flat regions without introducing a large number of false alarms in difficult regions.

While this would be simple to implement on the basis of a single characteristic, setting adaptive thresholds based on multiple sea-floor characteristics is more complicated. For this reason we depart from the concept of adaptive thresholds and train a neural network to determine the optimal performance in each region. This process is described in the next



(a) False Alarms (b) Targets

Figure 6.5: Histogram of ATR confidence values for false alarms (a) and targets (b) with respect to the **clutter density** sea-floor characteristic. The histogram is cumulative from right to left on the x-axis and represents the number of false alarms and fraction of detections that would be retained for a given confidence threshold.



(a) False Alarms (b) Targets

Figure 6.6: Histogram of ATR confidence values for false alarms (a) and targets (b) with respect to the **height** of the object. The histogram is cumulative from right to left on the x-axis and represents the number of false alarms and fraction of detections that would be retained for a given confidence threshold.

section.

6.4 Method

In this section we introduce an approach to reduce the number of false alarms detected from an ATR algorithm using the local sea-floor characteristics. In the previous section we observed that some combinations of sea-floor characteristics are associated with a large number of false alarms. However, the relationship between PFA and sea-floor characteristics is complex and non-linear. Therefore, we train a machine learning algorithm to distinguish between false alarms and real targets using the local sea-floor characteristics. In the remainder of this section we first introduce the machine learning algorithm and discuss the weighting of the training samples. Next we summarise the MUSCLE SAS data-set. Finally we introduce the ATR algorithm on which we test the method and describe the experimental set-up.

6.4.1 Neural Network

A Multi-Layer Perceptron (MLP) Neural Network (NN) is trained to filter detections from an ATR algorithm. A MLP is a feedforward artificial neural network that we use to map the sea-floor characteristics and ATR confidence value to a single confidence value. The structure of a MLP NN is shown in figure 6.7. The MLP NN consists of layers of nodes called perceptrons. The output of each perceptrons is the sigmoidal response to the sum of each inputs. The sigmoidal function is referred to as the activation function $/phi(.)$. The activation function can be any continuously differentiable sigmoidal function. In our implementation we use $/phi(.) = \tanh(.)$. The input to the activation function is controlled by a weight for each input. It is these weights that are derived during the training of the neural network.

The MLP NN is trained through back-propagation. Each connection (i, j) in the network has an associated weight $w_{j,i}$. Learning is the process of updating these weights as the training data is processed. The weighting is based on the training error. The error in output perceptron j in the nth training sample is:

$$e_j(n) = d_j(n) - y_j(n) \tag{6.3}$$

where d is the target value and y is the value produced by the perceptron. Back-propagation is a minimisation of the function:

$$\mathcal{E}(n) = \frac{1}{2} \sum_j e_j^2(n) \tag{6.4}$$

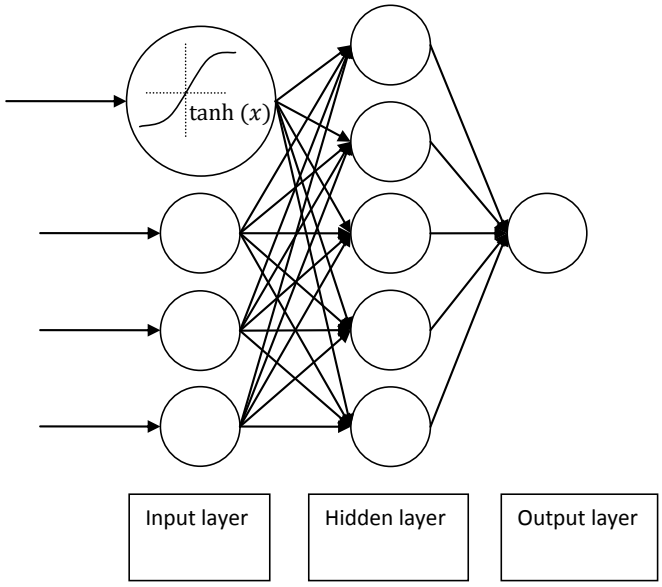


Figure 6.7: Structure of a MLP neural network

Using gradient descent, the change in each weight is:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \tag{6.5}$$

where y_i is the output of the previous neuron and η is the learning rate.

The derivative to be calculated depends on the induced local field v_j. For an output node this derivative is:

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n)\phi'(v_j(n)) \tag{6.6}$$

where ϕ' is the derivative of the activation function. For a hidden node the derivative is:

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n)) \sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} w_{kj}(n) \tag{6.7}$$

This function depends on the change in weights of the nodes in the output layer. Therefore, to calculate the hidden layer weights we must first calculate the weights for the output layer. This process is known as back propagation.

To generate the training data set the ATR algorithm is applied to a test mission for which the ground truth is known. The sea-floor characteristics are also calculated over the same mission. For each ATR detection we know the local sea-floor characteristics, the ATR confidence value and whether or not the detection is a target. The NN architecture is tuned depending on the number of training samples to prevent over-training. However, the machine learning algorithm is not the focus of this chapter. Therefore, apart from comparing the classification error over the training and test set to prevent over-training, there

was no attempt to optimise the performance of the NN filter. We test several combinations of features and training data sets which are described in section .

The NN is trained to minimise the classification error given the detections from another ATR algorithm and a set of sea-floor characteristics and a set of training labels. The classification error is 0 if the NN predicts the true class of the detection and 1 otherwise. Typically, the ATR algorithm will find more false alarms than targets. In the limit where all training sample are false alarms the NN can optimise the training error simply by rejecting every sample. To prevent this from occurring the positive and negative samples are normalised to have the same total weight.

6.4.2 Description of Data Set

The data sets, described in chapter 2, were acquired by the MUSCLE SAS system from CMRE and represent the state of the art in SAS imagery worldwide. The MUSCLE SAS has a resolution of 2.5cmx1.5cm in across and long track resolution and operates at 300KHz with a operational range of up to 150m. The data set consists of three different sea-floor regions (B, C, D), the data was provided by Defence Science and Technology Laboratory (DSTL) and we have retained their naming convention. Areas B and D are mostly flat, with area D displaying a significant number of marks from trawling. Area C is a mix of ripples, flat and rocky terrain and has a high density of MLOs. Areas B and D can be considered as easy regions where we expect close to 100% detection. Area C is extremely hard and we would expect a large number of false alarms and missed detections. DSTL has generated ROC curves for area C for a number of operators, these are included for comparison. In the data set there are approximately 400 cylinder, wedge and truncated cone targets (Some of these views represent repeat views of the same target from a different aspect).

6.4.3 Experimental Set-up

A Haar cascade ATR is trained to detect truncated cones, wedges, and cylinders in the data sets B-D. We use the approach of Sawas et.al. [5] and combine the output from the last 5 stages of the cascade; producing a continuous, rather than a binary, output. A real-valued continuous output is also a condition required for the neural network to adjust decision thresholds based on the sea-floor information. The Haar cascade is trained on 50% of the available targets, the same targets are used to train the neural network. The remaining 50% of the targets are used for testing.

A MLP neural network is trained on the textural and clutter characteristics and the confidence value provided by the ATR algorithm. The neural network has 2 hidden layers with 20 and 15 nodes respectively. The Anisotropy, Scale and MLO density are smoothed over a 100x100 pixel window centred on the target. The height map is not smoothed. Each

feature vector therefore has 5 elements consisting of [ATR output, Anisotropy, Scale, MLO density, Object Height]. An example of these features is shown in figure 6.1. The training process is summarised in figure 6.8

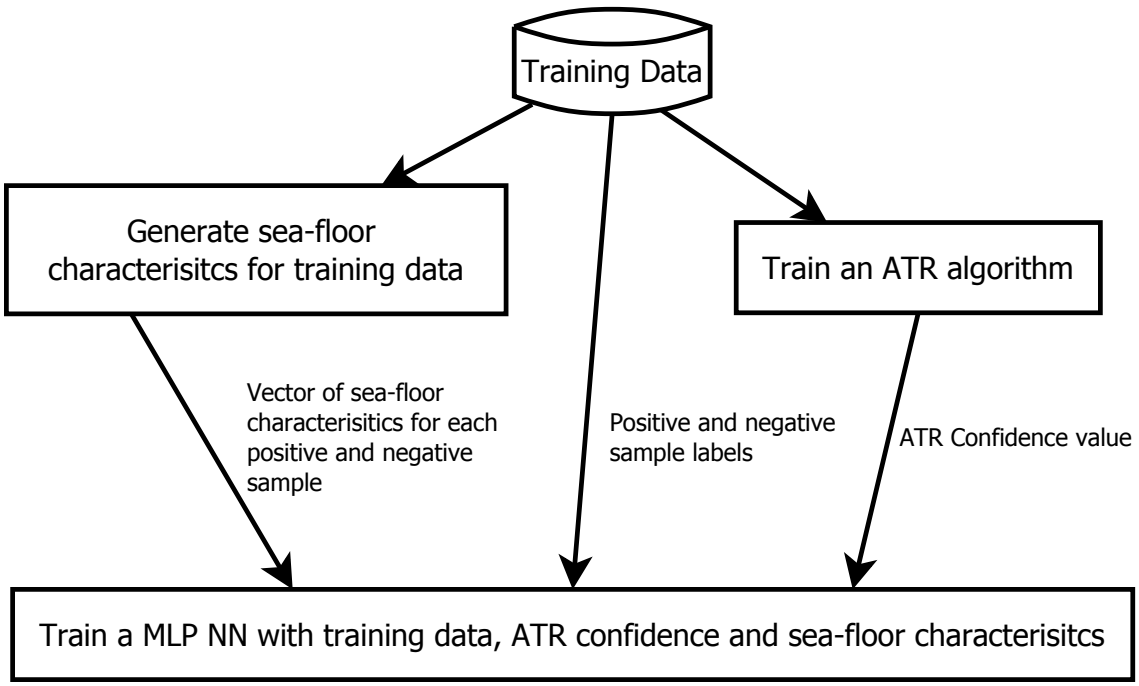


Figure 6.8: Flow diagram showing the training of the adaptive ATR algorithm.

The trained MLP NN is then used to classify ATR detections in the 50% of the data reserved for testing. As for the training data, sea-floor characteristics are calculated in a window about each detection found by the ATR algorithm. The MLP NN then returns a new confidence value based on a combination of the information in the ATR confidence value and the sea-floor characteristics. This is summarise in figure 6.9.

The Haar ATR algorithm detects approximately 4000 contacts over the 3 missions, 400 of these contacts are ground truth targets the remainder are false alarms. The neural network is first trained on 50% of the ATR detections from areas B-D and tested on the other 50%. Typically real world data is hard to obtain, therefore we also evaluate the performance on the algorithm when it is trained on just 5% of the available data. Finally a neural network is trained for comparison on just the ATR response and the sea-floor class.

6.5 Results And Analysis

In this section the results for the ATR are compared with the sea-floor filtered results. Figure 6.10 shows the ROC curves for all sea-floor types. Figures 6.11 and 6.13 show the results for rippled and flat sea-floor respectively. The total number of false alarms in the mission is shown on the x-axis, this can be converted to false alarms per km^2 by dividing by the total mission area of 0.1362 km^2. The ATR can be tuned to achieve a maximum

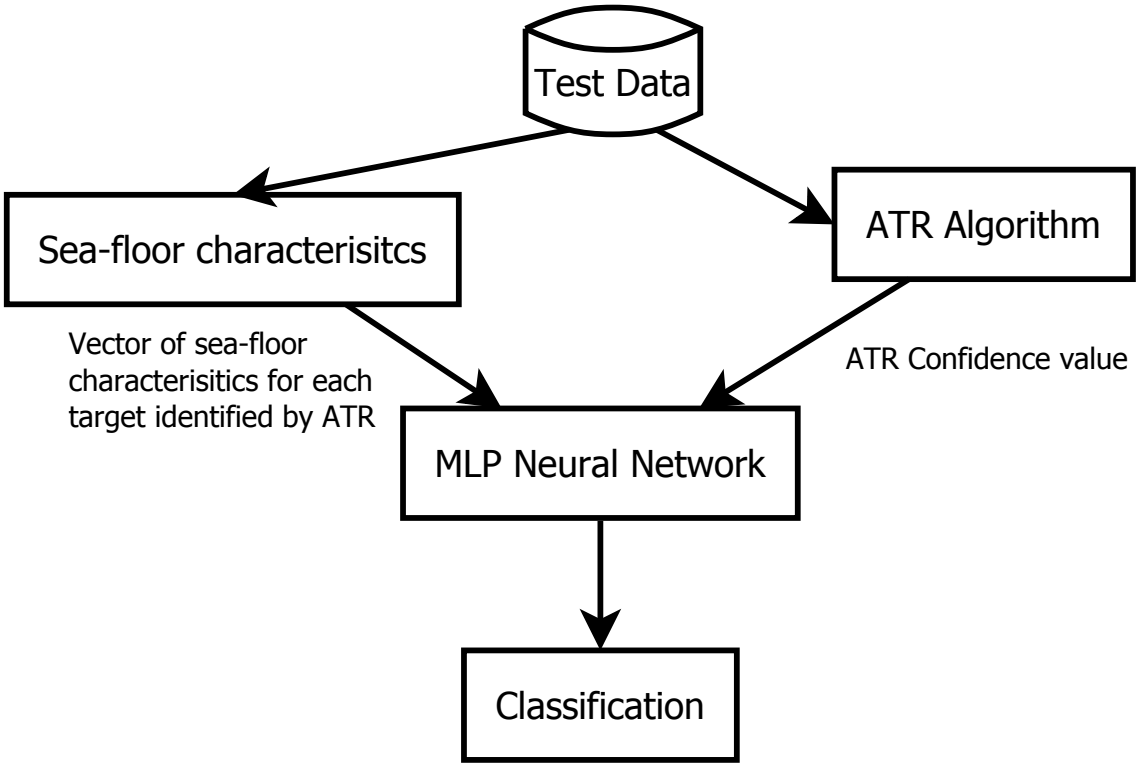


Figure 6.9: Flow diagram showing the application of the adaptive ATR algorithm to a test data set.

Probability of Detection (PD) of 95% for 3400 false alarms. The graphs in this section are truncated at 2500 false alarms in order to show the difference between methods.

The ATR is compared with the sea-floor filter trained on the 4 sea-floor characteristics, and the sea-floor filter trained only on the sea-floor class. These are referred to as ”Filter” and ”Filter (Class Only)” respectively. In figure 6.10 it can be observed that in the range between 85% and 92% PD the false alarm rate is reduced by more than 70%. The majority of the reduction in false alarms can be attributed to the performance of the sea-floor filter in rippled regions. Figure 6.11 demonstrates a reduction of over 90% in the false alarm rate.

For the complex regions shown in figure 6.12 the number of false alarms does not vary by more that 5% between the ATR, Filter and Filter (Class Only) for a given PD. This is not a surprising results as the ATR performance is already very good in these regions. Additionally the false-alarms with the highest confidence are visually similar to the real targets.

Results are also shown for the neural network trained on only 5% of the available data (Filter 5% of data). While the performance of the filter is reduced, the drop in performance is not as significant as might be expected. Although target statistics are limited to 20 real targets, the algorithm is trained on all the targets detected by the ATR algorithm. Therefore 5% of the data still represents 200 training objects.

The reduction in the number of false alarms, seen in figure 6.10, cannot be explained

solely by the reduction in false alarms in rippled regions. If we ignore the impact of the height map and consider only the sea-floor class, there are two effects that can improve the false alarm rate. The first is a re-weighting of the ATR output between different sea-floor types. The second is the optimisation of the decision threshold within a sea-floor type. The second effect is possible because the neural network allows the ROC curve to be optimised for a small range of sea-floor characteristics without considering false alarms from the other regions. While the results from Filter (Class Only) provide insight into the mechanisms for reducing false alarms, it is clear that the information provided by the clutter density estimate and the height map significantly improves the PFA

The most significant result for MCM operations is contained in figure 6.13. Here, it can be seen that the ATR performance matches that of the human operator on flat sea-floor. Other ATR algorithms tested by DSTL were even better, obtaining close to 100% detection of targets. Using sea-floor characterisation and segmentation the ATR can process the flat areas at 100% PD while the more difficult regions can be processed at 80% PD and marked for review by an operator. An interesting side effect of the algorithm, was that the majority of false alarms retained, looked like MLOs. The majority of false alarms removed, were clearly from ripples or MLTs. The improvement in the quality of false alarms is significant for increasing MCM operator trust in ATR algorithms.

However, following our analysis of the algorithm we have identified a problem with the weighting of the training samples. We do not account for the fact that the relative distribution of sea-floor types is specific to this mission. The classifier has learnt to minimise the classification error for the particular distribution of sea-floor regions and distribution of training samples. For example no targets were deployed in the rocky regions of sea-floor. Therefore, the classifier can maximise the classification error by excluding all targets in areas with rocky characteristics. This is a problem if we aim to apply the NN filter to previously unseen data. Additionally it makes it difficult to predict the performance of the algorithm in a new area. Fortunately, it is relatively simple to correct for this problem. We can correct for the target deployment by inserting evenly distributed augmented reality targets into the training data. We can adjust the background weights by creating an n-dimensional histogram of sea-floor characteristics and increasing the weight of the training samples relative to the sparsity of the data.

6.6 Conclusion

In this chapter we have introduced a novel approach to reducing the PFA in an ATR algorithm. Local sea-floor characteristics are used to weight the ATR output based on the difficulty of mine hunting in a region. We have demonstrated that a neural network trained on the detections from an ATR, and the sea-floor characteristics, can dramatically reduce the PFA. For the Haar cascade ATR tested, the false alarm rate was reduced by

over 70% at a 90% PD. The sea-floor characterisation is fast, adding less that 1 second to the processing of a 8000x3000 pixel SAS image.

It was shown that the reduction in false alarms is primarily due to the re-weighting of the decision threshold within a region. A lesser effect was seen due to the re-weighting between sea-floor regions. This suggests that our approach can be used to reduce the false alarm rate for any ATR algorithm which produces a real-valued continuous output.

The computational performance of the sea-floor characterisation algorithm is excellent and adds approximately 1 seconds to the processing of an 8000x3000 pixel SAS image. The Haar ATR algorithm takes approximately 0.5 seconds to detect the three object types. The combined approach is faster than real-time, and is therefore suitable for use on autonomous vehicles.

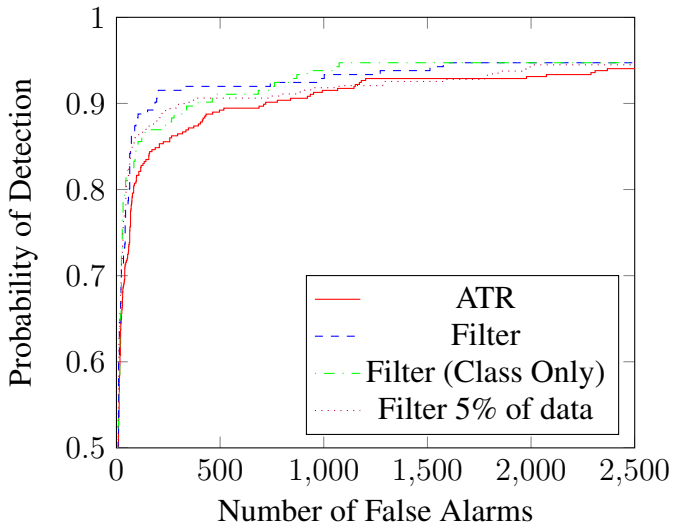


Figure 6.10: Results for all sea-floor types. ROC curves for the ATR and the ATR with the sea-floor filter for regions B-D.

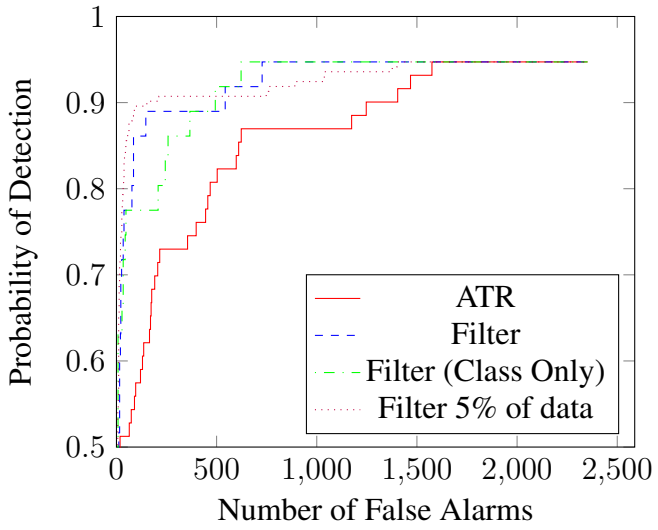


Figure 6.11: Results for rippled regions only. ROC curves for the ATR and the ATR with the sea-floor filter for regions B-D

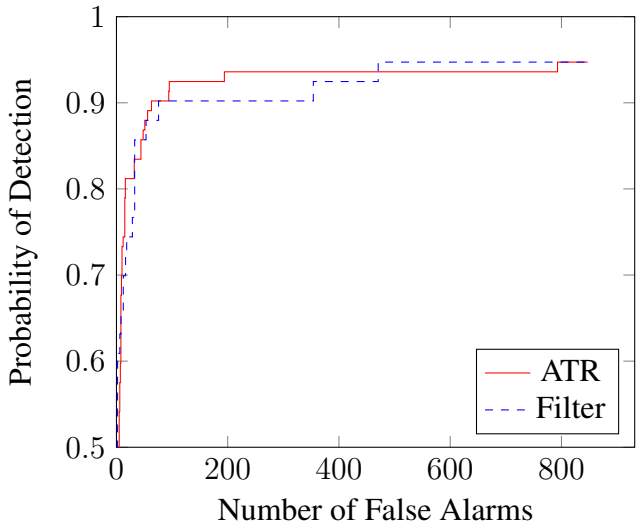


Figure 6.12: Results for complex regions only. ROC curves for the ATR and the ATR with the sea-floor filter for regions B-D.

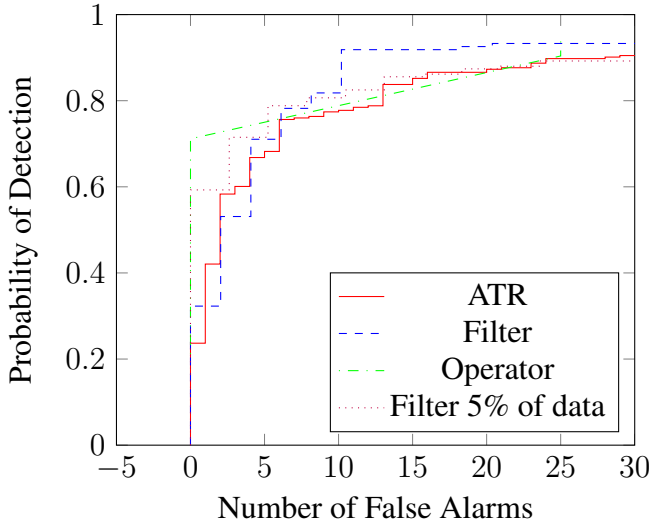


Figure 6.13: Results for flat regions only. ROC curves for the ATR and the ATR with the sea-floor filter for regions B-D. Results from an operator in region B are shown for comparison

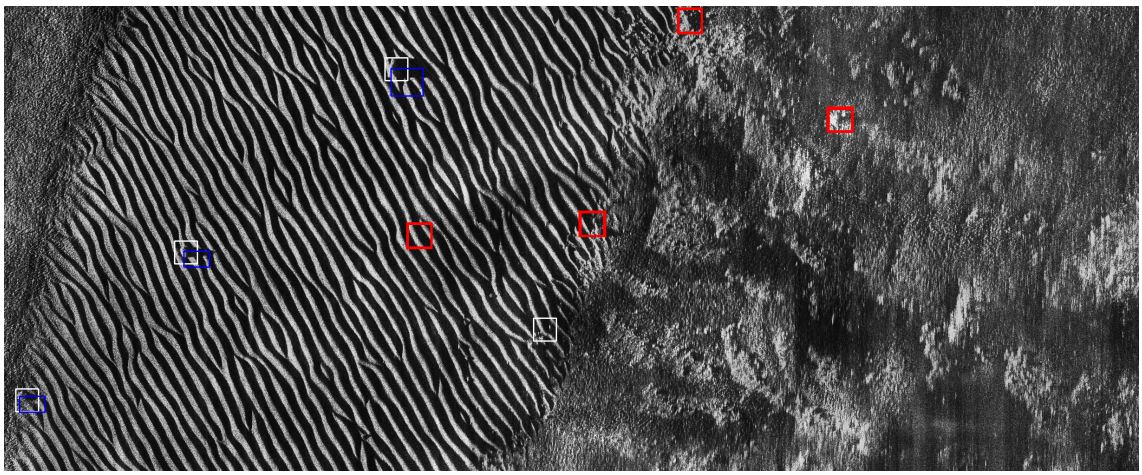


Figure 6.14: Results from an ATR trained to detect truncated cones, cylinders and wedges with a contextual sea-floor filter used to remove false alarms in difficult sea-floor regions. White boxes are detections retained by the filter, red boxes are objects removed by the filter, blue boxes are ground truth.

Chapter 7

Summary and Conclusions

7.1 Summary

This doctoral thesis has introduced novel methods for improving the performance of Automatic Target Recognition (ATR) algorithms in sonar images. Specifically we focused on the task of detecting Mine Like Objects (MLOs) in Sidescan Sonar (SSS) and Synthetic Aperture Sonar (SAS). To this end, we explored topics including pre-processing, machine learning and textual analysis of sonar images. In this chapter we restate the key results and the contribution of this thesis to the field.

First we restate the motivation for the work, previously introduced in chapter 1. State of the art sonar ATR algorithms produce excellent results in flat and moderately complex terrain [5]. However, under certain conditions, the performance can be degraded. As a result, real targets can be missed or large numbers of false alarms detected. This is problematic both for completely autonomous, and operator aided, Mine Counter-Measure (MCM) missions.

In the short to medium term operator aided MCM missions are likely to be the standard mode of operation. Therefore, commercially, there are two pressing issues. The first is that of gaining operator trust in autonomous algorithms in regions where the algorithm is known to perform well. The second is reducing the costs associated with data collection for training ATR algorithms. These problems were addressed in chapters 2 to 4.6.

We first discussed algorithms for removing artefacts, which degrade ATR performance, from sonar images. These algorithms prevent artefacts that are easily ignored by a human, such as the surface return, from being identified as mines by ATR algorithms. Next we addressed the issue of ATR algorithms that are over-specialised on a single set of operating conditions or sea-floor type. We demonstrated that over-specialisation can cause unpredictable performance in data that is different to the training data. For example, if the operator runs an algorithm trained on data collected from one model of sonar on another model, the majority of target objects are not detected. Re-training ATR algorithms for every sensor and environment is prohibitively time-consuming and expensive. In chapter

4 we introduced a novel ATR algorithm that demonstrated a significantly reduced dependence on the sonar type. Additionally the algorithm performance was competitive with state of the art ATR algorithms.

More recently, unpredictable algorithm performance has been an issue for completely autonomous MCM systems such as the Neptune [3] system developed by SeeByte Ltd. Here, a large number of false alarms can eliminate many of the time-saving benefits gained by using co-operating fleets of autonomous vehicles. This issue was addressed in chapters 5 and 6. First we identified sea-floor characteristics that could be used to identify regions of sea-floor that are difficult for ATR algorithms. Next we introduced a machine learning approach that can be used to tune the ATR performance such that in difficult areas only the targets with the highest confidence scores are reported.

In the remainder of this chapter we summarise each topic, state the novel contributions, the key results and discuss potential improvements and further work.

7.2 Conclusions

In this section we present the key conclusions from the thesis. The conclusions are split into three main subject areas "Imaging Artefacts and Preprocessing", "Automatic Target Recognition" and "Sea-floor classification".

7.2.1 Imaging Artefacts and Preprocessing

In chapter 2 we discussed the physical process of forming sonar images. Three main sources of image degradation were identified: absorption losses, beam pattern and surface return. Absorption losses and beam pattern have been studied extensively in the existing literature. In chapter 2 we implemented and compared a number of methods for correcting these effects. The Serpentine Forward-Backward Filter (SFBF) approach of Dobeck et al. [22] was found to normalise the image intensity whilst introducing the least mine-like artefacts. However, for applications where background features such as ripples need to be preserved, the approach of Capus et. al [11] was shown to be superior.

The impact of the surface return on ATR algorithms has been identified in a number of papers [24]. However, there exist few algorithms to remove them from the image. We demonstrated that one existing approach [29] breaks down in complicated images. Therefore, we introduced an A* search algorithm to track the surface-return. This enabled us to model our prior knowledge that the vehicle depth is relatively constant in a standard MCM survey. Consequently, the algorithm correctly identified the surface return in 90% of the test images. This included images with complex saturated patterns from marine growth and ripples. In images where the approach broke down this was always due to

surface return intersecting the saturated band of pixels caused by the beam pattern. While we could have approached this problem by first correcting for beam pattern, these regions are typically not processed by ATR algorithms.

In the second half of the chapter we addressed the impact of the image formation processes on Markov Random Field (MRF) segmentation of sonar images. We introduced a novel method using graph-cuts and locally adaptive class models to significantly improve the visual quality of segmentation of difficult SSS images. We addressed two problems identified by previous authors [24, 31, 32, 33, 34, 35]. Firstly. traditional approaches to solving MRF problems, such as Iterated Conditional Modes (ICM) and Markov Chain Monte Carlo (MCMC), are a compromise between computational efficiency and convergence to local minima. Using graph-cuts we are able to calculate the global minimum energy state of the MRF in a computationally efficient manner. Additionally by introducing a novel approach using local class models we removed the range dependent segmentation quality observed in other approaches.

7.2.2 Automatic Target Recognition

In chapters 3 - 4 we examined the problem of over-training in sonar ATR algorithms. Algorithms which calculate features directly from the image, such as [5, 76], are particularly prone to over-training. However, these algorithms are of interest as they typically outperform algorithms that segment the image before classifying the object[24, 31, 50].

In chapter 4 we demonstrated the problem of over-training for the Haar-Cascade algorithm. We showed that the Haar-Cascade is sensitive to changes in the Environmental and Operating Conditions (EOCs) and to the appearance of the target object with respect to pose. This causes two major problems for the commercial application of the Haar-Cascade algorithm. Firstly, a large number of real target images are required to train the algorithm. For an asymmetric object, such as a cylinder, 1000 target samples were required to prevent over-training. While we were able to produce this number of samples using an augmented reality simulator [18], typically only large navies have access to this much data. The second problem is that the algorithm may need to be retrained for different operating environments. Consequently new data must be collected and the performance of the algorithm must be re-evaluated.

In chapter 4 we introduced a novel algorithm to address the problems raised in chapter 4. In chapter 3 we observed that algorithms which first segment the image and then extract geometric features are less prone to over-training. This can be explained by the fact that the geometric features such as the target area and height are independent of rotation and the sonar type. Consequently, the machine learning algorithm has to solve a much simpler problem. Unfortunately, these algorithms lose a significant number of targets due to poor segmentation. Therefore they can not compete with the performance of algorithms such

as the Haar-Cascade.

Our 3D features algorithm reduces the impact of these problems by calculating rotation and EOC invariant features directly from the image. This is accomplished by adapting a number of techniques from template matching. The first novel contribution is that we remove the need for a prior object model by defining a large number of boxes with different dimensions. These are referred to as generating boxes. Unknown objects can be described by a combination of many generating boxes. The specific combination is learnt by the algorithm during training. Each feature is described uniquely by the size of the generating box. Therefore, once we have trained a classifier on one set of EOCs we can adapt to a new set of EOCs simply by defining a new projection model for the generating boxes. The second novel contribution is the introduction of the integer-skewed integral images. This accelerated the calculation of the template matching stage of the algorithm and enables the large number of calculations required to be performed in real time (less than the time required to obtain a SSS image).

We first demonstrate that the algorithm matches the performance of the Haar-Cascade algorithm for a number of different target and sonar types. However, unlike the Haar-Cascade the 3D features algorithm requires fewer training samples. For the asymmetric cylinder there was no significant change in performance for 1000 or 500 training samples and the algorithm still achieved a maximum Probability of Detection (PD) of 80% with 250 training samples. Additionally we observed that the 3D features algorithm used approximately half the number of decision trees compared to the Haar-Cascade. This significantly reduces the probability of over-training. Finally we demonstrated that there is a much smaller reduction in performance when the algorithm is trained on one sonar type and tested on another. In the case of the symmetric cone object the maximum PD was improved by 30% compared to the Haar-Cascade.

The 3D features algorithm represents an incremental improvement on the state of the art for ATR in sonar images. However, it is significantly more difficult to implement than the Haar cascade and has approximately 20 times the computational requirements. Additionally, for large images the memory requirements are significant.

7.2.3 Sea-floor classification

In chapters 5 and 6 we focussed on the performance of ATR algorithms in difficult terrain. First we addressed the relationship between ATR performance and sea-floor texture in a conference paper [120]. In chapter 5 we extended this work to develop a novel set of characteristics for identifying difficult terrain. Finally, in chapter 6 we developed a machine learning approach to reducing the number of false alarms generated from difficult terrain.

Our first novel contribution to the field was to develop a simple and robust set of sea-

floor characteristics. The first two of these characteristics, the anisotropy and scale, are calculated using the dual-tree wavelet transform. The dual-tree wavelet transform is a computationally efficient approach for generating strongly oriented, multi-scale wavelets. Using this approach we derived metrics for the dominant scale of the texture and the degree of order, or anisotropy, of the texture. We showed that both the scale and anisotropy of a texture have a significant impact on the performance of the Haar Cascade ATR algorithm. More importantly we showed that, unlike characteristics such as the wavelet energy, these characteristics are very stable between different sonar types. Additionally, due to their independence from image contrast, the characteristics are invariant to common pre-processing problems such as range dependent absorption.

Next, we used the two wavelet characteristics as input to a graph-cut segmentation algorithm. Importantly, due to the stability of the characteristics the segmentation algorithm did not require tuning for different sonar types. The remainder of chapter 5 addressed the problem of identifying the mine-like clutter density within a region of sea-floor. Here, we used the sonar image segmentation approach developed in chapter 2 to detect objects of a similar height to the target of interest. The correlation between these characteristics and the ATR performance was then addressed in chapter 6.

In chapter 6 we developed an approach to reduce the number of false alarms in difficult terrain. A similar approach was also developed by Williams et al. [7] using our earlier work [120]. In their approach they demonstrated that wavelet based meta-features could be used to improve the performance of ATR algorithms in difficult terrain. However, in their approach the meta-features are required at the training stage. In our work we addressed the problem of filtering ATR results from a previously trained ATR algorithm.

Out approach was to train a Multi-Layer Perceptron (MLP) Neural Network (NN) to filter detections from an ATR algorithm. To generate the training data set the ATR algorithm was applied to a test mission for which the ground truth was known. The sea-floor characteristics were also calculated over the same mission. For each ATR detection we then know the local sea-floor characteristics, the ATR confidence value and whether or not the detection is a target. This information was used to train the machine-learning algorithm.

The improvement in the performance of the ATR algorithm with the MLP filter was significant. We demonstrated that a neural network trained on the detections from an ATR, and the sea-floor characteristics, can dramatically reduce the Probability of False Alarm (PFA). For the Haar cascade ATR tested, the false alarm rate was reduced by over 70% at a 90% PD. However, we concluded that the excellent results were due to the classifier learning to minimise the classification error for the deployment pattern in the training and testing data. For example no targets were deployed in the rocky regions of sea-floor. Therefore, the classifier could minimise the classification error by excluding all targets in areas with rocky characteristics. As for the work on ATR these problems

could be solved with an accurate augmented reality simulator. If a uniform target density is simulated across the entire training data set then there is no longer any bias towards the number of targets deployed in each sea-floor region.

7.3 Future Work

In this section we examine the potential for future work. As in the previous section the observations are split into three main subject areas "Imaging Artefacts and Preprocessing", "Automatic Target Recognition" and "Sea-floor classification".

7.3.1 Imaging Artefacts and Preprocessing

While we implemented a range of pre-processing algorithms we did not implement a more recent method developed by Capus et al. [20]. In this method an iterative approach is used to reconstruct the physical factors degrading the sonar image. This approach allows the true reflectivity of the sea-floor to be reconstructed. As our work on sea-floor characterisation (chapter 5) is invariant to absolute intensity and contrast this is not currently important. However, if in future work we aim to include sediment characteristics, it may be worth implementing this approach.

There is little current research in the area of SSS image pre-processing, for the most part the challenges are considered solved. Instead, much of the research effort is focussed on the signal processing challenges in implementing SAS system [62]. However, the vast majority of active underwater vehicles still use SSS systems. Additionally, as completely autonomous vehicles are becoming a commercial reality, robust pre-processing algorithms are an essential component. As we have demonstrated there is often a large gap between algorithms that work on small data-sets and commercial algorithm that are robust in many different environments. To this end, both the A* algorithm for removing surface return and the graph-cuts segmentation approach represent important and novel contributions to the field. Both of these algorithms have been shown to be robust and are integrated with ATR and sea-floor classification products produced by SeeByte Ltd.

7.3.2 Automatic Target Recognition

In our work on 3D feature, we addressed the problem of changing EOCs by reducing the variance of the feature space on image formation model. However, we could also address this problem by reducing the impact of the EOCs on the image. Our proposed approach is to first perform a histogram transformation on the image. This transformation aims to match the image histogram to the mean training histogram. Next the sonar image is

resized such that it has the same resolution as the training images. Finally we address the problem of grazing angle and pose invariance. The naive approach to this problem is to train multiple classifiers. The training data for each classifier is separated by pose and grazing angle. However, this approach further increases the requirements for training data. Therefore, we propose an approach similar to that of [7] where sea-floor characteristics are used as meta-features to train an ensemble of classifiers. Instead of (or in addition to) using sea-floor characteristics as meta-features we would also use the grazing angle and object pose as meta-features.

Another interesting development during the last four years is the application of deep learning [128] to a number of large scale image recognition problems. Unfortunately, this approach is not currently applicable to the detection of mines in SSS due to the requirement for extremely large data-sets. This problem may be addressed by research into augmented reality simulators. While current simulators model many of the physical interactions in generating a sonar image they also use a number of simplifying assumptions. The result of these assumptions may be that the deep-learning algorithms over-specialises on the simulated data. Further research is required to produce a simulated sonar target which is indistinguishable from a real target.

An open-source sonar simulator would also address the main problem with the current literature on ATR in SSS images. Currently, each research group has access to a small data set which is often classified or can not be shared for commercial reasons. As a result an unbiased comparison of algorithms is difficult and requires a 3rd party with access to large quantities of sonar data. Investment in an open-source SSS and SAS simulator would address this problem. Additionally a simulator would allow algorithms to be trained in-situ to cope with changes in environment or target type. These applications are discussed in more detail in the next section.

7.3.3 Sea-floor classification

In our work on sea-floor classification (section 4.9), we asserted that the features had a reduced variance with respect to the slant range at which the target was imaged. This was shown theoretically for a very simplified world model but it remains to be demonstrated for real data. In future work we aim to investigate the range dependence of the performance of the Haar-Cascade algorithm and we can use these analysis tools to compare the performance with the 3D features at the same time. A similar case is true for the rotation invariance. While we had significant evidence to suggest that we had improved the rotation invariance we did not produce the tools to test this experimentally. Finally, in this work we compared two models of SSS sonar that are arguably more similar than for example Marine Sonic and MUSCLE SAS. To prove the work in this chapter conclusively we would like to repeat the work on SAS data. However, the current manual training and

testing process is prohibitively labour intensive therefore we would first invest some time in automated training and testing tools.

The work on sea-floor classification is applicable to reducing false-alarms both for human operators and autonomous missions. However, the main benefit is for autonomous missions. Assuming that the approach has been trained on representative data for the region we can limit the false alarms for complex regions without affecting the PD in flat regions. This allows all sea-floor types to be treated equally during the survey which simplifies control logic for the vehicles. In future work we intend to extend the analysis to predicting the performance of an ATR in previously unseen terrain. We also intend to explore the use of the height map to register SAS images for mosaicking and multi-view fusion. Finally we will use the sea-floor characterisation and segmentation to drive autonomous behaviours in mine hunting Autonomous Underwater Vehicles (AUVs).

The final question to ask is if this is the best approach to reducing the dependence of the classifier on the EOCs. For example we could remove effects such as the survey speed and sonar resolution by resizing the image to a constant resolution. Indeed the SFBF technique that we applied to remove sonar specific intensity changes from the image is not unique to the 3D features. However, it is difficult to see how effects such as range dependence could be removed at an image processing stage. While we could train multiple Haar classifiers specialised for a particular range this would further increase the number of training samples required and increase the probability of over-training. Additionally, it is unlikely that navies would be prepared to collect the large number of training samples that this approach would require. The alternative approach is to revisit model based approaches and improve the segmentation quality to increase the PD. Considerable advances have been made in image segmentation since model based approaches such as [24] were introduced. Indeed we demonstrate improvements in MRF segmentation using graph-cuts in chapter 2. The problem is by no means solved and there is plenty of scope for future research in this area.

7.4 Closing Remarks

In this thesis we have addressed a number of the current problems and challenges with ATR for MCM missions. We addressed the pre-processing of SSS images and introduced novel techniques to filter the surface return and segment sonar images. We introduced a novel ATR algorithm that significantly reduced the need for training data and generalised well to new sonar types. Finally, we introduced a novel technique for sea-floor characterisation which is significantly faster and more robust than existing techniques.

In our opinion, the largest barrier to the main-stream use of ATR algorithms is not the behaviour of the algorithms under difficult conditions but the way in which this behaviour

is managed. Indeed, unless there is a step change in the performance of ATR algorithms there will always be areas where they perform significantly worse than humans. The first challenge is to identify these areas, and this was addressed in chapter 5. The next challenge is to provide MCM operators with tools that can be used to manage the ATR algorithm output in these areas.

We believe that the most promising avenue of work in this area lies in the combination of accurate augmented reality simulators and on-line retraining [129] of the algorithm. A simple technique, which could be integrated into our work in chapter 4, is the TrAdaBoost [130] transfer learning algorithm. With this technique the operator could first review the images for which the ATR algorithm reports the most contacts. Once an image has been reviewed the ATR algorithm can be updated with the new results. In our experience the majority of false alarms in a mission are produced from a single type of texture. In preliminary (unpublished) work we found that retraining an ATR algorithm on just the 5 most difficult images in a mission could have a significant effect on the overall results.

To summarise, we believe that future gains in sonar ATR algorithm performance will be incremental at best. Therefore, future research should be directed towards gaining the best performance from the algorithm based on the local context. Whether this is best achieved through on-line retraining or context based training (or a combination of both) remains an open question.

References

[1] C. Shihua and L. Keyong, "Wavelet software." [Online]. Available: http://eeweb.poly.edu/iselesni/WaveletSoftware/ xiii, 133

[2] "SeeByte Ltd." [Online]. Available: http://www.seebyte.com/ 1, 8, 109

[3] SeeByte Ltd., "Neptune," 2014. [Online]. Available: http://www.seebyte.com/seetrack-neptune-software-successfully-integrated-into-the-oceanserver-iver2-vehicle/ 1, 73, 166, 184

[4] P. Chapple, "Automated detection and classification in high-resolution sonar imagery for autonomous underwater vehicle operations." in *Proc. Spie*, 2008. 2, 29, 165

[5] J. Sawas, Y. Petillot, and Y. Pailhas, "Cascade of boosted classifiers for rapid detection of underwater objects," in *Conf. on Underwater Acoustics*, 2010. 3, 6, 67, 68, 73, 74, 76, 175, 183, 185

[6] J. T. Cobb and J. R. Stack, "In situ adaptive feature extraction for underwater target classification," in *Proceedings of the Workshop on Applied Imagery Pattern Recognition*, 2007, pp. 42–47. 3, 67

[7] D. Williams and E. Fakiris, "Exploiting environmental information for improved underwater target classification in sonar imagery," *IEEE Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6284–6297, 2013. 5, 166, 187, 189

[8] M. Somers and A. Stubbs, "Sidescan sonar," in *Communications, Radar and Signal Processing*, vol. 131, no. 3, 1984, p. 243. 8

[9] C. Loggins, "A comparison of forward-looking sonar design alternatives," in *IEEE Oceans Conf.* Marine Technol. Soc, 2001, pp. 1536–1545. 8

[10] SeaBeam Instruments, "Multibeam Sonar Theory of Operation," Tech. Rep., 2000. [Online]. Available: http://www.ldeo.columbia.edu/res/pi/MB-System/sonarfunction/SeaBeamMultibeamTheoryOperation.pdf 8

[11] C. Capus, I. Ruiz, and Y. Petillot, "Compensation for changing beam pattern and residual tvg effects with sonar altitude variation for sidescan mosaicing and classification," in *European Conf. on Underwater Acoustics*, 2004. 8, 25, 184

[12] A. Martin, A. An, K. Nelson, and S. Smith, "Obstacle detection by a forward looking sonar integrated in an autonomous underwater vehicle," *MTS/IEEE Oceans Conf.*, pp. 337–341, 2000. 8

[13] N. Hurtos, X. Cufi, Y. Petillot, and J. Salvi, "Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing," in *2012 IEEE/RSJ International Conf. on Intelligent Robots and Systems*. Ieee, Oct. 2012, pp. 5298–5305. 8

[14] A. D. Waite, *SONAR for practicing engineers*, 3rd ed. Chichester, West Sussex: Wiley, 2002. 10, 11

[15] F. Maussang, J. Chanussot, and A. Hetet, "Automated segmentation of SAS images using the mean standard deviation plane for the detection of underwater mines," in *MTS/IEEE Oceans Conf.*, 2003, pp. 2155–2160. 11, 61, 86

[16] E. Jakeman and P. Pusey, "A model for non-Rayleigh sea echo," *IEEE Transactions on Antennas and Propagation*, vol. 24, no. 6, pp. 806–814, Nov. 1976. 11

[17] R. Hansen, *Introduction to synthetic aperture sonar*, InTech, 2011. [Online]. Available: http://www.intechopen.com/books/sonar-systems/introduction-to-synthetic-aperture-sonar 15

[18] P. Mignotte and J. Vazquez, "PATT: A performance analysis and training tool for the assessment and adaptive planning of Mine Counter Measure (MCM) operations," in *OCEANS MTS*, 2009. 20, 73, 79, 113, 117, 120, 161, 165, 185

[19] K. Benke and D. Hedger, "Improving feature perception in sonar displays by contrast normalisation and enhancement." *DSTO*, 1996. 21

[20] C. G. Capus, A. C. Banks, E. Coiras, I. Tena Ruiz, C. J. Smith, and Y. R. Petillot, "Data correction for visualisation and classification of sidescan SONAR imagery," in *SPIE Conf.*, vol. 2, no. 3, 2008, pp. 155 – 169. 21, 26, 87, 188

[21] P. Cervenka and C. Moustier, "Sidescan sonar image processing techniques," in *MTS/IEEE Oceans Conf.*, vol. 18, no. 2, Apr. 1993, pp. 108–122. 22, 24, 54

[22] G. J. Dobeck, "Image normalization using the serpentine forward-backward filter: application to high-resolution sonar imagery and its impact on mine detection and classification," in *SPIE Conf.*, vol. 5794, no. 381, 2005. 26, 27, 29, 87, 184

[23] S. Reed, R. I. Tena, C. Capus, and Y. Petillot, "The fusion of large scale classified side-scan sonar image mosaics." *IEEE trans. on image processing*, vol. 15, no. 7, pp. 2049–60, Jul. 2006. 29, 112, 113, 146

[24] S. Reed, "Automatic detection and classification models for sidescan sonar imagery," Ph.D. dissertation, 2004. 29, 33, 49, 52, 53, 57, 58, 59, 60, 61, 65, 66, 146, 148, 184, 185, 190

[25] P. Y. Mignotte, E. Coiras, Y. R. Petillot, and K. Lebart, "Adaptive fusion framework based on augmented reality training," *Journal on Radar, Sonar and Navigation*, vol. 2, no. 2, pp. 146–154, 2008. 29, 67

[26] D. Skinner and S. Y. Foo, "Classification using a radial basis function neural network on side-scan sonar data," in *SPIE Conf.* Ieee, Jun. 2007, pp. 1803–1806. 29

[27] Y. Petillot, S. Reed, and E. Coiras, "A framework for evaluating underwater mine detection and classification algorithms using augmented reality," in *Proceedings of the Undersea Defence Technology*, 2006. 29

[28] J. Tucker, "Coherence-based underwater target detection for side-scan sonar imagery," in *SPIE Conf.*, 2009. 29

[29] F. Crosby and J. Cobb, "Sonar processing for short range, very-high resolution autonomous underwater vehicle sensors," in *SPIE Conf.* Ieee, 2005, pp. 1–5. 29, 184

[30] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and . . .* , no. 2, pp. 100–107, 1968. 29

[31] E. Dura, J. Bell, and D. Lane, "Superellipse fitting for the recovery and classification of mine-like shapes in sidescan sonar images," *EEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 434–444, 2008. 33, 67, 71, 185

[32] J. Bell, Y. Petillot, S. Reed, E. Coiras, P. Y. Mignotte, and H. Rohou, "Target recognition in synthetic aperture and high resolution sidescan sonar," in *IET High Resolution Imaging and Target Classification*, 2006, pp. 99–106. 33, 57, 58, 65, 66, 67, 70, 85, 185

[33] M. An, J. Tory Cobb, B. Shenefelt, and R. Tolimieri, "Advances in group filter applications to sea mine detection," in *MTS/IEEE Oceans Conf.*, 2006, pp. 1–5. 33, 57, 58, 185

[34] R. Grasso and F. Spina, "Small bottom object density analysis from side scan sonar data by a mathematical morphology detector," in *Information Fusion Conf.*, 2006. 33, 57, 58, 64, 71, 185

[35] C. Rao, K. Mukherjee, S. Gupta, and A. Ray, "Underwater mine detection using symbolic pattern analysis of sidescan sonar images," in *MTS/IEEE Oceans Conf.*, 2009, pp. 5416–5421. 33, 57, 58, 64, 65, 71, 185

[36] S. Reed, Y. Petillot, and J. Bell, "An automatic approach to the detection and extraction of mine features in sidescan sonar," *IEEE Journal of Oceanic Engineering*, vol. 28, no. 1, pp. 90–105, Jan. 2003. 33, 57, 65, 66, 70, 85

[37] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy, "Three-class markovian segmentation of high-resolution sonar images," *Computer Vision and Image Understanding*, vol. 76, no. 3, pp. 191–204, Dec. 1999. 33, 47

[38] I. Leblond, M. Legris, and B. Solaiman, "Use of classification and segmentation of sidescan sonar images for long term registration," *MTS/IEEE Oceans Conf.*, 2005. 33

[39] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov random fields," *Computer Vision ECCV*, pp. 1–17, 2006. 34

[40] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine ...*, vol. 23, no. 11, pp. 1222–1239, 2001. 39, 48, 149, 150

[41] "GridCut." [Online]. Available: http://gridcut.com/ 39

[42] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124–37, Sep. 2004. 48

[43] H. Knutsson and C.-f. Westin, "Normalized and differential convolution methods for interpolation and filtering of incomplete and uncertain data," *Proceedings of Computer Vision and Pattern Recognition*, pp. 515–523, 1993. 49

[44] J. Nelson and N. Kingsbury, "Fractal dimension based sand ripple suppression for mine hunting with sidescan sonar," in *SPIE Conf.*, 2010, pp. 1–6. 55, 115, 118, 120, 132, 144, 166

[45] Q. Huynh, N. Neretti, N. Intrator, and G. Dobeck, "Image enhancement for pattern recognition," in *SPIE Conf.* Spie, 1998, pp. 306–314. 55

[46] N. V. Y. Petillot, Y. Pailhas, J. Sawas and J. Bell, "Target recognition in synthetic aperture sonar and high resolution side scan sonar using AUVs," in *International Conference on Synthetic Aperture Sonar and Synthetic Aperture Radar*, 2010. 56, 58, 65, 70, 71, 73, 74

[47] B. R. Calder, L. M. Linnett, and D. R. Carmichael, "Spatial stochastic models seabed object detection," in *SPIE Conf.*, 1993. 57, 58

[48] B. Calder, L. Linnet, and D. Carmichael, "A Bayesian approach to object detection in sidescan sonar," in *Image Processing and Its Applications Conf.*, vol. 2, Jul. 1997, pp. 857–861 vol.2. 57, 58, 60, 71

[49] R. Laterveer, D. Hughes, and S. Dugelay, "Markov random fields for target classification in low frequency sonar," in *MTS/IEEE Oceans Conf.*, 1998. 57, 58, 59

[50] I. Quidu, J. P. Malkasse, G. Burel, and P. Vilbe, "Mine classification using a hybrid set of descriptors," in *MTS/IEEE Oceans Conf.*, 2000, pp. 291–297. 57, 58, 67, 68, 185

[51] S. Reed, Y. Petillot, and J. Bell, "Automated approach to classification of mine-like objects in sidescan sonar using highlight and shadow information," *Radar, Sonar and Navigation, IEE Proceedings*, vol. 151, no. 1, pp. 48–56, Feb. 2004. 57

[52] J. Wilbur and S. G. Kargl, "Application of wavelets to acoustic resonance-elastic targets surounded by biologics," in *Proceedings of the ICASSP conference*, 1993. 57

[53] M. R. Azimi-Sadjadi, J. Wilbur, and G. J. Dobeck, "Isolation of resonances in acoustic backscatter from elastic targets using adaptive estimation schemes," *IEEE Journal Oceanic Eng.*, vol. 20, pp. 346–353, 1995. 57

[54] S. S. Abeysekera, P. S. Naidu, Y.-H. Leung, and H. Lew, "An underwater target classification scheme based on the acoustic backscatter form function," in *IEEE Proc. Acoustics, Speech and Signal Processing Conf.*, vol. 4, 1998, pp. 2513–2516. 57

[55] G. Goo and W. W. L. Au, "Detection and identification of buried objects in shallow water," in *Proceedings of SPIE, the International Society for Optical Engineering*, vol. 2765, 1996, pp. 201–214. 57

[56] G. Nicq and M. Brussieux, "A time frequency method for classifying objects at low frequencies," in *MTS/IEEE Oceans Conf.*, vol. 1, 1998, pp. 148–152. 57

[57] G. A. Carpenter and W. W. Streilein, "ARTMAP-FTR: a neural network for fusion target recognition with application to sonar classification," in *SPIE Conf.*, 1998, pp. 342–356. 57

[58] M. Granara, A. Pescetto, F. Repetto, G. Tacconi, and A. Trucco, "Statistical and neural techniques to buried object detection and classification," in *MTS/IEEE Oceans Conf.*, vol. 3, 1998, pp. 1269–1273. 57, 75

[59] Q. Huang, M. Azimi-Sadjadi, B. Tian, and G. Dobeck, "Underwater target classification using wavelet packets and neural networks," in *IEEE Trans. Neural Networks*, vol. 1, 1998, pp. 177–182. 57, 75

[60] M. R. Azimi-Sadjadi, D. Yao, A. A. Jamshidi, and G. J. Dobeck, "Underwater target classification in changing environments using an adaptive feature mapping," *IEEE Trans. Neural Networks*, vol. 13, no. 5, pp. 1099–1111, 2002. 57

[61] C. Yuan, M. R. Azimi-Sadjadi, J. Wilbur, and G. J. Dobeck, "Underwater target detection using multichannel subband adaptive filtering and high order correlation schemes," *Oceanic Engineering, IEEE Journal of*, vol. 25, pp. 192–205, 2000. 57

[62] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: a review of current status," *MTS/IEEE Oceans Conf.*, vol. 34, no. 3, pp. 207–224, Jul. 2009. 57, 188

[63] L. M. Linnett, D. R. Carmichael, S. J. Clarke, and A. D. Tress, "Texture analysis of sidescan sonar data," in *Texture analysis in radar and sonar, IEE Seminar on*, 1993, pp. 2/1–2/6. 58, 62

[64] L. M. Linnett, S. J. Clarke, C. Reid, and A. D. Tress, "Monitoring of the seabed using sidescan sonar and fractal processing," in *Proceedings of the Underwater Acoustics Group*, I. of Acoustics, Ed., vol. 15, 1993, pp. 49–64. 58, 62, 71

[65] L. Atallah, C. Shang, and R. Bates, "Object detection at different resolution in archaeological side-scan sonar images," in *MTS/IEEE Proc. Oceans Conf.*, vol. 1, 2005, pp. 287 – 292 Vol. 1. 58, 62, 65, 71

[66] J. D. Tucker, M. R. Azimi-Sadjadi, and G. J. Dobeck, "Coherent-based method for detection of underwater objects from sonar imagery," p. 65530U, 2007. 58, 63, 71

[67] T. G-Michael and J. Tucker, "Canonical correlation analysis for coherent change detection in synthetic aperture sonar imagery," in *Proceedings of the institue of acoustics*, 2010. 58, 63

[68] M. Lianantonakis and Y. R. Petillot, "Sidescan sonar segmentation using texture descriptors and active contours," in *MTS/IEEE Oceans Conf.*, vol. 32, no. 3, Jul. 2007, pp. 744–752. 60, 61, 148

[69] I. Karoui, R. Fablet, J.-M. Boucher, and J.-M. Augustin, "Seabed segmentation using optimized statistics of sonar textures," in *MTS/IEEE Oceans Conf.*, vol. 47, no. 6, Jun. 2009, pp. 1621–1631. 61, 112, 113, 117, 120, 148

[70] F. Maussang, J. Chanussot, A. Hetet, and M. Amate, "Mean-standard deviation representation of sonar images for echo detection: application to SAS images," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 4, pp. 956–970, 2007. 61, 86

[71] J. M. Bell, "A Model for the simulation of sidescan sonar," in *MTS/IEEE Oceans Conf.*, no. September, 1995. 62

[72] F. Maussang, J. Chanussot, A. Hetet, and M. Amate, "Higher-order statistics for the detection of small objects in a noisy background application on sonar imaging," *MTS/IEEE Oceans Conf.*, 2007. 63, 70, 71, 73, 74, 85

[73] S. Reed, Y. Petillot, and J. Bell, "Model-based approach to the detection and classification of mines in sidescan sonar," *Appl. Opt.*, vol. 43, no. 2, pp. 237–246, 2004. 60, 64, 71

[74] E. Hasanbelliu, J. Principe, and C. Slatton, "Correntropy based matched filtering for classification in sidescan sonar imagery," in *SPIE Conf.*, no. October. IEEE, Oct. 2009, pp. 2757–2762. 65, 66, 71

[75] J. Groen, E. Coiras, J. Del Rio Vera, and B. Evans, "Model-based sea mine classification with synthetic aperture sonar," *Radar, Sonar and Navigation, IET*, 2010. 65, 66

[76] E. Coiras, P. Mignotte, and Y. Petillot, "Supervised target detection and classification by training on augmented reality data," *Radar Sonar Navigation IET*, pp. 83–90, 2007. 67, 68, 71, 74, 75, 185

[77] N. Ma and C. S. Chia, "False alarm reduction by LS-SVM for manmade object detection from sidescan sonar omages," in *MTS/IEEE Oceans Conf.*, no. x. IEEE, Jun. 2007, pp. 1–6. 67

[78] V. Chandran, S. Elgar, and A. Nguyen, "Detection of mines in acoustic images using highorder spectral features," *IEEE Journal of Oceanic Engineering*, 2002. 67, 68, 71

[79] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004. 68

[80] J. Del Rio Vera, E. Coiras, J. Groen, and B. Evans, "Automatic target recognition in synthetic aperture sonar images based on geometrical feature extraction,"

EURASIP Journal on Advances in Signal Processing, vol. 2009, pp. 1–10, 2009. 69, 70, 71, 73, 85

[81] J. T. Cobb, K. C. Slatton, and G. J. Dobeck, "A parametric model for characterizing seabed textures in synthetic aperture sonar images," in *Proc. Spie*, vol. 35, no. 2, Apr. 2010, pp. 250–266. 73

[82] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pp. I–511–I–518, 2001. 74, 76

[83] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," *Computational learning theory*, vol. 139, pp. 119–139, 1995. 75

[84] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," *Pattern Recognition*, 2003. 76, 82

[85] C. Messom, "Fast and efficient rotated haar-like features using rotated integral images," *Australasian Conference on Robotics and*, 2006. 76, 87, 131

[86] "Open CV." [Online]. Available: http://opencv.org/ 76, 95

[87] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial . . .*, vol. 14, no. 5, pp. 771–780, 1999. 77

[88] S. W. Perry and L. Guan, "Detection of small man-made objects in multiple range sector scan imagery using neural networks," in *Proceedings of the CAD/CAC conference*, 2001. 86

[89] G. R. Cross and a. K. Jain, "Markov random field texture models." *IEEE transactions on pattern analysis and machine intelligence*, vol. 5, no. 1, pp. 25–39, Jan. 1983. 111

[90] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, 1979. 111

[91] B. Luyendyk, E. Hajic, and D. Simonett, "Side-scan sonar mapping and computer-aided interpretation in the Santa Barbara Channel, California," *Marine Geophysical Researches*, vol. 5, no. Figure 1, pp. 365–388, 1983. 112

[92] P. Blondel, J. C. Sempm, and A. A. Filtering, "Textural Analysis and Structure- ' hacking for geological mapping : Applications to sonar images from Endeavour Segment , Juan de Fuca Ridge," 1993. 112, 113, 161

[93] U. C. Herzfeld, "A method for seafloor classification using directional variograms, demonstrated for data from the western flank of the Mid-Atlantic Ridge," *Mathematical Geology*, vol. 25, no. 7, pp. 901–924, Oct. 1993. 112

[94] E. Pouliquen, B. Zerr, N. Pace, and F. Spina, "Seabed segmentation using a combination of high frequency sensors," *Oceans '99. MTS/IEEE. Riding the Crest into the 21st Century. Conference and Exhibition. Conference Proceedings (IEEE Cat. No.99CH37008)*, vol. 2, 1999. 112

[95] A. Bartholomä, "Acoustic bottom detection and seabed classification in the German Bight, southern North Sea," *Geo-Marine Letters*, vol. 26, no. 3, pp. 177–184, Jul. 2006. 112, 113

[96] J. Anderson, "Acoustic seabed classification: current practice and future directions," *ICES Journal of Marine Science*, pp. 1004–1011, 2008. 112, 113, 161

[97] P. Blondel and O. Gómez Sichi, "Textural analyses of multibeam sonar imagery from Stanton Banks, Northern Ireland continental shelf," in *MTS/IEEE Oceans Conf.*, vol. 70, no. 10. Elsevier Ltd, Oct. 2009, pp. 1288–1297. 112, 113, 161

[98] D. Williams, "Unsupervised seabed segmentation of synthetic aperture sonar imagery via wavelet features and spectral clustering," *IEEE (ICIP)*, pp. 557–560, 2009. 112, 113, 117, 161

[99] M. A. Cramer, J. E. Beach, T. A. Mazzuchi, and S. Sarkani, "Understanding information uncertainty within the context of a net-centric data model: a mine warfare example," *C2*, vol. 3, no. 1, pp. 1–42, 2009. 112

[100] D. Du and Y. Chen, "Investigation of the relationship between seafloor echo strength and sediment typea case study in Jiaozgou Bay, China," *Geo-Marine Letters*, vol. 27, no. 5, pp. 339–344, Mar. 2007. 113

[101] N. Redding, "Estimating the parameters of the k distribution in the intensity domain," *DSTO*, 1999. 113

[102] Z. Lubniewski and K. Bruniecki, "Seafloor characterisation and imaging using multibeam sonar data," *Acoustical Imaging*, vol. 31, pp. 437–445, 2012. 113

[103] L. Atallah, P. Smith, and C. Bates, "Wavelet analysis of bathymetric sidescan sonar data for the classification of seafloor sediments in Hopvågen Bay-Norway," *Marine Geophysical Researches*, pp. 431–442, 2002. 113

[104] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy, "Markov random field and fuzzy logic modeling in sonar imagery: application to the classification of underwater floor," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 4–24, Jul. 2000. 113, 146

[105] M. Dhibi, R. Courtis, A. Martin, and I. Quidu, "Multisegmentation of sonar images using belief function theory," in *MTS/IEEE Proc. Oceans Conf.*, vol. 123, no. 5, 2008, p. 3950. 113, 146

[106] I. Karoui, R. Fablet, J.-M. Boucher, and W. Pieczynski, "Fusion of textural statistics using a similarity measure: application to texture recognition and segmentation," in *MTS/IEEE Oceans Conf.*, vol. 11, no. 3-4, Mar. 2008, pp. 425–434. 113

[107] A. Martin and C. Osswald, "Experts fusion and multilayer perceptron based on belief learning for sonar image classification," in *MTS/IEEE Oceans Conf.*, 2008. 113, 146

[108] R. Haralick, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, 1973. 113

[109] D. Williams and E. Coiras, "On sand ripple detection in synthetic aperture sonar imagery," *IEEE International Conference on*, pp. 1074–1077, 2010. [Online]. Available: http://www.davidwilliamsphd.com/publications/DPW_ICASSP10_RD.pdf 115, 144, 166

[110] Y. Pailhas, Y. Petillot, C. Capus, and K. Brown, "Real-time sidescan simulator and applications," *Oceans 2009-Europe*, pp. 1–6, May 2009. 118

[111] J. Nelson and N. Kingsbury, "Dual-tree wavelets for estimation of locally varying and anisotropic fractal dimension," *IEEE International*, no. 0, pp. 1–4, 2010. [Online]. Available: http://www-sigproc.eng.cam.ac.uk/~jdbn2/icip2010.pdf 118

[112] Z. Liu and E. Sang, "Texture image classification using multi-fractal dimension," *Journal of Marine Science and Application*, vol. 2, no. 2, pp. 0–5, 2003. 118

[113] V. H. Diaz-Ramirez and V. Kober, "Target recognition under nonuniform illumination conditions." *Applied optics*, vol. 48, no. 7, pp. 1408–18, Mar. 2009. 118

[114] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: theory, design, and applications." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 15, no. 10, pp. 3089–101, Oct. 2006. 118

[115] N. Valeyrie, Y. Pailhas, C. Capus, and Y. Petillot, "Texture recognition in synthetic aperture sonar images with scattering operators," in *UAM*, 2011. 119, 120, 146, 166, 170

[116] J. Bruna, "Scattering representations for recognition," Ph.D. dissertation, 2012. 119, 120

[117] M. Mokji and S. A. Bakar, "Gray level co-occurrence matrix computation based on haar wavelet," *Computer Graphics, Imaging and Visualisation*, no. Cgiv, pp. 273–279, Aug. 2007. 120

[118] J. Fowler, "The redundant discrete wavelet transform and additive noise," *Signal Processing Letters, IEEE*, vol. 12, no. 9, pp. 629–632, 2005. 126

[119] I. Selesnick, "The dual-tree complex wavelet transform," *Sig. Processing Mag., IEEE*, no. November 2005, pp. 123–151, 2005. 128, 132

[120] O. Daniell, Y. Petillot, and S. Reed, "Unsupervised sea-floor classification for automatic target recognition," in *International Conf. Remote Sens.*, no. October, 2012. 131, 144, 166, 168, 186, 187

[121] E. Fakiris, D. Williams, M. Couillard, and W. Fox, "Sea-floor acoustic anisotropy and complexity assessment towards prediction of ATR performance," in *Proc. Int. Conf. Exhib. Underwater Acoust.*, 2013, pp. 1277–1284. 131, 144, 166

[122] B. Chakraborty, "Sea-floor classification using multibeam echo-sounding angular backscatter data: a real-time approach employing hybrid neural network architecture," *IEEE Journal of Oceanic Engineering*, vol. 28, no. 1, pp. 121–128, 2003. 146

[123] B. Chakraborty, V. Mahale, C. D. Sousa, and P. Das, "Seafloor Cclassification using echo-waveforms: a method employing yybrid neural network architecture," *IEEE Geoscience and Remote Sensing*, vol. 1, no. 3, pp. 196–200, 2004. 146

[124] H. Zhang, J. E. Fritts, and S. a. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 260–280, May 2008. 153

[125] C. Ratto, "Exploiting ground-penetrating radar phenomenology in a context-dependent framework for landmine detection and discrimination," *IEEE Geoscience and Remote Sensing*, pp. 1–12, 2011. 166

[126] C. Ratto, "Nonparametric bayesian context learning for buried threat detection," Ph.D. dissertation, 2012. 166

[127] C. Ratto, K. Morton, L. Collins, and P. Torrione, "Bayesian context-dependent learning for anomaly classification in hyperspectral imagery," *IEEE Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 1969–1981, 2014. 166

[128] L. Deng, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014. 189

[129] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. 191

[130] W. Dai, "Boosting for transfer learning," in *ICML*, 2007. 191