Unsupervised Maritime Target Detection

Asheer Bachoo

A thesis submitted to the Department of Electrical Engineering, University of Cape Town, in fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

University of Cape Town

June 2016

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Doctor of Philosophy in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author .....

Toronto
June 2016

Abstract

The unsupervised detection of maritime targets in grey scale video is a difficult problem in maritime video surveillance. Most approaches assume that the camera is static and employ pixel-wise background modelling techniques for foreground detection; other methods rely on colour or thermal information to detect targets. These methods fail in real-world situations when the static camera assumption is violated, and colour or thermal data is unavailable. In defence and security applications, prior information and training samples of targets may be unavailable for training a classifier; the learning of a one class classifier for the background may be impossible as well. Thus, an unsupervised online approach that attempts to learn from the scene data is highly desirable. In this thesis, the characteristics of the maritime scene and the ocean texture are exploited for foreground detection.

Two fast and effective methods are investigated for target detection. Firstly, online region-based background texture models are explored for describing the appearance of the ocean. This approach avoids the need for frame registration because the model is built spatially rather than temporally. The texture appearance of the ocean is described using Local Binary Pattern (LBP) descriptors. Two models are proposed: one model is a Gaussian Mixture (GMM) and the other, referred to as a Sparse Texture Model (STM), is a set of histogram texture distributions. The foreground detections are optimized using a Graph Cut (GC) that enforces spatial coherence. Secondly, feature tracking is investigated as a means of detecting stable features in an image frame that typically correspond to maritime targets; unstable features are background regions. This approach is a Track-Before-Detect (TBD) concept and it is implemented using a hierarchical scheme for motion estimation, and matching of Scale-Invariant Feature Transform (SIFT) appearance features.

The experimental results show that these approaches are feasible for foreground detection in maritime video when the camera is either static or moving. Receiver Operating Characteristic (ROC) curves were generated for five test sequences and the Area Under the ROC Curve (AUC) was analyzed for the performance of the proposed methods. The texture models, without GC optimization, achieved an AUC of 0.85 or greater on four out of the five test videos. At 50% True Positive Rate (TPR), these four test scenarios had a False Positive Rate (FPR) of less

than 2%. With the GC optimization, an AUC of greater than 0.8 was achieved for all the test cases and the FPR was reduced in all cases when compared to the results without the GC. In comparison to the state of the art in background modelling for maritime scenes, our texture model methods achieved the best performance or comparable performance. The two texture models executed at a reasonable processing frame rate. The experimental results for TBD show that one may detect target features using a simple track score based on the track length. At 50% TPR a FPR of less than 4% is achieved for four out of the five test scenarios. These results are very promising for maritime target detection.

To my wife and Aaran,
for all their patience.

Acknowledgements

The following acknowledgements are made:

- The Council for Scientific and Industrial Research (CSIR) and National Research Foundation (NRF) for funding some aspects of this work.
- Rob Calitz, formerly of Armaments Corporation of South Africa (ARMSCOR), for always motivating me to solve problems.
- Derek Griffith for providing the information and requirements for a long range optical surveillance system.
- My former colleagues at the CSIR, and my current ones at Tsotsos Lab, for all their ideas and advice.
- My wife for all her patience and support.
- John Tsotsos for motivating me to complete this thesis.
- And my supervisor Fred Nicolls for his great advice and patience.

Contents

List of Figures

List of Tables

Acronyms

2D 2-Dimensional
3D 3-Dimensional
AIVS3 Automated Intelligent Video Surveillance System for Ships
ARGOS Automatic Remote Grand Canal Observation System
ARMSCOR Armaments Corporation of South Africa
ASV Automatic Sea Vision
AUC Area Under the ROC Curve
CMO Close-Minus-Open
CPU Central Processing Unit
CSIR Council for Scientific and Industrial Research
CUDA Compute Unified Device Architecture
CUHK City University of Hong Kong
DTM1 Dynamic Texture Mixture with 1 component
DTM3 Dynamic Texture Mixture with 3 components
EM Expectation-Maximization
FNR False Negative Rate
FPR False Positive Rate
FFT Fast Fourier Transform
GC Graph Cut
GMM Gaussian Mixture Model
GPS Global Positioning System
GPU Graphics Processing Unit
HMM Hidden Markov Model
HOG Histogram of Oriented Gradients
HSV Hue-Saturation-Value
IMU Inertial Measurement Unit
IR Infrared
KDE Kernel Density Estimator
KL Kullback-Leibler

KLT Kanade-Lucas-Tomasi
LBP Local Binary Pattern
MAAW Maritime Activity Analysis Workbench
MAP Maximum a Posteriori
MHI Motion History Image
MHT Multiple Hypothesis Tracker
MRF Markov Random Field
NIR Near Infrared
PCA Principal Components Analysis
PTZ Pan-Tilt-Zoom
RANSAC Random Sample Consensus
RGB Red-Green-Blue
ROC Receiver Operating Characteristic
RHIB Rigid Hull Inflatable Boat
SG Stauffer-Grimson
SIFT Scale-Invariant Feature Transform
SLIC Simple Linear Iterative Clustering
SNR Signal-to-Noise Ratio
STM Sparse Texture Model
SVD Singular Value Decomposition
SVM Support Vector Machine
TBD Track-Before-Detect
TPR True Positive Rate
UAV Unmanned Aerial Vehicle
VAM Visual Attention Map

Chapter 1

Introduction

The work contained in this thesis addresses the area of unsupervised maritime target detection in video captured by static or moving cameras. Target detection is a relevant function for optical maritime surveillance systems supporting border control and anti-piracy around the world. It is the first part of more sophisticated operations such as object tracking and target classification. Most surveillance systems in the defence sector combine a number of different technologies to achieve improved situational awareness. Thus, optical target detection should be viewed as a complementary technology within a complex surveillance system containing sensors such as radar and lasers.

1.1 Problem Statement

The problem that is addressed in this work is the unsupervised detection of maritime targets in grey scale video. For our particular experiments, two optical configurations are evaluated:

- A long range surveillance system with a 300mm lens that is moving; the sensor motion arises from the pan-tilt unit that the camera is mounted on.
- A medium/wide field of view camera system for situational awareness.

The characteristics of the optical system are driven by the need to acquire visuals of targets that are far away from a wide/medium angle optical sensor. These two fields of view — wide and narrow — form the basic components of an optical surveillance system for situational awareness. The pan-tilt unit enables the following of a target once it has been detected and is used for high-precision pointing.

The problem that we aim to solve is novel because there is limited work in unsupervised maritime detection from a moving camera using a monochrome sensor. The methods are also applicable to static cameras. Existing research in this area proposes supervised detectors and the use of colour and/or infrared sensors for object and background separation. We address the problem in an unsupervised fashion by attempting to model directly the characteristics of the scene data.

1.2 Thesis Statement

This thesis investigates unsupervised methods for detecting maritime targets in grey scale video. Two methods are investigated:

- We explore region-based background texture models to describe the appearance of the ocean. Outliers of this model correspond to potential maritime targets. These models are also used in a graph cut method to improve target detection. The models are built spatially rather than temporally.
- Feature tracking is investigated as a means of detecting stable parts of an image frame that typically correspond to potential maritime targets; unstable features are background regions.

We show through empirical analysis on a set of data that these approaches are feasible for solving our problem. This research is only concerned with ways to describe the characteristics of the ocean appearance — target tracking is not the focus. We phrase the thesis statement as follows:

Robust unsupervised detection of a maritime target from a static or moving camera can be achieved using a region-based texture model, within a graph-cut framework, and feature tracking.

1.3 Objectives

The main objectives of this work are:

- To present unsupervised methods for detecting maritime targets in grey scale video captured by a static or moving camera.
- To present methods that are fast and efficient for real-time applications.
- To provide an empirical evaluation of the proposed methods on real-world data.

- To provide a comparison of the proposed methods to existing techniques.

1.4 Limitations and Assumptions

The envisaged application for the research contained herein is single target tracking in the maritime domain. Although some of our test sequences contain up to two targets, we are primarily concerned with a high precision pointing system that maintains a single target within its field of view for purposes of tracking the object in a closed loop system with a pan and tilt unit. Thus, we do not consider scenes with several targets. It is also assumed that targets are of reasonable size for detection. The methods presented here are concerned with finding objects on the ocean and, therefore, we are only concerned with objects on sea backgrounds; targets in littoral areas are not part of this study. It is assumed that an input video stream contains only ocean background, or ocean background with sky.

1.5 Methodology

The research method used to explore the thesis statement is empirical research. An empirical study is conducted with a design for a maritime target detection system. Computer vision methods are proposed to detect maritime targets. The methods are unsupervised in that no prior training patterns and labels are specified for learning these models. Targets are inferred from the video frames using a few heuristics for maritime video data. Experiments are carried out to test the feasibility of the proposed solution using empirical evidence. The experimental process evaluates the performance of a method for detecting maritime targets; we present both qualitative and quantitative analysis and also compare some of the results to existing methods to show the effectiveness of the proposed solutions.

1.6 Contributions

The main contributions of this work are:

- Maritime object detection from static or moving cameras.
- Online region-based texture models, using Local Binary Pattern (LBP) feature descriptors, for describing the ocean texture appearance.
- A Graph Cut (GC) formulation for improving the image labelling during foreground detection.

- Evaluation of feature tracking for a Track-Before-Detect (TBD) approach.
- Empirical analysis of the methods on real-world data.

1.7 Thesis Development

1.7.1 Background and Literature Study

Chapter 2 presents background details on the application area that is addressed in this work, as well as a literature survey on the state of the art in maritime target detection and tracking. Related work in background modelling, and feature detection and tracking is also discussed. The initial ideas that developed into this thesis emerged from the requirements of the South African Navy for an optical tracker. The optical tracker complements a wide area surveillance system by providing a narrow field of view, and tracking capabilities, for objects far away. This chapter also discusses the significance of our work by considering how it addresses some of the limitations in the prior art.

1.7.2 Maritime Scene Background Modelling

Two online region-based background models for describing the ocean's appearance are proposed in Chapter 3: the well-known Gaussian Mixture Models (GMMs) and the Sparse Texture Model (STM). The STM is composed of a number of histogram feature distributions. The models are formulated as one class classifiers consisting of K texture components: outliers of these models are potential targets of interest. The particular characteristics of these models are discussed in detail, including an online Expectation-Maximization (EM) approximation for the GMM update and a simple low-pass filter update strategy for the STM. The K components of the background models are initialized with data clustering. This is achieved with the k-means++ technique for careful seeding of the clusters, and an information theoretic data clustering algorithm that is suitable for histogram data. The textural characteristics of the ocean are described with histograms of LBPs.

1.7.3 Background Modelling Experiments

Chapter 4 presents experimental results on the proposed texture models. Five video sequences taken from the Council for Scientific and Industrial Research (CSIR) and the City University of Hong Kong (CUHK), and containing a total of 2183 frames are used in the experiments. Five sets of experiments are discussed to show the advantages of the texture models:

- **Model initialization:** The efficacy of the data clustering algorithm for the model initialization is shown. An emergent result is the suitability of the LBP features for describing the textural characteristics of a scene.
- **Texture saliency:** The texture models are shown to provide a global saliency measure such that targets are more salient than ocean regions.
- **Horizon detection:** The LBP descriptor is used to calculate a change in texture gradient. This is used to find the horizon line.
- **Foreground detection:** The performance of the GMM and STM for background modelling and target detection is analyzed using all five tests sequences. Performance metrics are derived from the Receiver Operating Characteristic (ROC) curves.
- **Comparison to existing methods:** We compare the results of foreground detection to existing methods for background modelling. The results presented by Chan et al. are used as a benchmark [1].

The GMM and STM offer similar levels of performance, although the STM is computationally faster. For most of the test sequences the models produce a False Positive Rate (FPR) of less than 2% at 50% True Positive Rate (TPR). Our methods are comparable to the state of the art, and in some cases they are better than the state of the art for foreground detection in maritime scenes.

1.7.4 Improved Foreground Labelling using Graph Cut

Variations in image data cause poor spatial coherence in foreground detection. The sources of these variations include image appearance changes, sensor noise and environmental effects. Pixels that exhibit these types of variations deviate from the general appearance that our texture models describe. Given the noisy observed data, one wishes to infer the class label of a pixel where the label denotes membership to either target or background. This chapter discusses GC optimization to improve foreground detection — an energy function with a data term and a smoothness term is minimized. The texture models are used to compute the data term; pairwise affinities are specified for the smoothness energy term. Our results show that the GC optimization is able to improve the FPR for all our test cases; in some situations the TPR is improved when holes in a target labelling are filled.

1.7.5 Feature Tracking for Track-Before-Detect in Maritime Scenes

In Chapter 6, we deviate from region-based texture models and explore feature tracking for unsupervised maritime target detection. In a maritime scene containing targets, the targets

exhibit locally stable appearances that one may exploit for detecting objects that persist over a number of video frames. Under most scenarios it is a fair assumption that salient features of the ocean texture will persist for much shorter intervals, due to the dynamic ocean appearance, compared to features on maritime targets. The idea that is described above falls within the concept of TBD, a well-known approach in the radar literature [2]. TBD is designed to work in a cluttered environment with the core idea being the accumulation of evidence over time for an observation until it is known that the observation originates from background or target. This is beneficial for maritime tracking in a cluttered environment with dynamic background appearances.

In the case of a maritime scene, one may monitor the lifespan of a track with the view that rigid objects will have longer lifespans than ocean features. It is felt that the TBD definition of our approach is appropriate because a number of other cues, such as appearance or motion, may be used to accumulate evidence of a target; track length is just one simple example related to visual appearance persistence. We propose using the length of a track as the confidence score for a track. The proposed method is able to label detected features as either background or foreground. This labelling is useful for constructing appearance models or for providing additional cues in a computer vision model. The feature tracking implementation executes at 28 frames-per-second on 1360×1024 video for approximately 200 keypoints, using a coarse to fine block matching strategy on Harris corners and SIFT descriptors for the feature appearances. The software executes on a Graphics Processing Unit (GPU) using the Compute Unified Device Architecture (CUDA) framework. A FPR of less than 4% at 50% TPR is achieved for four out of the five test sequences.

Chapter 2

Background and Literature Study

This chapter presents an overview of the literature on detection and tracking of maritime targets from video. It also presents related work in background modelling, and feature detection and tracking. The field of maritime target detection and tracking from video is generally confidential or top secret as it falls within the military sector. Thus, the literature study is limited to works available to the public.

2.1 Background

The initial work for this thesis was done for the South African Navy through the ARMSCOR; the research was aimed at replacing the current optical trackers on the South African frigates. The optical trackers complement a wide area surveillance system by providing a narrow field of view, and tracking capabilities, for objects far away. Long range surveillance using optics is influenced by several key factors[1]:

- Sensors in the visual band are necessary if one wishes to identify the target. Moreover, thermal sensors do not offer high optical resolution and are much more expensive than visual cameras.

[1]These notes were provided courtesy of Derek Griffith of the CSIR

- In long-range surveillance through the atmosphere, contrast and colour saturation are both strongly influenced by atmospheric scattering. The atmospheric scattering becomes stronger with the inverse fourth power of the wavelength. Thus, blue-visible wavelengths are much more strongly affected and there is good reason to move into the Near Infrared (NIR) band for better atmospheric transmission and contrast.

- Targets of military and security importance usually go to considerable trouble to suppress their colour signature.

- Colour images of poor saturation and contrast give the visual impression of "poor" image quality. This potentially has a psychological as well as a real impact on the performance of the human in the loop.

- Most colour cameras still rely on the Bayer filter principle. This has multiple disadvantages. Firstly, by nature these colour filters cut out all photons besides those in the colour bands. That is, there is a large loss of potential photonic signal and a consequent reduction in signal-to-noise ratio. This is very bad under low light and night conditions. Secondly, the Bayer filter pattern effectively halves the true available spatial resolution of the sensor. Lastly, because the Bayer filter samples the three primary colours at different photosites, there is potential for colour moire and aliasing effects which can be very distracting to the observer. The answer to Bayer filter disadvantages is the 3-CCD camera — one CCD sensor for each colour band with a colour splitter prism block. These cameras are more than three times the cost and produce three times the amount of image data of a single sensor camera.

In light of these factors, monochrome sensors with good NIR response were preferred for near and far surveillance in the maritime applications.

The optical characteristics of the tracker are one aspect; the second aspect is the image processing and computer vision methods for target detection and tracking. Several products were considered for the image processing components. However, specialized solutions did not exist. It was found that a number of companies provide software and hardware solutions for object tracking but none of these solutions were particularly suitable for maritime environments. While they afford a vast array of settings and options for tracking multiple objects, they were far too generic and did not model the particular characteristics of maritime video data to address the problem. Furthermore, it is difficult to obtain performance figures for these systems. Thus, this work focuses on a specialized solution for maritime image processing.

2.2 Literature Review and Related Work

This section discusses prior work in maritime target detection and tracking. The prior work is organized into two categories: systems with static cameras, and systems with moving cameras. This literature review is not exhaustive — a recent survey paper has additional material with more details [3]. In addition to the maritime field, we also present related work in background modelling, and feature detection and tracking.

2.2.1 Maritime Target Detection and Tracking with Static Cameras

Sanderson, Teal and Ellis used Fourier space characterization to identify small craft in a maritime scene [5]. A frequency template of the current sea state was constructed using a set of windows in a video frame with the current sea state — it is not explained how this video frame is selected. The Fast Fourier Transform (FFT) was calculated for each selected window and the template attributed to the sea state was simply the mean of the FFT transforms. These reference frequencies were suppressed in an input image and target hypotheses were generated from the output of this filtering step using frame differencing (a static camera was assumed). The system used a set of *a priori* motion constraints to detect potential maritime targets through association of the hypotheses with targets in the previous frames. This paper presented no quantitative results for the detection or tracking of the targets although the authors claimed it had been tested on a sequence of several small maritime vessels. The monochrome test data was acquired by a static camera. A resulting detection and track was presented for a single frame by the authors.

The work of Smith and Teal is similar to that of Sanderson et al. as described above [6]. A static video camera captured NIR data that was filtered through the analysis of pixel variances. Once this was done a histogram construction technique was executed with a user-defined starting point; the result was the characteristic grey level histogram of the sea. This simple model of the sea appearance was then used to accept or reject potential target image windows using statistical comparisons (no actual details were provided for this comparison). The candidate target windows were processed to generate motion cues and these were matched and tracked using objects from previous frames (as in Sanderson et al. [5]). The experimental results were also qualitative rather than quantitative, and did not present any performance metrics for the method. Some results were shown for a single test sequence.

Voles et al. performed moving target detection in maritime scenes by computing statistical differences between image frames [7]. A video frame was filtered using anisotropic filtering and split up into non-overlapping tiles. Statistical features such as energy, entropy, homogeneity and contrast were computed for each tile and compared to the corresponding tiles in the previous frame to detect a change. A motion map was also computed through frame differencing using

the original input image. The motion map and change detection image were subsequently combined to detect moving regions. Qualitative experimental results were presented for data captured by a static camera; no actual performance metrics were provided by the authors although they state the use of several sequences for testing.

Voles and Teal continued their work and presented a method for nautical scene segmentation using variable size image windows and reclustering of features [8]. The centroid of a group of characteristic vectors, extracted from variable size windows, was found through iterative clustering. This centroid was the primary prototype that describes the sea state. The Mahalanobis distance from the prototype was calculated for each feature vector and it was used to detect the presence of targets. The feature vector for each image window consisted of statistical measures for energy, entropy, homogeneity and contrast. This approach assumed that the sea state is unimodal in the multidimensional feature space. Data for the experiments was captured using a static camera; although performance metrics are presented for two lengthy sequences, there is no explanation of how these metrics were computed (e.g. using bounding box overlap). The authors reported 91% correct segmentation of the maritime vessels in a test sequence; in another sequence 95% of the yachts were detected.

A hybrid color-based foreground object detection method using a static camera for automated marine surveillance was proposed by Socek et al. [10]. The primary method combined a probabilistic model of the background with a low pass filtered reference image and Bayesian classification for change detection. The main assumption relied on the idea that background pixels can be described by particular features with significant probability — they can be moving but are stationary in general. Appearance information was described by colour and appearance change by colour co-occurrence. The foreground segmentation method detected a change, classified it and performed foreground segmentation post-processing and Bayesian background model learning and updating using information about classified pixels. The post-processing was chosen over the more common morphological operations and it used graph partitioning — edge weights were set by using the difference between neighbouring colour features. Similar to some of the previous work that has been discussed, the authors claim that the method has been tested on several maritime sequences. However, the results that are presented are minor and there are no performance figures.

The work of Bouma et al. focused on automatic detection of small surface targets using thermal data [17]. Their methods used multi-scale edge detection to locate the horizon for stabilization and determination of the ocean region. Background estimation, with a robust estimator, was used to detect targets below the horizon line. The background subtraction process also made use of the horizon line to take orientation into account. This was used to prevent artifacts arising near the horizon. Objects were detected using hysteresis thresholding based on the difference between a background image and the current image. A horizon detection accuracy of 80% was claimed; an object detection accuracy of 80%, and 3% false detections per frame,

was reported for four classes of 29 sequences.

Sullivan and Shah used a maximum average correlation height (MACH) filter to combine multiple training images into a single template [18]. The template was correlated with an input image — the highest peak corresponded to the most likely vessel location. A peak was classified as a vessel if it was greater than a threshold. Templates were created for each vessel type in the test data. They addressed change in appearance size by doing the correlation with a set of scaled templates and choosing the scale with the highest peak. When a vessel was detected, a Kalman filter tracker was initialized for the object. The algorithm was used to provide early warning for unauthorized port access. The authors reported a a mean recognition accuracy of 88% and a mean false alarm rate of approximately 5 % for port access. Unfortunately, they did not present any detection or tracking performance metrics. It is assumed that the cameras were static as they were in fixed positions for monitoring port access.

The Automatic Remote Grand Canal Observation System (ARGOS) was presented by Bloisi and Iocchi for real-time boat traffic monitoring in the Grand Canal of Venice in Italy [21]. The system tracked multiple targets through the day and night and made use of calibrated cameras to transform image coordinates to world-referenced coordinates. The main function of the system was optical detection and tracking of moving targets and detection of predefined events such as speed limits, parallel travel, wrong direction and forbidden stops (with transmission of data to a control center). Target detection was achieved using an online mixture of Gaussians for background subtraction in colour space; optical flow was employed on foreground segments for clustering detections and for eliminating noise. The clustering algorithm returned a bounding ellipse for each detected object and they were tracked with a Multiple Hypothesis Tracker (MHT) using Kalman filters and nearest neighbour data association. The system was tested on prerecorded video, for velocity and position estimation, using ground truth created from a Global Positioning System (GPS) unit on a boat. A qualitative assessment, by visual inspection, was done on live video for detection, tracking and object count. For both cases the data was captured by a static colour camera. The authors noted that most errors in the system were caused by wakes in the water or poor colour separation between background and foreground. They also mentioned that the data association was error prone when counting a large number of boats. The average False Negative Rate (FNR) over several sequences for detection and tracking was 0.028.

The work of Gupta et al. presented another system, Maritime Activity Analysis Workbench (MAAW), similar to ARGOS [24]. The system was geared towards anti-piracy and anti-terrorism and focussed mainly on small vessels in littoral areas such as bays, harbours and rivers. It used several different sensors with varying resolutions and some of the video data were also reported to contain compression artifacts. As the cameras were static, the image background was modelled using a single online Gaussian function where pixels were weighted to adjust to changes. A threshold was used to detect foreground pixels. The object

tracker clustered pixels into segments (using simulated annealing) and associated the segments to tracks. MAAW was also capable of interpreting the behaviour of maritime targets using supervised learning — a case-based method was used for object and activity classification, such as "cruising" or "sightseeing". Their paper presented preliminary results for object and activity classification rather than tracking and detection.

The work of Wijnhoven et al. considered online learning for ship detection [27]. A fast online learning approach using stochastic gradient descent on Histogram of Oriented Gradients (HOG) features for linear classifiers was employed, with a regularization constraint similar to SVMs. The online classifier processed just a single training sample at each iteration in a stream-like fashion. The training images were scaled to detect objects of different sizes. The authors presented experimental results for offline and online training. Their results showed that offline training has 0.68 area under the precision-recall curve — 85% of ships were detected but with many false positives. The online training showed that with a limited training set a useful detector can be created. However, when adding more samples for online training detection does not always improve (possibly as a result of over-training). They also showed that when learning 70-90% of the training data, the online detector outperforms the offline detector. However, the online detector also processes more negative samples and this could be the deciding factor in its good performance. For each classifier, the detection threshold was determined empirically. Test video was recorded over three days with varying weather conditions. A total of 150 different ships were recorded — 50 ships were used as training data and the other 100 were used to evaluate the classifier performance. A sliding window detector was used during testing. The test data was images rather than video.

Szpak and Tapamo used background subtraction to detect foreground maritime targets [28]. A single Gaussian modelled a pixel's grey-scale intensity; a small set of their experiments showed that for most maritime scenes in their data, the unimodal data assumption was valid. Their background model also incorporated a delay parameter that specified when the model would be updated. This was done to prevent the model from learning foreground appearances. Objects were detected at the pixel level if their intensity was outside a range, determined by the models' variance, from the mean. The binary image of detections was smoothed by a heuristic filter and then segmented with a level-set function that minimized the Chan-Vese energy. The level set minimization was done at each frame, using results from the previous frame, to track objects. The authors reported promising results, quantitatively and qualitatively, on 30 video sequences. They reported that target detection was impeded when the scene had low contrast. Most of the false positives in their system was caused by glint on the sea. Frost and Tapamo extended this work to include shape priors as well as additional energy functionals for tracking, such as contrast and homogeneity descriptors [29].

A method for robust moving ship detection using water region detection was proposed by Boa et al. [30, 31]. The method used a pre-trained Support Vector Machine (SVM) to classify water

and non-water segments. The SVM is trained using colour, Gabor, and Laws' texture features. First, graph based segmentation was performed and water segments were labelled using the SVM classifier. The non-water segments were then grouped into regions using motion similarity and context information e.g. non-water regions are surrounded by water. A user-defined parameter specified the level of segment merging. Motion saliency was thereafter computed, using motion contrast between ship and surrounding local background, and a threshold was applied to the saliency value to detect moving ships. The authors reported precision of 96.4% and recall of 97.6% for water detection; the ship motion detection reported precision greater than 75% and recall greater than 79% for four different video sequences. It would appear that the video sequences were captured by a static camera because no camera motion compensation process was reported.

Colour background modeling using a multivariate Gaussian, per pixel, with full covariance was implemented by Kaimakis and Tsapatsoulis [32]. The model parameters were adapted online using class ownership information. During the learning process, the covariance matrices were also adjusted slightly to prevent singular matrices from occurring. The likelihood images formed from the Gaussian background model were convolved with a 2-Dimensional (2D) Gaussian filter to ensure target compactness. Targets were detected by thresholding the likelihood image, performing morphological filtering and then locating the foreground clusters. Clusters with a low colour variance were pruned as they were assumed to be ocean class. The algorithms were implemented for a Pan-Tilt-Zoom (PTZ) unit; the methods only executed when the camera was static after the unit moved. It was reported that the majority of false positives occurred on windy days and false negatives were mainly caused by occlusions. Overall, the system reported low false positives and false negatives on over 50 video sequences.

2.2.2 Maritime Target Detection and Tracking with Moving Cameras

Lee, Huang and Chen used Infrared (IR) video data for automatic ship detection and tracking [4]. In their work it was assumed that targets emit more energy than their surroundings and, thus, targets in thermal imagery will have the brightest pixels. Static thresholding was proposed to detect candidate target regions in a video frame. The image processing steps consisted of brightness correction, computation of image statistics for thresholding and some post processing using morphological operators and component labelling for detecting candidates. Once candidate targets had been detected, they were associated to tracks using a gate, and the distance between a track and the object's centroid. A track confidence score was then computed, using a contrast-difference correlation algorithm, on the candidate region and track regions. Objects that persisted during tracking, over a number of frames, were flagged as true targets whereas unstable tracks were of a spurious nature and were discarded. A threshold was defined for the correlation measure and it set the tracker to a tracking or coasting mode. This track confidence was calculated by counting the number of pixels within a window that

were close to the pixel intensities in the tracked region of the previous frame. The authors do not state if the camera was static or moving. However, it can be inferred that the method is applicable to a camera with small amounts of motion. They presented qualitative experimental results, showing detection and tracking, for three image sequences with a low sampling rate of one frame grabbed every four seconds from a video-taped sequence of a slow moving ship.

Voles continued his maritime research in his PhD thesis [9]. The segmentation method described in the previous section was combined with a weak perspective model to obtain object regions. Corner features were extracted for each object region and they were used for motion estimation across the video frames. Motion compensation was performed by tracking the horizon for global displacements. A Kalman tracker was used for tracking object position and velocity. The system was tested on two videos captured by a moving platform and a low number of false negatives was reported. False negatives arose from small targets with low contrast and the false positives mainly resulted from wakes of moving vessels.

The mean shift method was used by Bibby and Reid for visual tracking at sea [11]. Unlike the previous methods, this work was devoted to single target tracking using a stabilized pan-tilt-roll unit. It did not feature automatic target detection; the user had to initialize the target manually before the tracker started. With regard to the mean shift method, histograms of Red-Green-Blue (RGB) colour and image gradient magnitude were used for the target model. This was combined with suppression of the background histogram to localize the target. A small set of experimental results was presented for the method without any performance metrics.

Bibby and Reid extended their work to use a bag of pixels representation for appearances [12]. The complete method made use of image registration between frames, level set segmentation of the target and online learning of the colour appearance model. A warp function for frame registration was embedded in the level set formulation to achieve object tracking. The model update was performed once the registration and segmentation were completed, using simple linear learning. The models and tracking were initialized using a detection model or manually by a user. Quantitative results for the object tracking were limited in this paper as no performance metrics (e.g. tracking error) were provided. However, a quantitative analysis of their model's cost function was compared to other popular ones using 20 000 frames of video data that showed the merits of their approach.

Fefilatyev [13], and Fefilatyev and Goldgof [14], proposed an algorithm with a combination of colour-based horizon detection, edge detection, post processing and component labelling to find marine objects in single images and videos. The image and video data was captured by buoy mounted cameras for far lying objects. Thus, in typical videos ships and boats appeared to rest just above the horizon line due to the line between camera and marine vessel being parallel to the sea level. Edges above the horizon line were considered to belong to targets and form a convex hull around the object. The Canny edge detector was used with settings suited

to objects with long edges such as yachts and barges. A crucial component of the method was the horizon detection — the object detection is dependent on correct horizon detection. Once edges above the horizon line were found, erosion and dilation were performed for post processing. Components were then extracted using connected components labelling; connected regions were assumed to belong to one object. In video sequences, this method was combined with the Kalman filter for tracking multiple targets — a track manager initiated and deleted tracks. Performance figures were presented for horizon detection and for target detection using bounding box overlap between the ground truth and the algorithm's output. In follow up work, image registration was incorporated to improve the tracking [15]. The registration step contributed to maintaining global reference coordinates. The full system was described in more detail, with some minor additions and more experimental results, in a recent paper [16]. The extensive experimental results in this paper, based on 50 000 evaluation frames, showed that their system was quite robust. They reported the usual problems that are experienced with image areas that have no texture or a lack of salient features.

A method for aerial search of humans was proposed by Westall et al. [19]. The core component of their proposal used point target detection with *a priori* information followed by temporal tracking. Four point detection methods and two tracking methods were evaluated. The authors explored colour spaces for maximizing background and foreground separation. This was a crucial component of their technique as it consisted of a human hair colour model for foreground probability distributions and an ocean colour model as well. The proposed method searches for human heads that have a size of 1-3 pixels in aerial video. Four filtering methods — using morphological filtering (preserved sign method of close-minus-open) and basic median filtering — were used to discover potential point targets by taking the difference of the original image and the filtered image. The system then integrated detections over time to track low Signal-to-Noise Ratio (SNR) targets in high clutter using the colour models. Dynamic programming and Hidden Markov Models (HMMs) were compared for tracking targets. The data for the experiments consisted of 10 simulated videos and one real video that were post processed to compensate for motion. Several performance metrics were presented, such as false alarm rate, missed detection rate, first frame of true target detection and false alarm track length. The authors concluded that their methods were not suitable for real-world applications due to the high FPR. In related work, the Close-Minus-Open (CMO) filtering technique was used with a HMM and fused colour spaces [20]. The colour fusion showed improvements using the simulated data but had mixed results for real world data.

In follow up work from ARGOS, Bloisi et al. used Haar-like features and supervised learning to detect maritime targets using a PTZ unit [22]. The detections were integrated by a tracker to get stable tracks with at least 10 observations. The data association between tracks and observations was done using the Bhattacharyya distance between Hue-Saturation-Value (HSV) histograms. The maritime object classifier was trained offline using 4000 negative and 1500 positive examples; a second classifier was built for false positives such as wakes and reflections.

The authors reported a detection rate of 0.92 and a false alarm rate of 0.25.

The Automated Intelligent Video Surveillance System for Ships (AIVS3) was a system for maritime target detection, tracking, and classification as well as interpretation of scene activities and issuing of alerts using moving sensors [23]. This system performed horizon detection using the Hough transform to find the water region. As the video frames were of a low resolution, scene clutter was already filtered out. The water pixel intensities were modelled as a regression on pixel coordinates; non-water pixels had a high residual in the model and were marked as candidate targets. A perspective compensating morphological operator was thereafter used to remove noise in the image containing candidate targets. A track/suspend/match paradigm was used for tracking — an object is tracked using a Kalman filter until its track confidence drops; when this happens, tracking is suspended until new observations are matched to the track. The system also incorporated a decision forest — using cues such as Gabor features, colour histograms and edge histograms — for classifying watercraft, and a finite state system that mapped target motion attributes to a symbol vocabulary for threat levels. The qualitative experimental results appear promising.

The system of Krüger and Orlov used layer-based detection and tracking of small boats in thermal video using a pan and tilt unit [25]. In their work, layers refer to the difference pieces of hardware and software for detecting and tracking targets. Camera orientation was estimated using an Inertial Measurement Unit (IMU). The camera orientation was used to improve the edge-based localization of the horizon line. This system was designed to detect distant boats and, thus, those targets appeared near the horizon line. The target search areas were fixed relative to the horizon. Temporally stable features (such as maximally stable extremal regions and Shi-Tomasi corners) were detected by the system and they were fused and used to compute alarms or for tracking targets. Some qualitative experimental results were presented for a few test cases showing horizon detection and tracking.

An IR system for real-time 24-hour maritime safety and security was Automatic Sea Vision (ASV) [26]. Various sensors such as cameras and GPS units were incorporated with a user interface. The subsystems were capable of data enhancement, target detection, target tracking and the publishing of track information. The system detects the water surface by finding the horizon line and coast. The sea was characterized by the mean and standard deviation of the image intensities. Outliers of this distribution were classified as object pixels and grouped into object regions. The paper provided insufficient details on the actual method implemented for detection. The image coordinates of objects were converted to world coordinates for tracking. The object attributes calculated were shape, azimuth and radial distance, and apparent height and width; tracks are built from the detections and data was associated to tracks using position, size and velocities. The track objects were maintained or eliminated using an association cost based on these aforementioned measures. The paper is not clear on whether tracking filters, such as a Kalman filter, are used. Several results are presented and a false alarm rate of less

than 1% and detection rates over 80% are reported. Most of the false alarms in the system were caused by wakes.

Makantasis et al. combined a mixture of Gaussians background subtraction model with a Visual Attention Map (VAM) of the ocean [33]. Low level features, such as edges and colour, were computed for 8×8 blocks in the input image. Global block uniqueness was established by computing the sum of absolute differences between the features of a block i and all the features of other blocks. This produced the VAM. The binary output of the background model was multiplied with the VAM to produce the final attention map. The authors also trained a neural network offline using object and background samples. The network was adapted online when sufficient data was available. Qualitative experimental results were presented for a few test sequences and a precision-recall curve was shown for the object tracker for a short video sequence.

Multiple levels of classifiers were implemented by Dawkins et al. for detecting, tracking and classifying objects within aerial video in the maritime domain [34]. First salient pixels were segmented using colour, edge information, and temporal variance. The salient pixels were then classified as wake or non-wake using a semi-supervised classifier and connected component labelling was run to generate a set of salient blobs. The second level of classification performed another level of wake suppression to create the final detections. The tracker associated the detections to current tracks or performed multi-frame analysis of the detections to create new tracks (by hypothesizing small linear motion based on a set of detections). SVM and Adaboost classifiers classified the tracks into known categories using a set of features for the objects, such as colour histograms and bags of words. The system could not handle land regions, failed when no ocean was showing and could not track when zoomed in to a target. The experimental results were mainly of a qualitative nature and showed some promise.

In another work, a pixelwise rigidity criterion was used to segment maritime targets [35]. Pixel trajectories were estimated for a video sequence and they were used to detect objects with rigid motion. This method relied on the observation that pixels belonging to the dynamic ocean would violate the rigidity assumption. The method was promising but it was computationally intensive for practical application. A large number of video frames was also needed for the trajectory computation.

2.2.3 Background Modelling

This section provides an overview of related work that motivated the proposed texture modelling approach. Pixel-wise representations model each pixel in the image frame. These models may or may not account for spatial dependencies between the neighbouring pixels. Parallel implementations of these models are also quite common. Region-based models describe entire

regions in an image using an appropriate data model.

Maritime video surveillance applications that have pixel-wise background models generally make use of Gaussian distributions to model the data [28, 32]. These methods are usually unsupervised. In some models the user has only to specify the number of components K in the mixture model; in other cases a predefined number of Gaussians are used. In the supervised case, known class samples are available for learning the model parameters. In some instances, binary class exemplars may be determined using scene assumptions for particular cues such as motion or colour.

One of the most popular pixel-wise background modelling techniques is the one developed by Stauffer and Grimson [37]. There are three variables that the user must set for their method: the number K of components in the model, the learning constant α, and the proportion of data, T, that the background model must account for. The data is modelled using a mixture of Gaussians.

Every new pixel value is checked against the existing K Gaussian distributions until a match is found. A distribution is regarded as a match if a pixel's value falls within 2.5 standard deviations of its mean value. When none of the K distributions match, the least likely distribution (the one furthest from the pixel) is replaced with a distribution that has the current pixel value as its mean, an initial high variance and a low prior weight.

The learning rate is computed from the distance between a data sample and the k-th normal distribution. It is used to update the parameters of the k-th normal distribution. During the background modelling process, distributions with low variances and high priors are preferred. The background model is represented as the first B distributions, ordered by the aforementioned preference, whose priors sum to greater than T. Heikkilä and Pietikäinen proposed a similar method where each pixel was modelled as a set of locally adaptive LBP histograms [38].

Sheikh and Shah used a kernel density estimator to model the image background [39]. It consisted of joint domain-range samples (x, y, r, g, b) where (x, y) was a location in the image, and (r, g, b) was the colour of the corresponding pixel in the image plane. This representation accounted for colour and spatial dependencies. The kernel density estimator also ensured that the representation was a valid probability distribution. Temporal persistence was proposed as a detection criterion and this was used to construct a foreground model. Both the background and foreground models were used competitively in a Maximum a Posteriori (MAP) Markov Random Field (MRF) framework to label pixels.

Sheikh, Javed and Kanade presented a method for background subtraction with a freely moving camera [40]. The trajectories of salient features in the video were analysed to build a sparse basis model of the background trajectories. The background was subtracted by removing trajectories that were spanned by the basis model, and salient foreground features could be

labelled subsequently. The labelled foreground and background trajectories were used to create foreground and background appearance models. These models were used in a MRF formulation to ensure smoothness in the labelling. This approach can be considered region-based because there are two global appearance models.

Yin and Collins proposed using the Motion History Image (MHI) for moving object localization in aerial thermal imagery. This method works well for images with rigid structured backgrounds because it uses frame differencing to compute motion changes [41]. The MHI approach effectively aligns a set of frames in a video and sums the differences between frames to create a motion energy map.

A dynamic texture is a sequence of images that exhibits stationarity properties over time. In everyday situations, some examples include sea waves, smoke and foliage. Computer vision models for dynamic textures are generally formulated for videos captured by static cameras; all the frames of the video are registered and optimization can be done in a straightforward fashion. A well-known example is the dynamic texture model of Doretto et al. that describes the time-varying spatial properties of stationary video signals [42]. An autoregressive model coupled with an eigenbasis representation of the data was used to describe dynamic textures such as the ocean and foliage. It is a particularly effective way of modelling the background in maritime scenes and, thus, can provide useful segmentation information to separate background and foreground. A work that does this can be found in [43]. The dynamic texture methods assume that an input video has been acquired with a static camera.

Chan et al. extended the dynamic texture model of Doretta et al. to a mixture of dynamic textures [1] as well as a generalized Stauffer-Grimson background model [44]. Clusters were learned by specifying initial seeds, using random trials for best parameters, or component splitting given the number of desired components. Segmentations were smoothed with a 5×5 majority filter. Several results were presented for synthetic datasets; qualitative results were presented for the real-world sequences (there was no ground truth for these sequences). Their contribution to background modelling used the dynamic texture model within the Stauffer-Grimson framework. The model was pre-trained on background data before it was tested and adapted online. The aforementioned models for dynamic textures were developed for static cameras and they fail if there is camera motion. Their model of a mixture of dynamic textures was tested on synthetic and real-world sequences with promising results. However, there are several points to note:

- The number of components were specified upfront. This is a realistic requirement for most applications and we do the same in this work.
- The temporal window size was crucial for learning the dynamics of the appearance. For example, in a fast changing scene, the window period is short to account for the changing dynamics. The authors used temporal window lengths of 20, 51 and 75 in real-world

sequences. Note that it is almost impossible to achieve this length of registered video frames in a situation where the camera is moving.

- The authors also assumed that the statistics of the mixtures do not change over time i.e. appearance and motion statistics stay the same.

Most dynamic texture models are limited by the static camera assumption — very little work has been done on moving dynamic textures. Camera sensors mounted on moving platforms are not guaranteed to have registered image frames under normal conditions. This presents a difficult situation because the problem then becomes ill posed. Huang et al. proposed a method for the registration of dynamic textures captured by a moving camera [45]. In the dynamic texture model of Doretto et al., the average image was computed directly and the temporal evolution was modelled by a dynamical system. A joint model was obtained for the appearance and the temporal dynamics. If the image frames are not registered, this method fails. Thus, it is desirable to have a model that accounts for the camera motion. In the work of Huang et al., powerful priors were proposed for the dynamic texture and the derivative distribution of the mean image. Well-registered images will have strong derivatives. The method first learned a prior for the average image and then it learned a prior for the dynamic texture. The matrix describing the appearance model is made time dependent to account for the camera motion. The model is then optimized over the camera motion model, average image and dynamic texture model — there are many variables and a Bayesian approach with gradient ascent is performed. The method was not feasible for real-time tracking or detection applications and it is not an online model.

Vidal and Ravichandran assumed small camera motion for their dynamic texture model [46]. A linear dynamical system combined with a 2D translation model estimated the optical flow of the dynamic texture. Their approach worked as follows:

- The appearance loading matrix was made time dependent.
- The camera motion was assumed to be very small and the loading matrix $C(t)$ for appearance, which is time varying, is approximately constant in a time window of size T.
- $C(t)$ was combined with a translation model. Thus, it captured both the appearance and camera motion.

This method is not feasible for our problem because the small camera motion assumption is easily violated.

The work of Voles et al. is the most similar to the methods proposed in this thesis [7, 8]. In their works, a single cluster is computed and it describes the appearance of the sea. This

cluster was used to detect foreground pixels that deviate from the model. The appearance features were calculated from grey level co-occurrence matrices. The appearance model is a global one.

An area that is directly related to our work on background texture models is feature space representation. How do we represent a set of features for a particular class, in high dimensional space, so that this representation may be used to infer a state or class label given a candidate feature? This question is addressed by some of the work done on computer vision models and pattern classification. Without going into details, some of the earlier ideas of this work are reported as they make an interesting avenue for additional reading or future work. Much of this can be found in the works of Bishop [47], Duda et al. [48], and Prince [49].

We distinguish between two types of computer vision models for feature vectors: *probabilistic* and *non-probabilistic*. Probabilistic models use statistical methods to compute the probability of observing a pattern or class. Non-probabilistic models, on the other hand, are not density based and generally use a set of functions or prototypes to compute a decision boundary for classification.

A popular density-based method is the GMM. It is a weighted sum of Gaussian distributions that describes multi-modal probability densities. The GMM is discussed in greater detail later in the next chapter. The Kernel Density Estimator (KDE) is a non-parametric function for estimating the probability density function of a random variable. It depends only on the data points in the training set. The KDE can be computationally inefficient as the density evaluation for a single sample is dependent on all the data points in the estimator. The binned KDE has been proposed to speed up computation; one may also consider a k-d tree data structure for finding neighbouring points in feature space. A simple way to represent a distribution of features is through the use of a histogram. In the case of multi-dimensional features, the histogram bin is a cell in multi-dimensional space. One may partition the feature space into cells using tree structures or clustering algorithms.

In recent years, the computer vision community has shown an increased interest in random trees, forests and ferns for classification tasks. These classifiers aim to learn the posterior distribution over class labels conditioned on the feature vectors. In the case of binary decision trees, there is a random binary test at each internal node in the tree that creates a path from the root to a leaf node for a feature. The paths that the various features take leads to a partitioning of the feature space. In the training phase, the labels of features are stored at the leaves. To improve the performance of the decision trees, random forests were proposed — these are several independent trees that each vote for the class label of a feature. In a random fern, the binary test function at each internal node is the same — this is extremely efficient for implementation.

A popular approach for unsupervised learning of a feature space is data clustering. Clustering

partitions a feature space into clusters such that features in the same cluster are more similar to each other than to features in other clusters. The mean-shift procedure is a robust feature space analysis method that does not require the number of clusters to be specified. We make use of data clustering to initialize the texture models. The interested reader can follow up on other advanced topics, such as SVM [50] and convolutional neural networks [51], that are not the focus of this thesis.

2.2.4 Feature Detection and Tracking

A few of the works that were discussed in the literature review on maritime target detection and tracking incorporated a measure of temporal stability for detecting maritime targets [4, 19, 20, 22, 25, 35]. In the work of Lee et al., candidate target regions that existed for a period of time were selected as true targets by the IR tracking system [4]. Westall et al. explicitly proposed TBD using dynamic programming for detection of slow moving persistent targets in their maritime video data. This approach was pixel-based and applied to videos that were first stabilized using software. Krüger and Orlov used TBD, maximally stable extremal regions and Shi-Tomasi feature tracking to detect targets, but they do not provide any implementation details or mention how these methods were combined [25]. Their algorithms were tested on thermal data rather than visual data. In the work of Bechar et al. a measure of rigidity based on optical flow and pixel trajectories was computed in the neighbourhood of each pixel [35]. The results were very promising but the method was also computationally intensive and the entire video had to be processed first before each frame could be labelled.

Most of our work on TBD is in the area of feature detection and tracking. TBD for maritime targets is implemented as keypoint tracking in video. The points in an image that perform very well for image correspondences are referred to as *interest points*, *keypoints*, *feature points* or *corners* and they have particular characteristics that are good for matching. These points are found using *feature detection*. Furthermore, the appearance of a small region of pixels around the interest point is captured in a vector called a *feature descriptor*. Feature descriptors contain image information such as raw intensity, edges, gradients or orientations. Correspondences between feature points in different image frames are found using feature *matching* or feature *tracking*. Many computer vision applications require sparse or dense point correspondences between a set of images. These correspondences can be used to:

- Align images for photo stitching.
- Perform software-based video stabilization.
- Compute camera pose.
- Estimate depth from stereo.

- Recognize objects.
- Estimate motion in video.

One of the best known and most popular works in feature detection and matching is by David Lowe [78]. His Scale-Invariant Feature Transform (SIFT) method is one of the most widely-used feature detection/descriptor techniques. The Harris corner detector is another well-known method that is fast and efficient for extracting feature points [79]. The Harris corner detector and SIFT descriptor are employed in this work and they are discussed in detail in Chapter 6. Perhaps the earliest and most well-known of feature trackers is the Kanade-Lucas-Tomasi (KLT) tracker. It is based on the work of Lucas and Kanade [80], which presents a method for tracking an image patch, and Tomasi and Kanade [81], who describe how to select best features for tracking using a criterion closely linked to Harris corners.

2.3 Significance of the Proposed Work

The following observations are made from the literature review on maritime target detection and tracking:

- Methods that rely on static cameras mainly employ background modelling techniques and frame differencing for target detection [7, 10, 17, 21, 24, 28, 29, 32], or supervised detectors [5, 6, 18, 27, 30, 31].
- Previous works that detect maritime targets from a moving platform generally rely on the discriminating characteristics of colour [11, 12, 14, 15, 16, 19, 20, 23, 33, 34] or thermal [23, 25, 26] data for segmenting the scene and detecting targets. Some systems also used supervised detectors [22, 23, 33, 34].

The proposed work is novel and significant in that it addresses the gaps in the literature for our particular problem:

- We detect targets in grey scale video by modelling the textural appearance of the ocean — we do not make use of colour or thermal data as is commonly done. The preference for a monochrome sensor was explained earlier in the chapter.
- Our approach is unsupervised as we neither know before hand what the potential targets look like nor do we have training exemplars for the ocean appearance.
- The method is applicable to scenes where the camera is static or moving.

Our work on texture appearance models is similar to Voles [9]. However, we propose the following variations and improvements:

- The texture window size stays fixed when the descriptors are being computed. The work of Voles varied the window size to account for perspective changes — it was assumed that a single texture class described the ocean. We assume there are multiple texture classes and, thus, the window size is kept fixed because the model's different texture components account for the variations in the ocean's appearance.
- The LBP texture descriptor is used in this work rather than features derived from the grey level co-occurrence matrices.
- Our model describes the feature space as multi-modal rather than uni-modal.
- The texture model is updated online to account for changes in the feature space.
- A graph cut method is used to improve the image labelling during target detection.

The last part of this thesis examines feature tracking as a means of establishing stable and persistent objects on the ocean. A few authors have considered this as a component of maritime target detection and tracking [19, 20, 25, 35]. Our work examines the feasibility of this approach for our test data. A GPU implementation of a feature tracker is described for a track-before-detect concept.

2.4 Summary

This chapter has presented prior work in the problem area that is the focus of this thesis. The literature survey has been used to highlight the challenges and limitations in the field. We have also highlighted the significance of the proposed work within the aforementioned context. In the next chapter a background model is proposed for maritime scenes captured by a moving camera.

Chapter 3

Maritime Scene Background Modelling

3.1 Introduction

In real world scenarios, it is impractical to manually initialize a target of interest for object tracking. This approach fails when the target is very small, moving fast and/or the camera platform is also moving. The process of manually putting a bounding box around an object that will be tracked is cumbersome and prone to error. For fast moving objects, manual initialization may be impossible. To this end, a system that automatically detects and tracks potential targets has several advantages over the aforementioned efforts.

Unsupervised object detectors are generally preferred for automatic target detection because they directly model the data that is captured and attempt to infer object presence without the need for training data. This chapter advocates that approach. A method is proposed for modelling the statistical visual texture properties of the ocean. This background model is used to identify anomalies that are present in the visual texture. In the ideal case these anomalies, or outliers, are potential targets of interest that can be transformed into measurements for a tracking system.

Optical detection and tracking of targets on the ocean is a difficult problem, more so when the camera is moving in long range surveillance. There are several aspects of the problem to be considered:

- The ocean is a dynamic background with complex appearance changes caused by the motion of the water and its spectral properties.

- Different types of appearances are caused by:
  - White caps (foam) on the ocean.
  - Sea swells.
  - Water disturbance, such as a wake, caused by vessels.
  - Uneven lighting, such as shadows and glint, caused by the angle of the sun to relative to the surface of the ocean and the viewing direction.
  - Low contrast in the scene.
- Partial and full occlusion of a target in deep waves can make target detection difficult.
- Camera motion introduces additional complexity to the problem and it can arise from:
  - A pan and tilt unit that scans the ocean for interesting targets. This unit may be on the land, or on the ocean itself e.g. mounted in a patrol boat.
  - Strong wind conditions that move the camera system.
  - The movement of the water in the ocean that moves the vessel the camera is situated on.

We propose a region-based texture model for capturing the ocean's spatial appearance in video. Each homogeneous ocean region in an image is described by a texture component; a set of components provides a global description of the entire ocean. At the core of our approach is the use of a histogram-based texture descriptor. We show through experimental analysis that the methods are suitable for both static and moving cameras under certain scenarios. Although we do not model the temporal dynamics of the scene, we show experimentally that the spatial statistics are sufficient for detecting maritime targets by comparing our work to existing methods. The proposed models have the following characteristics:

- **Region-based:** The model is region-based and does not consider the temporal dynamics of the ocean. We forgo inclusion of a temporal model for appearance because frame-to-frame registration of a dynamic texture acquired with a moving camera is a difficult problem (the challenges in this regard are discussed in the next section). Region based models offer a degree of invariance to changes in translation and rotation since they do not have to be registered to individual pixels. Furthermore, our model exploits the repetitive nature of the ocean texture and constructs a global model of the ocean's appearance as in the work of Voles [9].
- **One-class Classification:** The texture model can be considered a one-class classifier; the class that is modelled is the ocean appearance. Outliers of this model are potential targets of interest.

(a) (b)

Figure 3.1: Two examples of the maritime target detection scenarios that this work addresses.

- **Static or Moving Camera:** The global appearance model of the ocean is suitable for static or moving cameras.
- **Unsupervised:** The model uses unsupervised learning and simple heuristics about the nature of the scene to model the appearance of the ocean.
- **Online:** The model is updated online to accommodate appearance changes over time.

In the next section, a formulation is presented for the detection of maritime targets in video.

3.2 Formulation

The background texture models that are proposed are suitable for maritime scenes where the majority of the video frame contains the visual ocean texture. For the particular problem that we address, the scene typically contains one or two targets and the camera platform may be moving and unstabilized. This scenario is typical in long range surveillance of targets far out at sea, and optical systems with medium field of view cameras. Two examples of these scenarios are shown in Figures 3.1(a) and 3.1(b). The proposed solution for detecting these targets has several characteristics.

Suppose we have a set $\mathbf{X}$ of random variables $\mathbf{x}_i$, where i is the index of the measurement in the set and generally denotes a 2D location on the image plane. The $\mathbf{x}_i$ are measurements of image properties such as intensity, colour, texture or motion. Suppose further that we have 2 classes: $w = 0$ that denotes the background and $w = 1$ the object class, and we wish to assign each $\mathbf{x}_i$ to one of these classes. If a class conditional probabilistic data model for the background exists, $\mathbf{x}_i$ can be assigned to $w = 0$ when the following is true:

$$p(\mathbf{x}_i|w = 0) > \gamma\tau_p \tag{3.1}$$

and it can be assigned to $w = 1$ when the condition is false. Here, the threshold τ_p is computed from the data samples and $\gamma \in \mathbb{R}$ is a user-defined weighting factor that controls the sensitivity of the classification to background or foreground. This is a probabilistic model because we are describing the class conditional density model. One may also consider a non-probabilistic formulation such as

$$d(\mathbf{x}_i) < \gamma\tau_d \tag{3.2}$$

where $\mathbf{x}_i$ is assigned to $w = 0$ if its distance to some prototype or set of vectors is less than $\gamma\tau_d$. We can refer to the above formulation as a one class classifier or anomaly detector. In general, the threshold $\tau = \tau_p$ or $\tau = \tau_d$ is computed from the data samples. In later sections we will mainly refer to τ and assume that γ is implicit.

Thus, we are concerned with detecting the *outliers* of our texture model. It is apt to mention the definition of an outlier as put forth by Hawkins [52]:

> *An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.*

It is desirable to have a method that builds a model of the background, in an unsupervised fashion, and uses this model to detect outliers that correspond to potential maritime targets.

One of the main characteristics of our background models is that we describe the global ocean appearance as a set of K texture components. Each component k is different from the others in appearance and it describes local regional properties of homogeneous pixels. Furthermore, we propose updating these region-based components in a sequential fashion as new image frames become available. A region-based model, rather than the popular pixel-wise models, is preferred because precise frame to frame image registration can be avoided and one can account for camera motion. As a result of this, the temporal dynamics of the texture appearance are not modelled. Another advantage of our method is that stationary and moving objects can be detected.

These K texture components can be the components of a GMM or a set of characteristic prototypes determined, for example, by a clustering process. The set of K components will also have associated parameters $\boldsymbol{\theta}$. The regional properties of a texture component are summarized by a histogram descriptor. In this work we make use of the LBP descriptor.

The proposed approach considers a set of simple heuristics for modelling the ocean background:

- It is assumed that a typical scene contains only ocean with or without targets. In some scenes, a part of the sky may be present and this will lead to the presence of a horizon line. Thus, we assume that a majority of the texture data in a scene will originate from the ocean. If we have K texture components explaining the scene, the components that explain the most data will likely be ocean components. In other words, if a single

component explains a large fraction of the data, then this component accounts for the ocean. Large components are preferred for the background model.
- If a texture component accounts for a very small fraction of the data, it is likely to be a target.

These heuristics are reasonable for most maritime scenes with a small number of targets. Thus, if a set of K components can be learned such that they account for a large fraction of the scene data, the outliers of this model will be potential maritime targets. We need only learn the background model — inliers of this model are the ocean class and outliers are objects of interest. Our formulation shares much in common with the works of Stauffer and Grimson [37], Sheikh and Shah [39], Voles et al. [7, 9], and Szpak and Tapamo [28]. Most of the aforementioned approaches are pixel-wise models; our model is region-based and is related to the works of Voles et al. [8, 9]. The details of our methods are provided in the next section.

3.3 The Proposed Methods

Our methods can be described as the following sequence of functions:

- Feature extraction.
- Pixel classification i.e. foreground detection.
- Background model update.

In this work, feature extraction refers to the process that transforms a subset of the observed video data into a vector of descriptive numerical values. The resulting feature vector is associated with a pixel location (x, y) on the image plane; it describes the particular visual characteristics of the image plane at the aforementioned location. It is assumed that the feature vector is computed within a suitable window size. We use the LBP descriptor to describe the image texture characteristics. The patterns of the LBP descriptor are textons, assumed texture units, and they are summarized in a histogram of texton frequencies.

Once texture features have been computed for each pixel, we classify the pixels using the background model. If the model has not been created (e.g. at the start of the video), it is initialized using the extracted texture features and the pixels are classified. If the model exists, pixels are classified and the model is updated.

Two representations are examined for the background model: the well-known GMMs, and a STM that consists of histogram feature distributions and no additional parameters. The main idea that we advocate is the reduction of the image feature space into a few significant components that are described by our particular model. The LBP texture features lie in a certain part of the multi-dimensional image feature space and the dense regions of this space, which are also referred to as clusters or peaks, correspond to the significant features in a scene.

We aim to describe these regions in the data. The background model is updated online, for each video frame, in a sequential fashion. These processing steps are now discussed in detail.

3.3.1 Feature Extraction

Texture features are extracted from 8-bit grey scale data. Chapter 2 explained the reason for using monochrome data for the particular application area. A number of image features have been proposed over the years for describing the particular characteristics of a scene. More often than not, when hardware resources are limited a pragmatic approach is preferred for the choice of image features. A suitable texture descriptor is required for the characteristics of the ocean appearance.

The performance of most image processing and computer vision methods is highly dependent on the quality of information that can be extracted from the images. In the case of a background model for maritime scenes, one seeks features that strongly separate the target from the background. There are a multitude of features that one may consider, per-pixel or regionally, for this particular problem (see, for example, Sonka et al. [53] or Szeliski [54]):

- Raw pixel intensity (visual or infrared).
- RGB colour.
- Statistical features such as entropy and homogeneity.
- Motion features.
- Features computed using filters such as Haar, Gabor or Sobel.
- Descriptors such as LBP, HOG, SIFT or histograms using some of the above features.

A feature vector $\mathbf{x}_i$ that is associated with each pixel in an image contains a subset of the above types of features.

The visual appearance of the sea can be characterized as texture as it does not demonstrate uniform intensities; there are varying degrees of tonal content that have a repetitive nature. Real world image data exhibits many variations as a result of the particular surface properties of an object. Sensor noise is also a contributing factor that prevents surfaces that appear uniform to the eye from being uniform in a digital image. In image data, the periodic arrangement of pixel values that shows different levels of visual fineness and coarseness is referred to as texture.

A histogram is an estimate of a probability distribution. It consists of bins — these are non-overlapping intervals of values — that count the number of values falling into the bin. It is important to remember that values being counted in a bin may be multi-dimensional.

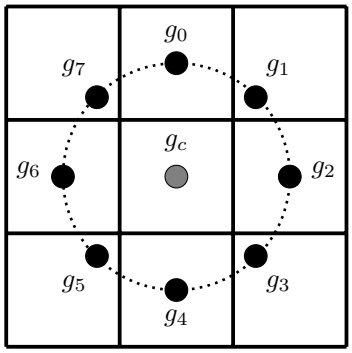


Figure 3.2: LBPs.

Histograms are generally normalized so that the values across all bins sum to one. This provides a summary of the frequency of occurrence of particular elements. Histograms are frequently used to estimate probability distributions for cues such as grey level intensity, color, edge orientations and magnitudes, texture and textons. For the particular area of maritime target detection, the distribution of LBPs is proposed as an effective texture feature for the sea appearance.

The following discussion on LBP features is for a monochrome image and is based on the work of Ojala and Pietikäinen [55]. For a P-bit pattern, the joint distribution G describes the texture:

$$G = \{g_0 - g_c, g_1 - g_c, g_2 - g_c, ..., g_{P-1} - g_c\} \tag{3.3}$$

where the g_i are evenly spaced apart on a circle and at distance R from a center pixel g_c. This texture operator is highly discriminative — it can describe edges, spots and constant areas in an image. Figure 3.2 shows the grey level arrangement for $P = 8$ and $R = 1$ (a 3×3 neighbourhood). Grey values $\{g_2, g_4, g_6, g_8\}$ may be calculated using interpolation since they do not lie at the center of the pixel.

A local binary intensity invariant pattern can be generated with

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i \tag{3.4}$$

where

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \tag{3.5}$$

In this work, the original 3×3 neighbourhood is used where $P = 8$ and $R = 1$. The number of binary patterns is reduced by using uniform patterns. A uniformity measure is defined as the

number of bitwise transitions in the pattern

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| \tag{3.6}$$

$$+ \sum_{i=1}^{P-1} |s(g_i - g_c) - s(g_{i-1} - g_c)|. \tag{3.7}$$

Patterns with a uniformity measure of at most 2 are used and this provides $P(P-1)+3$ bins [56]. A unique label is assigned to each pattern with $U \leq 2$ and all other patterns are grouped under one label. This results in 59 patterns. In practice, the mapping to uniform pattern label is performed using a lookup table. The number of patterns can be reduced even further by making them rotation invariant but our preliminary exploration showed that this did not work well for our application — the texture direction cues for foreground and background separability are lost in the rotation invariant representation. The descriptor may also be extended to create a joint distribution of LBP and contrast information at the expense of greater storage and computational requirements.

A feature descriptor $\mathbf{x}_i$ is computed by generating the LBP for each pixel in a window. The lookup table is used to determine the unique uniform pattern label. Once this is done, the histogram descriptor is populated with all the patterns in the image window and normalized so that it sums to one — this facilitates comparison to other descriptors. Each LBP contributes a count of one in the unnormalized histogram. There are several advantages when using the LBP texture descriptor:

- The LBP is invariant to monotonic changes in grey level intensity.
- It describes spots, edges and other types of structures.
- The LBP is fast to compute and maps directly to a histogram bin. There is no need for additional processing such as space partitioning to determine histogram cells.
- The histogram distribution provides a convenient and powerful way to summarize the structure in an image patch.

3.3.2 Texture Similarity Measures

A similarity measure is a mathematical function that quantifies the similarity of two feature vectors. Similarity measures are used to quantify error, group similar objects or compute the probability of observing a particular sample. The similarity measures used in this work are now described for two feature vectors $\mathbf{p}$ and $\mathbf{q}$ that summarize the texture properties of image regions using histograms. In the following similarity measures a value of zero implies that $\mathbf{p}$ and $\mathbf{q}$ are the same.

There are several similarity measures that one may consider for comparing two LBP histograms. If the histogram is viewed as a multi-dimensional point in Cartesian coordinates, the Euclidean distance measure, also known as the L^2 norm, is a common metric for comparing two points. The Euclidean distance for an m-dimensional point is

$$D_{EUC}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=0}^{m-1} (p_i - q_i)^2}. \tag{3.8}$$

For particular algorithms, one may exclude the costly square root operation. Although this distance measure is commonly used in practice it is worth noting that its interpretation in high dimensional space is not obvious. This topic is touched on later in the section on high dimensionality and interested readers are pointed to Zimek et al. for additional reading [57]. The L^2 norm does not account for correlations in the data.

The Mahalanobis distance, of which the L^2 norm is a particular form, computes the distance between a random vector $\mathbf{p}$ and the centroid $\mathbf{q}$ of a distribution with covariance matrix $\mathbf{\Sigma}$:

$$D_{MAH}(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^{\mathrm{T}} \, \mathbf{\Sigma}^{-1} \, (\mathbf{p} - \mathbf{q})}. \tag{3.9}$$

This reduces to the Euclidean distance when $\mathbf{\Sigma}$ is the identity matrix. The similarity measure D_{MAH} accounts for the correlations in the data set. This means that the distance measure is computed in a transformed coordinate system where the vectors are normalized.

For probability distributions, there also exist specific measures for the difference between two distributions. One such measure is the Kullback-Leibler (KL) divergence for discrete probability distributions

$$D_{KL}(\mathbf{p}, \mathbf{q}) = \sum_{i=0}^{m-1} p_i \ \log \frac{p_i}{q_i}. \tag{3.10}$$

One should note the following for the KL divergence:

- D_{KL} is not a symmetric measure for variables $\mathbf{p}$ and $\mathbf{q}$.
- We use the natural logarithm in Equation 3.10.
- The histograms must be normalized i.e. sum to one.
- Whenever $p_i = 0$ we have $p_i \log \frac{p_i}{q_i} = 0$.
- When $p_i \neq 0$ and $q_i = 0$, $D_{KL}(\mathbf{p}, \mathbf{q}) = \infty$.

To work around the problematic $q_i = 0$, after computing the unnormalized LBP histogram, we increment the count of each bin by one before normalization. This ensures that there are no zero frequencies in the histogram and it was found to work well in practice. It is not recommended that one ignore zero terms or add tiny values to the frequencies during the KL

divergence computation.

Another effective similarity function for histograms is the chi-square distance

$$D_{CHIS}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{m-1} \frac{(p_i - q_i)^2}{p_i + q_i}. \tag{3.11}$$

3.3.3 Gaussian Mixture Model

A sample set of feature vectors extracted from the ocean class is distributed in a large multi-dimensional space. This distribution is generally multi-modal. A single Gaussian function will not approximate the distribution of these points very well. A probability distribution with multiple modes can be represented by a mixture of Gaussian distributions.

The probability of observing a sample $\mathbf{x}_i$ using a GMM is

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.12}$$

where

$$\mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{m}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)} \tag{3.13}$$

is the normal distribution. In the mixture model, K is the number of components, π_k the prior weight for component k, and m the length of the feature vector. The prior weights satisfy

$$\sum_{k=1}^{K} \pi_k = 1. \tag{3.14}$$

Each Gaussian or normal distribution k is fully described by the mean vector $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$.

The GMM is used to describe the ocean appearance. Each mixture component, when initialized appropriately, will summarize the textural characteristics of a particular part or parts of the ocean. The mixture model is learnt using all the feature descriptors in an image frame. The mean vector of a component is the average texture feature distribution; the covariance matrix describes how the LBPs of the mixture component — texture elements such as lines and spots — vary around the mean feature distribution. The mean vector or feature distribution $\boldsymbol{\mu}_k$ describes the general appearance of the k-th component.

A fully parametrized $\boldsymbol{\Sigma}_k$ increases the complexity of the model. It is quite common to make the matrix $\boldsymbol{\Sigma}_k$ diagonal or proportional to the identity matrix. When the latter option is chosen it corresponds to spherical clusters. A diagonal covariance matrix has several computational advantages, provided that none of the diagonal elements are zero:

- It implies that the vector components in $\mathbf{x}_i$ are independent; thus, the normal distribution can be represented by the product of single one-dimensional functions.
- The determinant $|\mathbf{\Sigma}_k|$ can be computed by taking the product of the diagonal elements.
- The inverse $\mathbf{\Sigma}_k^{-1}$ is given by the reciprocal of the diagonal elements.
- The vector components are independent when $\mathbf{\Sigma}_k$ is diagonal, simplifying the online update of the model because the Gaussian functions can be updated independently and in parallel if desired.

When using a diagonal covariance matrix, the dependency relationships between vector components are lost. The full covariance matrix provides a rich description of how one type of LBP varies with another. This loss of information should be explored when deciding between the full and diagonal matrix. Our model uses a diagonal covariance matrix and the vector components are updated independently.

In most computer vision problems, $\mathbf{\Sigma}_k$ can become singular if some of its elements are zero. Singular matrices are non-invertible and this is problematic for evaluating the probability function. One may perform a simple regularization on the diagonal matrix to obtain

$$\mathbf{\Sigma}'_k = \mathbf{\Sigma}_k + \lambda \mathbf{I} \tag{3.15}$$

where the regularization parameter λ ensures that the diagonal entries are non-zero. In this work, λ is set empirically.

The parameters of a Gaussian mixture model are usually estimated using the EM method. The EM algorithm is an iterative method for finding the MAP or maximum likelihood estimates of parameters in a statistical model. In the EM algorithm, unobserved latent variables are introduced into the model. One can model a complex probability density function, here in discrete form, over data $\mathbf{x}_i$ for parameters $\boldsymbol{\theta}$ by using a hidden latent variable $\mathbf{z}$:

$$\begin{aligned} Pr(\mathbf{x}_i|\boldsymbol{\theta}) &= \sum_k Pr(\mathbf{x}_i, z = k|\boldsymbol{\theta}) \\ &= \sum_k Pr(\mathbf{x}_i|z = k, \boldsymbol{\theta}) Pr(z = k) \end{aligned} \tag{3.16}$$

In the GMM, the latent variable specifies the mixture component that a data sample belongs to.

The basic steps of the EM algorithm are (taken from Bishop [47]):

1. Choose initial values for $\boldsymbol{\theta}^{\text{old}}$.

2. E-step: Evaluate $Pr(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ where $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{n-1}\}$, $\mathbf{Z} = \{\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_{n-1}\}$ and n is the number of feature vectors.

3. M-step: Evaluate $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{Z}} Pr(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ \log \ Pr(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}))$

4. Check for convergence of either the log-likelihood or the parameter values. If the convergence criterion is not satisfied, then $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$ and return to step 2.

In the GMM, it can be shown that the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ maximizing Equation 3.16 are the sample mean and sample covariance respectively. The EM for GMM works as follows for n data samples and K components (see Bishop for more details [47]):

1. Initialize $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and π_k.

2. E-step: Compute the responsibilities

$$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=0}^{K-1} \pi_j \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

3. Recompute the parameter values using the current responsibilities:

$$n_k = \sum_{i=0}^{n-1} r_{ik} \tag{3.17}$$

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{n_k} \sum_{i=0}^{n-1} r_{ik} \mathbf{x}_i \tag{3.18}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{n_k} \sum_{i=0}^{n-1} r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_i - \boldsymbol{\mu}_k^{\text{new}})^{\text{T}} \tag{3.19}$$

$$\pi_k^{\text{new}} = \frac{n_k}{n}. \tag{3.20}$$

   where n_k is the effective number of points assigned to the k-th distribution.

4. Evaluate the log-likelihood

$$\sum_{i=0}^{n-1} \ln \left\{ \sum_{j=0}^{K-1} \pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \right\}$$

   and check for convergence. If there is no convergence return to step 2.

In our implementation, we use the median rather than the mean of squared differences to compute the variance of a vector element. The diagonal covariance matrix makes this straightforward. This is done for robustness to outliers.

The EM approach is costly if one does the full probabilistic interpretation with soft data associations. Moreover, an online approach is desired for frame by frame video processing. Our Gaussian mixture model is updated online in a similar fashion to Stauffer and Grimson using hard data association [37]. The online appearance model of Jepson et al. also has a very similar approach [58]. These ideas have been adapted for the proposed region-based background model.

Assume that we have a total of n samples $\mathbf{x}_i$ and each sample has been assigned to a mixture component k. The number of samples assigned to each component k is n_k. In this work, hard assignments are performed — this means that each sample $\mathbf{x}_i$ is assigned to one component only. A sample is assigned to the k-th component with the smallest Mahalanobis distance:

$$k = \arg\min_j D_{MAH}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

Now, given the n_k samples assigned to each component k, we explain the sequential update of the parameters of component k. Dropping subscripts from the vectors and matrices, denote

$$\begin{aligned}\mathbf{x} &= [x_0, \cdots, x_{m-1}] \\ \boldsymbol{\mu} &= [\mu_0, \cdots, \mu_{m-1}] \\ \boldsymbol{\Sigma} &= \text{diag}\left(\sigma_0^2, \cdots, \sigma_{m-1}^2\right)\end{aligned}$$

where m is the dimension of the feature vector. The updating of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with $\boldsymbol{x}$ is shown next.

We have the learning rate parameter α and the sample weight parameter $\beta = \frac{1}{n_k}$. The parameter α is fixed during target detection and defines the speed at which the parameters change; β is a weight defining the amount of contribution of a sample to the parameter update. Using each sample $\mathbf{x}$, an element-wise update is performed sequentially on the parameters of component k. This is possible due to the independence of the diagonal covariance matrix. For each sample $\mathbf{x}$ compute

$$\rho_i = \alpha\ \beta\ e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}}, \tag{3.21}$$

the weight of the causal low pass filter for online update of the mean and covariance elements, where $i \in \{0, \cdots, m-1\}$, and update the elements of the mean and covariance matrix:

$$\mu_i^{new} = (1-\rho_i)u_i^{old} + \rho_i x_i \tag{3.22}$$

$$\sigma_i^{new} = (1-\rho_i)\sigma_i^{old} + \rho_i(x_i - \mu_i^{new})(x_i - \mu_i^{new}). \tag{3.23}$$

After processing all the samples, update the prior and classification threshold

$$\pi_k^{new} = (1-\alpha)\pi_k^{old} + \alpha\frac{n_k}{n} \tag{3.24}$$

$$\tau_k^{new} = (1-\alpha)\tau_k^{old} + \alpha\tau_k', \tag{3.25}$$

where τ_k' is computed as the median of the Mahalanobis distances of the samples assigned to the k-th component. The parameter ρ gives more weight to samples that are close to the mean. Samples that are far away from the mean, such as outliers, will have a very small contribution to the evolution of the parameters. The mean distribution $\boldsymbol{\mu}$ is normalized to sum to one after a sequential update.

The general aspects of the method that we have described here are known in the literature. However, the novelty of our approach lies in its application to histogram data, the sequential update scheme and how we compute the threshold for target detection. In the next section we look at a simple model that consists of just a set of feature distributions.

3.3.4 Sparse Texture Model

In addition to using a GMM to describe a maritime background, it is proposed that one may directly use a set of K histograms to model the background. This model stays close to the definition of the texture feature distributions of Ojala and Pietikäinen [59] and we refer to it as the STM:

- Each component is a histogram $\mathbf{h}_k$, with a prior weight π_k that quantifies the amount of data the histogram explains. Larger values are preferred for ocean textures.
- Features are compared to the model components using D_{CHIS}.
- Each component k has an associated ball of radius τ_k. A feature $\mathbf{x}_i$ is classified as ocean if its distance to the nearest histogram in the STM is less than τ_k.

Analagous to the GMM, each $\mathbf{h}_k$ describes a particular textural characteristic of the ocean. The parameter τ_k can be viewed as the classification boundary around $\mathbf{h}_k$. The STM is somewhat related to the work of Ypma and Duin on support objects for domain approximation [60]. The set of K histograms and associated balls can be viewed as a type of domain approximation. Unlike the k-centers method, the balls here have different radii. A texture feature $\mathbf{x}_i$ is assigned to a histogram in the model having the smallest chi-square distance. The classification threshold τ_k is the median of chi-square distances of samples assigned to component k.

The STM is updated online in a sequential fashion in a similar way as Heikkilä and Pietikäinen, although they use a pixel-wise LBP histogram model [38]. An important difference compared

to their method, apart from ours being region-based, is that our detection threshold varies with time. Once again, assume we have n_k samples with hard assignments for the k-th component. Introduce the learning rate parameter α and the sample weight parameter $\beta = \frac{1}{n_k}$. The parameter α is a fixed parameter during target detection and defines the speed at which the parameters change; β is a weight defining the amount of contribution of a sample to the parameter update. For each sample $\mathbf{x}_i$ assigned to the k-th component, compute

$$\rho = \alpha \ \beta \ e^{-\frac{D_{CHIS}(\mathbf{x}_i, \mathbf{h}_k)}{\tau_k}} \tag{3.26}$$

and then update

$$\mathbf{h}_k^{\text{new}} = (1 - \rho) \ \mathbf{h}_k^{\text{old}} + \rho \ \mathbf{x}_i. \tag{3.27}$$

After updating a histogram sequentially, the prior and classification threshold are updated in the following manner:

$$\pi_k^{new} = (1 - \alpha)\pi_k^{old} + \alpha \frac{n_k}{n} \tag{3.28}$$

$$\tau_k^{new} = (1 - \alpha)\tau_k^{old} + \alpha \tau_k' \tag{3.29}$$

where τ_k' is the median of the chi-square distances of the samples $\mathbf{x}$ assigned to the k-th component. Once again, π_k indicates the fraction of data that the component explains.

Histogram normalization is not performed after the update because both the $\mathbf{x}_i$ and the $\mathbf{h}_k$ sum to one:

$$\begin{aligned}
\sum_{i=0}^{m-1} h_i &= \sum_{i=0}^{m-1} [(1 - \rho) \ h_i + \rho \ x_i] \\
&= \sum_{i=0}^{m-1} (1 - \rho) \ h_i + \sum_{i=0}^{m-1} \rho \ x_i \\
&= (1 - \rho) \ \sum_{i=0}^{m-1} h_i + \rho \ \sum_{i=0}^{m-1} x_i \\
&= (1 - \rho) \ 1 + \rho \ 1 \\
&= (1 - \rho) + \rho \\
&= 1,
\end{aligned} \tag{3.30}$$

where h_i is the frequency in bin i of the histogram.

3.3.5 Model Initialization

Most computer vision models are sensitive to the method of parameter initialization. It is common practice to run random trials to determine the best set of parameters for a model. The

K components of the background model are initialized using samples assigned to K homogenous clusters. The most common way, perhaps, for grouping features into homogenous clusters is data clustering. The cost function that most clustering methods optimize maximizes inter-cluster distances and minimizes intra-cluster distances.

The MacQueens k-means algorithm is a popular clustering algorithm in which the number of clusters (K) is known [61]. It is an iterative process that assigns patterns to the closest cluster using a distance function. The k-means algorithm treats all variables equally in deciding the cluster membership of a pattern. It can also be described as an algorithm that computes a hard partition — each pattern can belong to one and only one cluster. A disadvantage of the algorithm is its sensitivity to the prototype initialization. However, it has rapid convergence and is computationally efficient.

One may overcome the limitations of the k-means random cluster initialization step by considering the k-means++ seeding method as proposed by Arthur and Vassilvitskii [62]:

- Choose a center $\mathbf{c}_0$ randomly from X.
- Select the next center $\mathbf{c}_i$, choosing $\mathbf{x} \in X$ with probability $\frac{D(\mathbf{x})^2}{\sum_{\mathbf{x} \in X} D(\mathbf{x})^2}$.
- Repeat the above until K centroids have been selected.

The distance $D(\mathbf{x})$ denotes the shortest distance from $\mathbf{x}$ to the closest center that has already been chosen.

Interestingly, the k-means algorithm also has a sequential update mode for updating a cluster prototype [61]:

$$\mathbf{c}^{new} = (1.0 - \alpha)\mathbf{c}^{old} + \alpha \mathbf{x}_i \tag{3.31}$$

where α is the learning rate. The learning rate decreases monotonically as more samples are used. Note that this corresponds to the learning rule for the mean vector of a Gaussian distribution and the histograms of the STM.

The k-means method uses the Euclidean distance to compare samples. Although this works in practice for some types of data, there are better dissimilarity measures for histogram features such as the KL divergence. The standard k-means algorithm is a least-squares estimator and it minimizes the within cluster sum-of-squares-objective. It is not trivial to change the distance function to an arbitrary one because the centroid computation is compromised and the algorithm may not converge. However, if the centroid is chosen to be equal to one of the data vectors in the cluster, any similarity function may be used.

We use the information-theoretic feature clustering algorithm of Dhillon, Mallela and Kumar [63]. The method is suitable for high dimensional features, such as histograms, that are of a distributional nature. It maximizes the between-cluster Jensen-Shannon divergence and

minimizes the within-cluster Jensen-Shannon divergence. The clustering method bears some resemblance to the k-means algorithm and this is how we implemented it:

1. Define the number of clusters K.
2. Initialize the cluster starting points $\mathbf{c}_j$ for $j = 1, \ldots, K$ using k-means++.
3. Assign a pattern $\mathbf{x}_i$ to the nearest cluster $\mathbf{c}_j$: $\mu_{ij} = 1$ if $D_{KL}(\mathbf{x}_i, \mathbf{c}_j) \leq D_{KL}(\mathbf{x}_i, \mathbf{c}_t)$ for $1 \leq t \leq K$; otherwise $\mu_{ij} = 0$ for $t \neq j$. The variable μ_{ij} is a binary variable that indicates the membership of $\mathbf{x}_i$ to cluster $\mathbf{c}_j$.
4. Recompute $\mathbf{c}_j$:
$$\mathbf{c}_j = \frac{\sum_{i=0}^{n-1} \mu_{ij} \mathbf{x}_i}{\sum_{i=0}^{n-1} \mu_{ij}} \tag{3.32}$$
for $1 \leq j \leq K$. Normalize the histogram $\mathbf{c}_j$.
5. Repeat steps 3 and 4 until the centroids do not change or a maximum number of iterations has been reached.

Our method differs from Dhillon et al. in how the cluster initialization is done. They create a set of clusters and then do splitting or merging of the clusters to ensure that the support set of every feature histogram is contained in the support set of at least one cluster distribution. This ensures that every feature histogram is part of some cluster and that at least one KL divergence for a feature is finite. We ensure that the KL divergence is finite by seeing that there are no zero bins in the LBP histogram (see Section 3.3.2); clusters are initialized using the k-means++ with D_{KL}.

Once the clustering has been performed, a model can be initialized using the samples assigned to each cluster. The clustering is run only once at the start, for the first video frame, to initialize the model. Thereafter, each texture component adapts online in a sequential fashion. During model initialization, there are some scenarios to take note of:

- **Absence of targets:** If there are no targets in the scene, the initialization works in a straightforward fashion and the components, when initialized, will describe the ocean texture appearance.
- **Presence of targets:** In typical scenarios, there is a possibility that one or two targets will be present in the scene. This can happen when the system starts up or has to reset. It is argued that when the ocean makes up a majority of the scene, after a few frames of updating, the average distribution of the components will not be significantly affected by the targets (outliers). This is attributed to the histogram representation of the texture properties and the statistics of the scene.

- **Presence of sky:** If a sky is present, and consists of homogenous texture, the initialization may proceed and it is expected that one texture component will describe the sky. If the sky texture is not homogenous, horizon detection should be performed to localize the ocean.

3.3.6 Model Maintenance

Each time the background model is updated, it must be monitored to ensure that it is in fact describing the ocean appearance. Define π_{min}, the minimum weight for an ocean texture component. A simple approach that monitors the weights π_k is used. After each component update step, the following checks are done:

- If there are no samples assigned to component k, the weight π_k will decay according to

$$\pi_k^{new} = (1 - \alpha)\pi_k^{old}. \tag{3.33}$$

- If $\pi_k < \pi_{min}$, the k-th component is re-initialized using samples assigned to it and π_k is set to π_{min}. If there are no samples to initialize the component, π_k remains unchanged and will decay to zero. The component is marked as inactive until there are samples to initialize it. If at each update step π_k drops below π_{min} and is re-initialized successfully, the component is typically drifting and will stabilize once π_k increases to above π_{min}.
- If $\pi_k > \pi_{min}$ the component is marked as active.

The simple strategy outlined above works well for our test cases. One may also explore monitoring τ_k for the stability of a texture component. We leave this for future work. The value π_{min} is critical and should be larger than the fraction of an image that can be labelled foreground. Thus, this also tells us that the model will likely fail for very large targets or multiple targets covering a large portion of the image.

3.3.7 High Dimensionality

In the work of Ojala and Pietikäinen [59], the original 3×3 neighbourhood is used together with contrast information for the LBP features. This is referred to as the LBP/C distribution and in their paper it is a two-dimensional 256×8 histogram (8 is the number of bins for contrast information). We use the same definition but exclude the contrast component; the resulting features are then reduced from 256 to 59 histogram bins using uniform patterns. Subsequent dimensionality reduction or feature selection is not performed on this vector. Our experimental

results, and that of Ojala and Pietikäinen [59], show that the performance of the LBP operator is not hindered by its high dimensionality for texture characterization.

One may consider the popular Principal Components Analysis (PCA) for dimensionality reduction [64]. PCA is a method for multivariate data analysis that transforms a set of features to a set of linearly uncorrelated features. It is an orthogonal transformation that transforms the data into a new coordinate system such that the axes of this coordinate system point along the largest variance directions in the data. The PCA transformation matrix contains the eigenvectors of the covariance matrix; the eigenvectors are ordered in descending order of their eigenvalues. One generally selects m eigenvectors that account for a fraction of the variance of the data (e.g. 95% of the data).

There are some challenging aspects when using a PCA transformation:

- The full covariance matrix must be computed followed by computation of the eigenbasis. Alternatively, one may apply the Singular Value Decomposition (SVD) directly to the data matrix to compute the eigenvectors [65].
- Online update of the eigenvectors in the presence of outliers is required for the background texture model. This is not trivial.

The PCA transformation is left for future exploration and we point the interested reader to the work of Ross et al. for a method that incrementally updates a subspace model [66]. Robust PCA is also necessary in the presence of outliers.

In contrast to the avenue of dimensionality reduction, there are several works that employ textons for texture description. Interestingly, a majority of these works use k-means clustering in Euclidean space to partition the data-space into non-uniform bins. These bins characterize textons — combinations of features — that characterize the texture in an image. The texton frequencies are accumulated, locally or globally, to be used as a texture cue for computer vision tasks. For example, Georgescu et al. do mean shift clustering in high dimensions for texture classification [67]. Texton histograms of up to 560 bins are created. Similarities between these histograms are measured using D_{CHIS}. Note that the LBP descriptor is a histogram of texton frequencies in an image or region; it does not require k-means clustering to partition the feature space because the binary pattern itself indexes directly into a histogram bin.

3.4 Conclusion

This chapter presented a region-based approach for modelling the textural characteristics of the ocean. The global appearance of the ocean is described using a model of K components

where each component describes a homogeneous texture region of the ocean. The low level texture features are LBP. The global model has two variations, the GMM and the STM, that can be updated online in an efficient manner. Particular heuristics are introduced to deactivate components that are likely foreground class. Crucial to the background model construction is the method of initialization of the K components, as the feature vectors are high dimensional and distributional in nature. An information-theoretic feature clustering algorithm is used; this is complemented with the k-means++ seeding to improve the clustering convergence. In the next chapter, several experimental results and comparisons show the feasibility of the proposed methods for long range maritime surveillance.

Chapter 4

Background Modelling Experiments

4.1 Introduction

This chapter reports on several experiments using the GMM and STM texture models. Both qualitative and quantitative results are presented. The experiments demonstrate the merits of the proposed background model and they are divided as follows:

- **Texture model initialization:** The results of data clustering for initializing the texture models are shown for a set of Brodatz image mosaics and real-world images. Two of the Brodatz mosaics also contain a maritime texture sample from one of the video sequences. These experiments were designed to show the discriminating characteristics of the LBP descriptor and the effectiveness of the clustering algorithm for initializing the model.
- **Texture saliency:** In this experiment, the texture models are used to compute a simple global saliency measure such that targets in the image are the most salient and ocean pixels are the least salient.
- **Horizon detection:** A typical maritime scene contains a horizon line in the image. Horizon line detection is a useful function for detecting maritime targets because the line can be used to demarcate the ocean and reduce false positives. The LBP descriptor is used to calculate a simple change in texture gradient for finding the horizon line.
- **Detection:** The performance of the GMM and STM for background modelling and target detection is analyzed using all five tests sequences. Horizon detection is switched on for the two sequences NamacuraYacht, and NamacuraRough.

- **Comparison to existing methods:** We compare the results on Boats1 and Boats2 to existing methods for background modelling. The results presented by Chan et al. are used as a benchmark [44].

First, some aspects of the model implementation are described. Then the datasets that were used for the experiments are discussed, followed by a summary of the experimental setup. Thereafter, all the experiments and their results are explained in detail. Lastly, the chapter ends with the conclusions.

4.2 Implementation

The background model was implemented in C++ using the Linux operating system. The software made use of a combination of Central Processing Unit (CPU) and GPU processing to achieve 6.0 frames-per-second for the GMM and 8.5 frames-per-second for the STM after model initialization. The model initialization is computationally intensive and not real-time. This implementation was not fully optimized and it is expected that the processing frame rate will increase with a more robust and clean implementation. The clustering method was implemented on the CPU and it would benefit from a GPU implementation in the future. A lookup table was created for mapping an LBP feature to a uniform texture feature

A regular super pixel representation was used to achieve the fast processing rate. The image was divided into evenly spaced cells and the centres of these cells were used for the feature extraction process. Approximately 15 000 super pixels were processed. The super pixel representation reduces the memory requirements for the application and it also provides a way to reduce redundancy in the data due to neighbouring pixels having similar texture properties. If boundary-preserving super pixels are desired, one may consider the Simple Linear Iterative Clustering (SLIC) technique [68]. The SLIC super pixel construction may also help reduce the boundary effects along the horizon line in maritime videos that have a horizon. For scenes with rigid structure, the work of Chang et al. discusses temporal super pixels for video representation [69].

4.3 Datasets

During the course of this work, it was learnt that a number of the datasets in the literature:

- Are not available to the public as they are part of classified projects.
- Have very limited or no ground truth available.

- Are of very low resolution and have poor quality in general.
- Are captured by static cameras.

This created a set of difficult circumstances especially in terms of comparison of our methods to existing works.

The test data consists of five video sequences, with a total of 2183 frames, from the CSIR[1] and the CUHK[2]:

- The data supplied by the CSIR was acquired with a Prosilica GC1380 video camera using a 300mm Nikkor f/4D IF-ED lens mounted on a pan-tilt unit. The GC1380 has excellent NIR response for long range surveillance. Under most scenarios, the lens is focused at infinity as potential targets are far away from the camera system. Thus, the depth of field is not significantly shallow. There are 3 videos in this set and they were manually annotated with target bounding boxes for ground truth. These sequences are referred to as Rhib, NamacuraYacht and NamacuraRough.
- Two video sequences were obtained from the CUHK and we used them to compare our results to the work presented by Chan et al. [44]. These videos were doubled in size, from 360×200 to 720×400, using cubic interpolation to provide sufficient samples for the region descriptors in our texture model. Low resolution (180×100) foreground masks were available for the ground truth. The video sequences in this set were captured with a medium field of view static camera and were the only publicly available datasets with ground truth that have comparative results in the literature. These sequences are named Boats1 and Boats2.

The datasets are summarized in Table 4.1, and Figure 4.1 shows sample frames from the datasets. A description of the datasets follows:

- **Rhib**: This test video shows a Rigid Hull Inflatable Boat (RHIB) at the center of the frame. It is undergoing 3-Dimensional (3D) motion and can be seen rotating slightly and translating, horizontally and vertically, as it moves in the water. Sections of the target contain specular reflection owing to the fact that the video was acquired on a sunny day. There is no visible water disturbance caused by the target motion but a large number of white caps can be seen moving across the frame. The long focal length lens creates a large perspective effect as the foreground is sharp and the background is blurred. In this video the camera pans to the right; the water motion is fast and moving in the opposite direction. The video also contains significant camera shake.

[1] `http://prism.csir.co.za/`
[2] `http://visal.cs.cityu.edu.hk/downloads/`

- **NamacuraYacht:** Two vessels are observed in this video — a Namacura patrol boat and a larger fishing boat. The data was captured late evening and the camera has small amounts of motion. The scene also contains seagulls flying around. A horizon is present and the fishing boat is close to the horizon line in the video. The Namacura is difficult to see when it dips into the ocean. Both vessels stay in approximately the same position and the motion they undergo is caused by the movement of the ocean beneath them. Disturbances in the water can be seen near the two vessels. The lens creates a visibly large perspective effect as in the previous scene.

- **NamacuraRough:** In this scene, the Namacura is at the center of the video and moving to the right; the camera pans and follows the target. At about 70% into the video the target turns and moves to the left, and the camera pans in the opposite direction to follow it. The contrast is not very high in the video and the background, near the horizon, is very hazy. There is significant camera shake because the video was captured on a windy day and the pan-tilt unit was not stabilized. The target undergoes varying amounts of occlusion as it moves through the water. In some cases, a very small fraction of the object is visible as it dips into a sea swell. The water motion is quite fast and there are several white caps on the ocean moving across the frame. The target itself exhibits low textural details and there is sometimes a long wake visible resulting from its movement through the water.

- **Boats1:** This scene is captured with a static camera and contains two targets. There is uneven lighting on the ocean surface caused by the presence of glint on the left third of the frame. Firstly, a small boat enters the frame from the top right a few seconds into the video. It moves in a straight line to the left of the frame. Secondly, a yacht enters the frame from the left a third of the way into the video, and moves at high speed to the right of the frame causing a long wake to appear and stay on to the end of the video segment. Both targets exit the frame by the start of the last third of the scene. The perspective effect is also present in this video.

- **Boats2:** Boats2 shows a jet ski entering the scene above, after just a few seconds, when Boats1 ends. The long wake from Boats1 is still present and gradually disappears. The jet ski moves fast, from the left to the right of the frame, and creates a long wake with lots of water motion in the foreground that is present to the end of the video segment. The target exits the scene after one third of the video has elapsed. Like the previous video, the ocean surface has uneven lighting and the video is captured with a static camera.

In the Boats1 and Boats2 sequences, the video is empty of targets for a short while before they enter the scene.

(a) Rhib.

(b) NamacuraYacht.

(c) NamacuraRough.

(d) Boats1.

(e) Boats2.

Figure 4.1: Sample frames from the test sequences.

Table 4.1: Summary of datasets containing a total of 2183 frames.

Name	Resolution	#Frames	Duration (s)	Source
Rhib	1360x1024	126	6	CSIR
NamacuraYacht	1360x1024	401	20	CSIR
NamacuraRough	1360x1024	1056	52	CSIR
Boats1	720x400	300	12	CUHK
Boats2	720x400	300	12	CUHK

4.4 Experimental Setup

The performance of the maritime target detection methods is analysed using ROC curves. The ROC curve is a plot of the performance of a binary classifier as one varies the classification threshold. The plot shows the FPR versus the TPR as the discrimination threshold is varied. The TPR is the fraction of positives that are correctly labelled. The FPR is the fraction of negative samples labelled positive. The Area Under the ROC Curve (AUC) provides a measure of the accuracy of a binary classifier. We use the ROC curve rather than a precision-recall curve because both true positives and true negatives are important in our problem. The ROC curve tells us how well the model describes the ocean class and how well it classifies outliers.

For each test sequence, we present the AUC of the ROC curve and an analysis of the system at 50% TPR. The following must be noted:

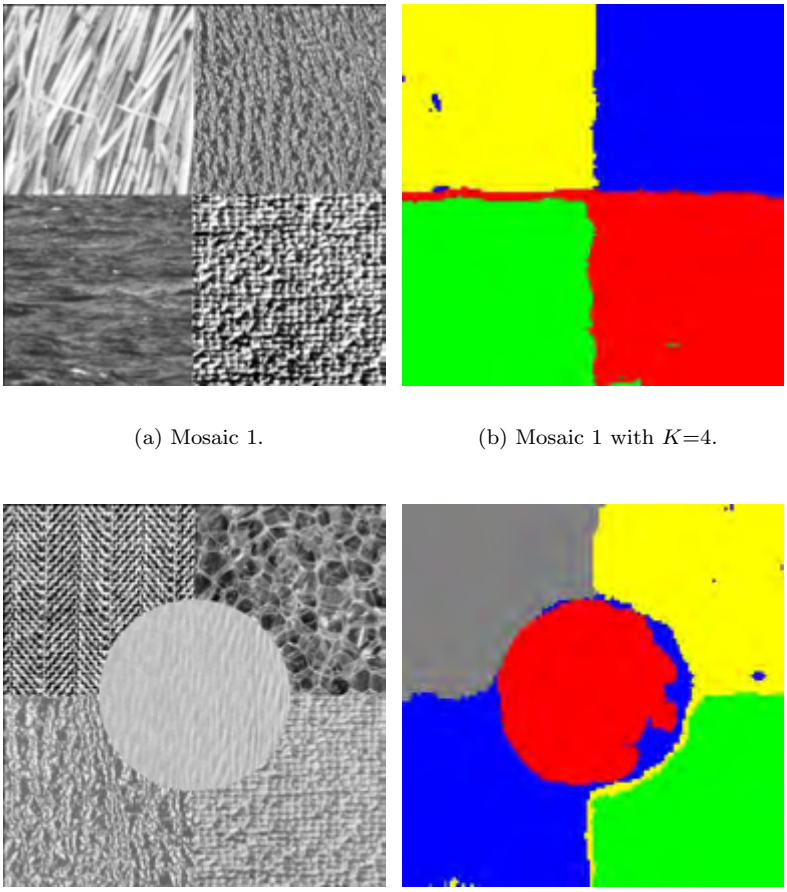
- **CSIR Test Sequences:** The ROC curves for these sequences will show some under-performance because the bounding boxes in the ground truth contain both positive and negative samples. An example of this is shown in Figure 4.2 for the Rhib video — although most of the target is correctly labelled, the TPR rate that is actually calculated will be approximately 0.6 and not close to 1.0. This can also be seen in the ROC curve for the Rhib sequence (Figures 4.9(a) and 4.10(a)) — the detector has great performance up until a TPR of approximately 0.7 after which the FPR starts to increases sharply. There is a similar trend in the other sequences as well. One may use the bounding box overlap to compute performance measures but this becomes problematic when the classification threshold is varied because the labelling gets noisy. Thus, we analyze the performance of the texture models at 0.5 TPR as this provides an accurate assessment. We also verified some of the outputs qualitatively.
- **CUHK Test Sequences:** These sequences contained low resolution ground truth masks. Their ROC curves provide an accurate measure of the performance of the background models.

The following parameters were used for the detection experiments and most of them were set empirically:

Figure 4.2: Bounding box and effect on TPR.

- Number of model components: $K = 4$. The value of K should be small enough that it describes the majority of the texture elements in the scene. This value worked well for the test cases. Adaptive selection of the number of components was not used because it was computationally intensive and it produced mixed results on our datasets.
- Minimum texture component weight: $\pi_{min} = 0.075$. This parameter is used to reject small clusters that may correspond to targets; it also ensures that the selected compouents model the majority of data. This value was selected based on the typical maximum target size in a scene.
- Texture window size: 32×32 pixels. This window size provided sufficient samples for the LBP descriptors without being too large such that target detection was compromised.
- Learning rate: $\alpha = 0.05$. The value of the learning rate determines how fast the models adapt to new data and this value corresponded, roughly, to the inverse of the video frame rate.
- The weighting factor γ was varied from 1.0 to 4.0 to generate the ROC curves for object detection.
- The regularization parameter for the covariance matrices of the GMM was set to 10^{-8} (Equation 3.15). It was found to be stable in practice.
- The image was divided into a regular grid of approximately 15 000 super pixels. We used the middle of the super pixel as the centre of the texture window.

The sections on the various experiments begin next.



(a) Mosaic 1. (b) Mosaic 1 with K=4.

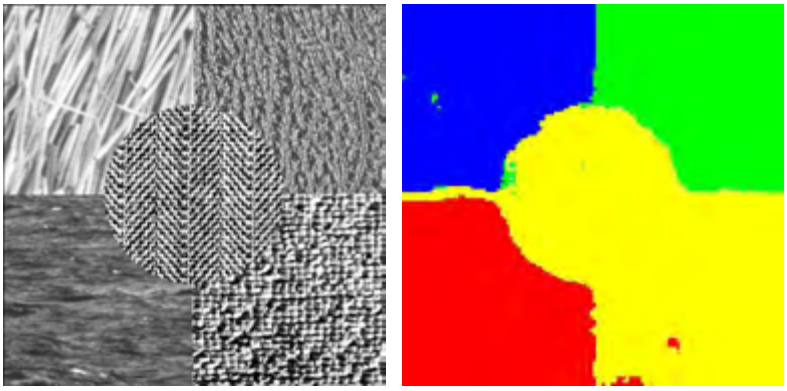
(c) Mosaic 2. (d) Mosaic 2 with K=5.

Figure 4.3: Correct initialization of texture components.

4.5 Texture Model Initialization

The results of texture feature clustering using the method described in Section 3.3.5 are shown in this section. The textural characteristics of the image pixels are described using the LBP descriptor. Once K clusters have been created, pixels are assigned to the closest prototype; the results show the prototypes, in unique colours, that pixels have been assigned to. It is observed that the clustering method and texture descriptor are suitable for describing the textural characteristics of a scene.

Figure 4.3 shows the results of assigning the label of the closest cluster to each super-pixel without any post-processing. Post-processing operations can be used to refine the boundaries of the segments and remove the small noisy regions. Visually, the clustering algorithm initializes



(a) Mosaic 3. (b) Mosaic 3 with K=5.

Figure 4.4: Incorrect initialization of texture components.

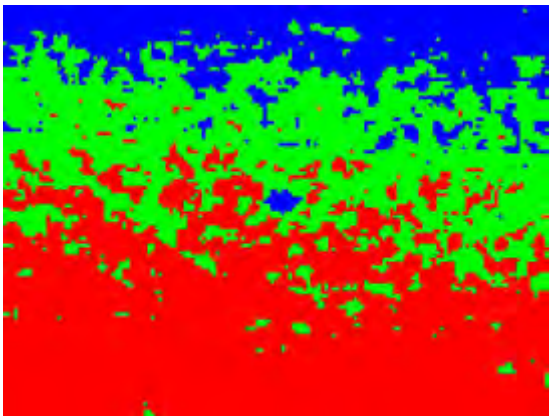
the regions correctly. In the case of Figure 4.4 the clusters are not found correctly. In this image, there are two regions that are very similar in appearance and the model assigns them to the same component. The fifth cluster is set as inactive by the initialization algorithm because its weight is very low (i.e. few samples are assigned to it). Thus, we only see four labels for K=5 in this example.

An example of real-world data clustering is shown in Figure 4.5. The image is captured with a long focal length lens. Although the ocean surface is approximately planar with the same repetitive texture patterns, the texture appearance is different at varying distances from the lens. This is evident in the way the data is clustered as shown by the colour of the labels — the perspective effect can be seen clearly and regions appear to be clustered by distance from the camera. This is a very good result because the texture components also contain implicit distance, or spatial, information. The clusters themselves also correspond to large expanses of ocean texture and is exactly what the model should describe. Although the target has a cluster label, the results presented later show that it is an outlier of the regions that the texture components describe.

4.6 Texture Saliency

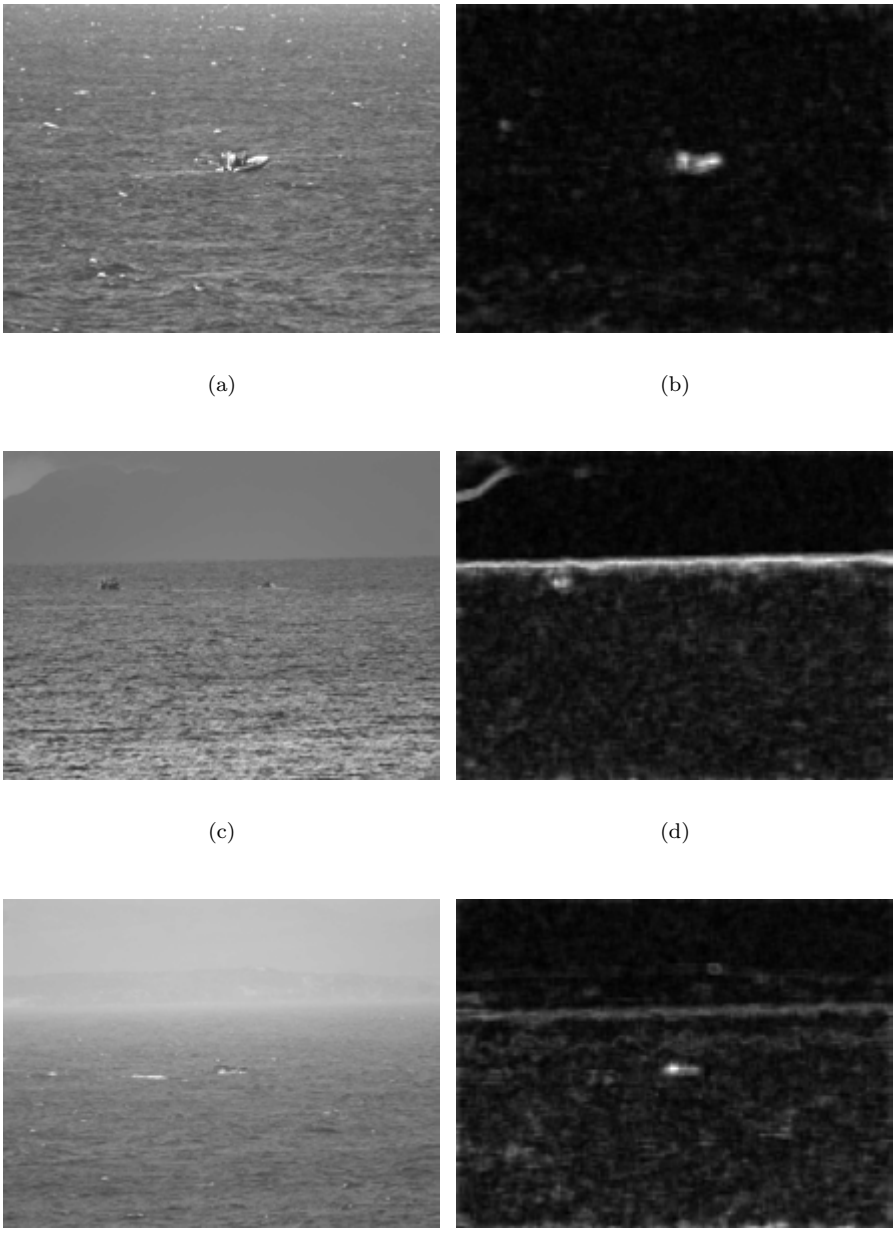
The background texture models can be used to compute a saliency map for a video frame. From a statistical point of view, the frequently observed textural elements in the scene will lie close to one of the components of the model, whereas less frequently observed patterns will be a significant distance away from the prototypes. Textural distinctiveness for salient region detection has been examined by Scharfenberger et al. using texture prototypes [70]:

(a) Rhib.



(b) Clustering on Rhib with K=3.

Figure 4.5: Clustering of texture features for model initialization in a real-world image.

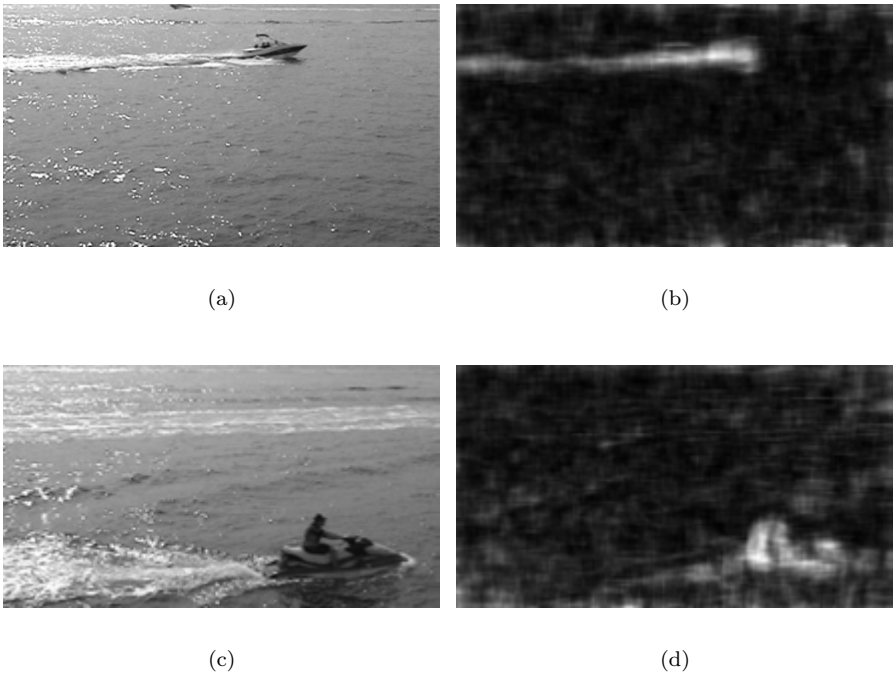


(a) (b)

(c) (d)

(e) (f)

Figure 4.6: Saliency maps computed using the STM.



(a) (b)

(c) (d)

Figure 4.7: Saliency maps for Boats1 and Boats2, computed using the STM.

- Texture features are computed and the feature dimensions are reduced using PCA.
- The reduced features are clustered using k-means to produce m texture atoms.
- The texture atoms are used to reduce the computational load for calculating pairwise saliency measures. This is combined with scene constraints and texture atom occurrence frequency to produce the saliency map.

A simple approach is suggested here using the background model:

- Compute the distance of each super-pixel to the closest texture component. In the GMM we used D_{MAH}; for the STM D_{CHIS} is used.
- Unit normalize the distances, using the minimum and maximum values, to create the saliency map.

One may also use the prior weights and pairwise measures between the prototypes for additional saliency information. Figure 4.6 and 4.7 show examples of the saliency maps that were computed using the STM. The targets of interest appear as bright blobs.

4.7 Horizon Detection

Surveillance systems operating along a coast or from up in the air, like Unmanned Aerial Vehicles (UAVs), will likely at times contain a horizon line in their field of view. The horizon line may be used for image based stabilization or for locating the sky, and ocean or land. In our case the horizon line is detected to reduce false positives that may arise in the sky or along the coastline above the water. We also observed that a sharp transition from ocean to coastline gives rise to pixels that are outliers of our model — this effect can be observed in some of the saliency maps in the previous section. This occurs because the texture windows near the horizon boundary contain LBPs from both ocean and sky, and they have extremely different appearances. We perform horizon detection to ignore detections above the ocean and near the horizon boundary.

Our method uses the LBP descriptor and line detection to detect the horizon; current methods employ colour models and sky-ground separation for locating the horizon [16, 71, 72]. The LBP descriptor offers the advantage of considering texture, or structure, as a cue for finding region boundaries. Thus, it will not be affected by factors such as poor colour saturation or low contrast. The Random Sample Consensus (RANSAC) method is used to find the horizon line [73]. Candidate points for the horizon are selected in the following way:

- Compute the vertical texture similarity

  $$dh_{ij} = D_{CHIS}(\mathbf{x}_{ij}, \mathbf{x}_{ij-1})$$

  for all super pixels, where i is the column and j is the row in the image. Position (i, j) is the centre of the super pixel. Texture information along the horizontal direction of the image plane may be incorporated if desired.

- Normalize all the computed similarity measures, using their minimum and maximum values, so that they lie in the range $[0, 1]$. Candidate horizon points are selected if

  $$\begin{aligned} dh_{ij} &> 0.5 \\ dh_{ij} &> dh_{ij+1} \\ dh_{ij} &> dh_{ij-1}. \end{aligned}$$

  These conditions ensure that all the selected points are local maxima, in the vertical direction, and that they have a minimum relative strength.

The RANSAC algorithm iterates the following:

- Randomly select a pair of points from the candidate set and compute the line passing

through them.

- Use the line equation to compute the number of candidate points that are inliers. A point is an inlier if its L1 norm from the line, along the j coordinate only, is less then 5 pixels. Count the number of inliers to compute the line score.

After 200 iterations, the line with the highest score is selected as the best fit to the point data.

A qualitative assessment of the horizon detection was done using the NamacuraYacht and NamacuraRough videos. Some results are shown in Figure 4.8. Figures 4.8(a) and 4.8(b) show instances of correct horizon direction. It is worth noting that Figure 4.8(a) is a rather difficult scenario because the image has low contrast along the coastline and the texture change from the ocean to the land and to the sky is very gradual. In this case, the orientation of the detected line is correct but the location is lower than one would it expect it to be. However, we also see that the ocean region above the line has textural characteristics similar to the coastline. This explains the location of the detected line.

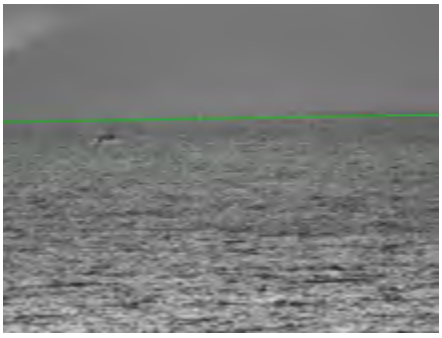
Figure 4.8(b) highlights the slight inaccuracy of our method in detecting the horizon line location. There are two reasons for this. Firstly, the super pixel representation implies that we lose pixel level accuracy when detecting boundaries. Secondly, the texture window size (32×32) also introduces ambiguity at boundaries. Irrespective of these shortcomings, the method was seen to work well on our two test sequences. An instance of an incorrect detection is shown in Figure 4.8(c). The detected line is skew and it is likely caused by insufficient inliers in the candidate point set.

4.8 Foreground Detection Results

This section presents the main performance analysis for target detection using the background texture models. For each video sequence and texture model, five random trials were run at the different settings and we selected the best performing results in terms of AUC for each test sequence. Horizon detection was switched on for NamacuraYacht and NamacuraRough. The ROC curves for the GMM and STM are shown in Figures 4.9 and 4.10 respectively. A summary of the performance of the models for a TPR of 50% is shown in Table 4.2.

Our first observation is that the ROC curves for both texture models have the characteristic shape of a good binary classifier. With the exception of the NamacuraRough test sequence, the AUC of the ROC curves are all over 0.85. This implies that the texture model, as a binary classifier, performs very well and is much better than random guessing. The NamacuraRough video is a difficult test case and we provide a detailed discussion on this sequence later in this section.

(a) Correct horizon detection.



(b) Correct horizon detection.

(c) Incorrect detection.

Figure 4.8: Horizon detection.

Table 4.2: AUC and FPR at 50% TPR.

Sequence	GMM(AUC)	STM(AUC)	GMM(FPR)	STM(FPR)
Rhib	0.8827	0.8718	0.0033	0.0044
NamacuraYacht	0.9208	0.9088	0.0083	0.0159
NamacuraRough	0.7722	0.7882	0.1056	0.0843
Boats1	0.9590	0.9612	0.0147	0.0052
Boats2	0.9666	0.9733	0.0071	0.0051

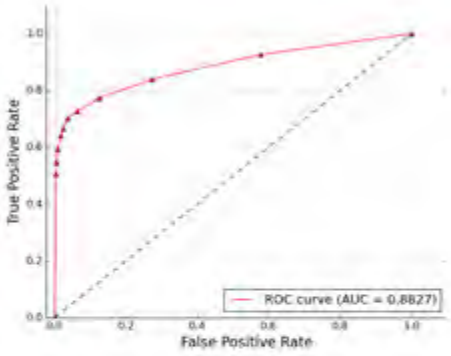
The two models, GMM and STM, have comparable performance. None of them have a clear performance advantage over the other in terms of TPR and FPR. The GMM performs better than the STM for the Rhib and NamacuraYacht sequences; the STM has better performance for the other sequences. With the exception of the NamacuraRough sequence, both texture models have a FPR of less than 2% for a 50% TPR. The STM has a better running time than the GMM and this is the only clear advantage that we see. As a result of both texture models performing similarly, a general discussion of some of the results for each test video is now presented. Several results for foreground detection are shown in Figures 4.11, 4.12 and 4.13.

4.8.1 Rhib Sequence

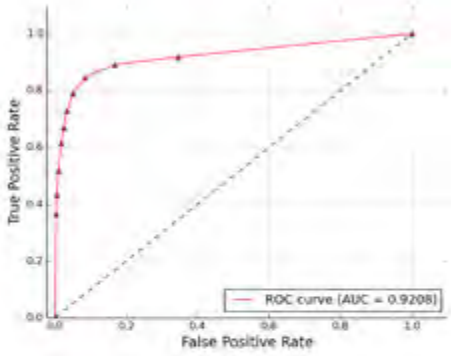
Detections for the Rhib sequence are shown in Figures 4.11(b) and 4.12(b). This dataset contains large amounts of water motion and there are several white caps appearing and disappearing as they move through the water. The background models perform very well on this scene in spite of the camera motion, target motion and changing appearance of the ocean. Compared to the other two sequences in the CSIR dataset, this one has a target of reasonable size with distinguishing details that are not diminished had the target been further away from the camera.

The few false positives are caused by some of the white caps on the ocean. Note that these false positives are large in size whereas some of the other white caps on the ocean, that are labelled negative, are smaller or narrower. This demonstrate that the LBP descriptor offers a level of robustness to white caps that are observed on the ocean. For small white caps the ocean patterns dominate the histogram descriptor and the influence of the white cap patterns on the background model is not drastic.

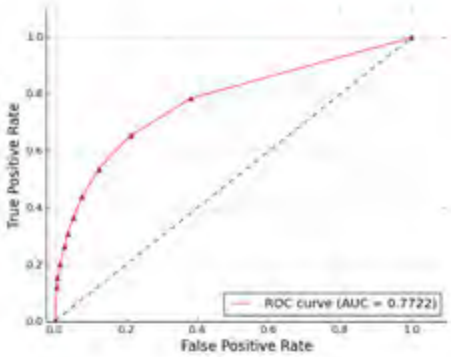
In this particular scene, the statistical properties of the ocean are ideal for the background model and the heuristics that were proposed work well.



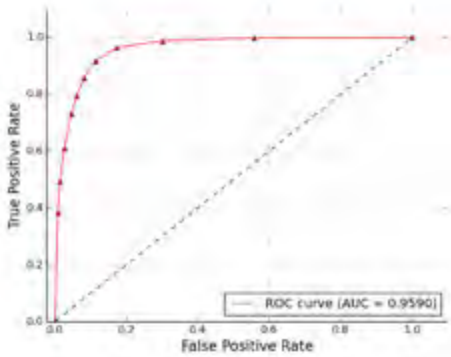
(a) Rhib.



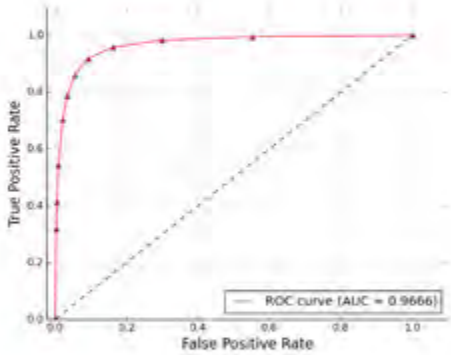
(b) NamacuraYacht.



(c) NamacuraRough.

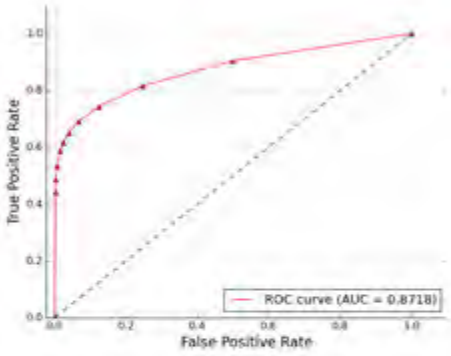


(d) Boats1.

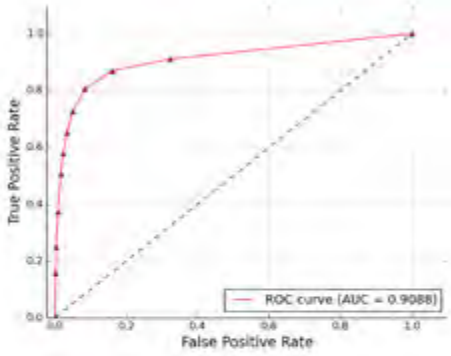


(e) Boats2.

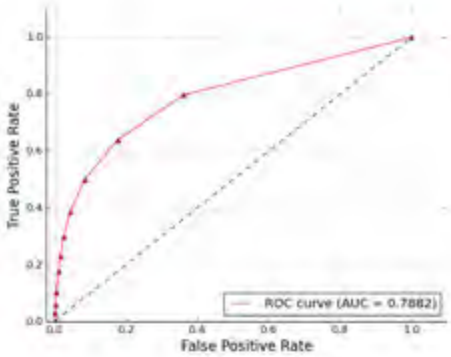
Figure 4.9: ROC for GMM.



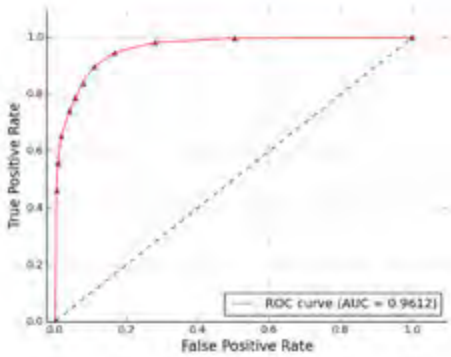
(a) Rhib.



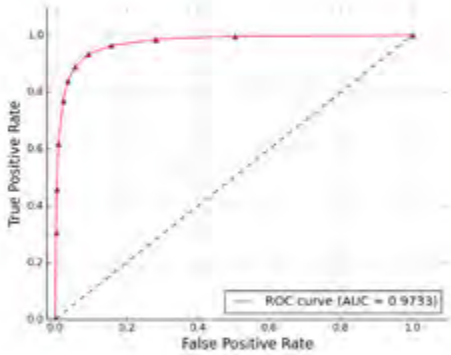
(b) NamacuraYacht.



(c) NamacuraRough.



(d) Boats1.



(e) Boats2.

Figure 4.10: ROC for STM.

4.8.2 NamacuraYacht Sequence

The texture models performed reasonably well on the NamacuraYacht sequence. However, we observed two challenges in this video:

- Firstly, the horizon line is a natural texture boundary and LBPs on or near this boundary have mixed texture characteristics. Thus, there is ambiguity when attempting to classify super pixels close to the horizon. Super pixels close to the horizon line are ignored once the horizon line is detected. However, a thin line of misclassified pixels is still present as shown in Figures 4.11(d) and 4.12(d). One may also vary the detection threshold to reduce the false positives along the horizon at the expense of also missing potential targets. In the next section, a graph cut solution is proposed for removing noisy false foreground detections such as those near the horizon.
- Secondly, the Namacura boat is small in size and its textural patterns are at times indistinct, making it difficult to detect. An example of a small set of detections on the Namacura vessel is shown in Figure 4.11(d); in Figure 4.12(d) the Namacura is not detected and it is also quite difficult to discern with the human eye.

A qualitative assessment of the results for this sequence showed that the small Namacura boat was in general difficult to detect. Additional image features, such as grey level intensity and contrast, may provide more discriminating characteristics for this type of scenario.

There is one important aspect that must be clarified. In scenes containing a horizon line, the model is initialized using data from the whole image frame. Thus, for the video sequences with a horizon present, one or two components of the model will describe the sky and coastline if they are present. It was found that this did not affect the detection results. It also provided a more meaningful description of the natural scene due to the sky and coast regions in our test sequences being homogeneous. Note that this approach is not recommended if the region above the horizon is inhomogeneous texture. One could perform the horizon detection first and then initialize and update the model with measurements below the horizon line.

4.8.3 NamacuraRough Sequence

The NamacuraRough video sequence is arguably the most difficult video in the test sets. Like the NamacuraYacht test case there is a horizon line present, with parts of the sky and coastline being homogeneous texture. There are several partial occlusions of the target with varying degrees of severity throughout the sequence.

The model was initialized using the first video frame and thereafter detections above the horizon line were ignored. Some detection results are shown in Figures 4.11(f) and 4.12(f). In this test

sequence false positives are caused by white caps, the target's wake, or parts of the ocean that have a texture appearance different from the general appearance of the ocean. Most of the false positives are small in size and are scattered sparsely on the ocean. As with the previous test sequence, Figure 4.12(f) shows some false positives close to the horizon that arise from the texture boundary between water and coastline.

In Figure 4.11(f), the detection threshold was adjusted so that the number of false positives was low. Here it can be seen that just the cabin of the Namacura is detected. The detection threshold used in Figure 4.12(f) was adjusted so that more pixels of the target were detected; as expected, the number of false positives also increased. In spite of the severe camera shake, low image contrast, target occlusions and extreme dynamics of the water appearance, the texture models performed very well and the target was detected. It was also noticed that parts of the target that are not detected appear to be textureless.

The low ROC AUC (0.77 for the GMM and 0.79 for the STM) is caused by the large number of partial occlusions of the target in the video. Although a bounding box for the target was available for each video frame the percentage of target that was visible was quite small, making foreground detection difficult. Thus, we expect the ROC curve to appear slightly worse than for the other videos in the CSIR dataset.

4.8.4 Boats1 and Boats2 Sequences

The Boats1 and Boats2 sequences contain the ideal scenarios for testing the texture models. The videos are free of targets for a short while before they enter the scene which allows initialization and update of the models for a number of frames without the presence of object outliers. The absence of a horizon also implies that the models deal directly with only ocean data. Some results using the GMM are shown; the STM performed similarly and this can be observed in the ROC curves.

Figure 4.13(b) shows detection results on the Boats1 sequences using the GMM. The hull of the large boat is detected but smaller details, such as the windscreen and peoples' heads, are missed. The small target at the top of the frame is also missed due to its size. It is detected if the detection threshold is varied but an increase in false positives occurs as expected. A portion of the wake is detected as a potential target and on closer inspection it can be seen that this part of the wake contains a black streak surrounded by white foam, giving it an appearance similar to the hull of the boat. The texture model is able to account for the uneven lighting on the ocean and it is not adversely affected by the specular reflection, shadows or most of the wakes caused by the moving targets.

A detection result for the Boats2 sequence is shown in Figure 4.13(d). The texture models performed very well on this sequence. It is reasoned that the size of the target was a contribut-

ing factor as it provided a good set of samples and statistics for the outlier descriptors. The model is unaffected by the large wake that the target creates in the foreground. The parts of the target that are not detected are textureless.

4.8.5 Robustness

A sequence of frames from the Boats2 sequence demonstrating the robustness of the texture models, in this case the GMM, is shown from Figures 4.14(a) to 4.14(i). The detection is largely unaffected by the wake of the target in the foreground that appears when the target moves across the frame.

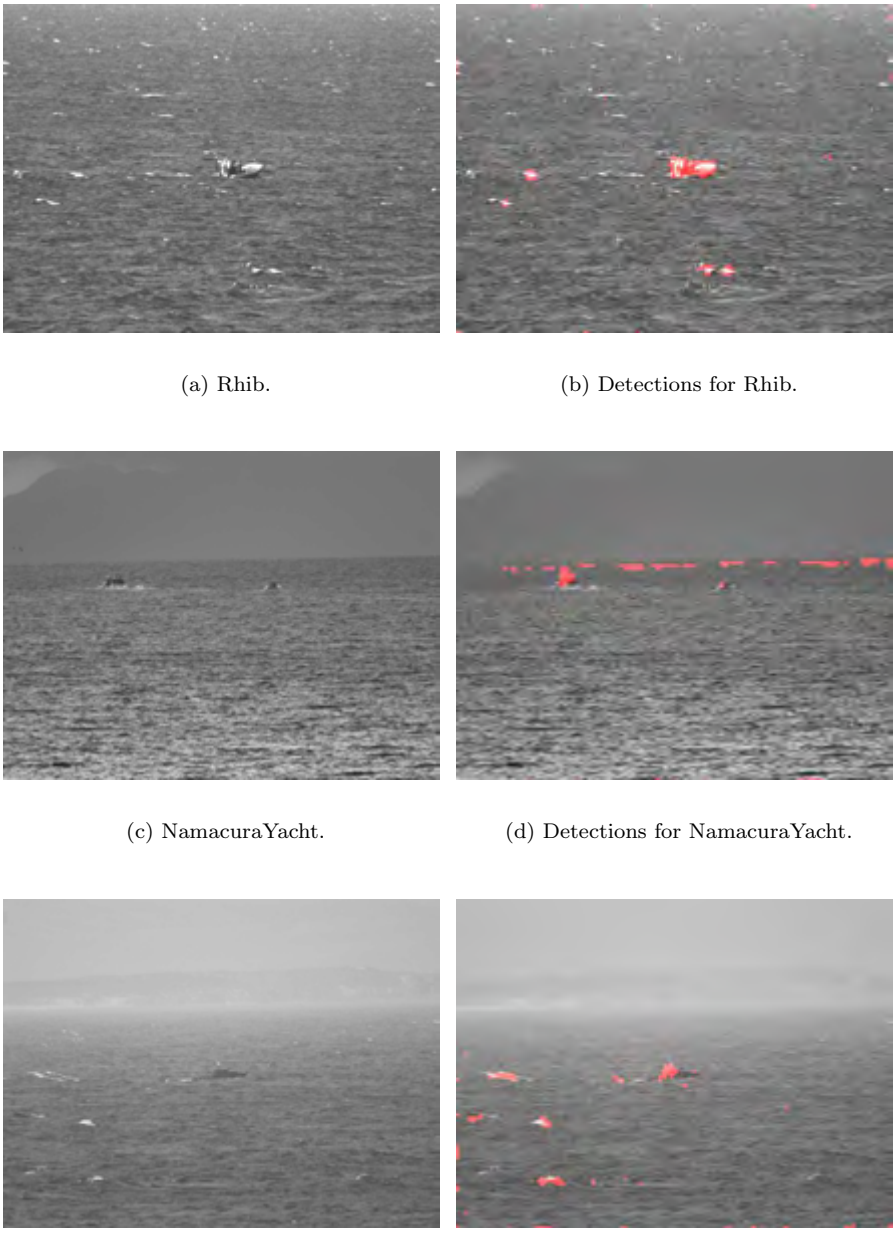
4.8.6 False Positives and False Negatives

Examples of false positives and false negatives are shown in Figure 4.15. Some of these problems can be corrected by tuning the detection threshold. However, it is important to mention them so that one can understand the limitations of the methods.

Figure 4.15(a) shows examples of some of the false positives that can be created by the detection methods. The horizon detection is switched off for clarity. In this example there are several issues to report:

- The texture boundary between water and coastline creates outliers that are labelled positive.
- Some of the elements on the coastline have a textural property quite different from the land and sky, and they are labelled positive.
- There are white caps on the ocean that create false positives.
- Some false positives are created by tiny waves with multiple ridge like patterns that are different from the general appearance of the ocean.

In the NamacuraYacht scene, shown in Figure 4.15(b), an example of a false negative is visible in that the Namacura yacht is undetected. Its visibility is extremely low and although a blob can be seen it looks very much like the ocean texture appearance. The false positives along the horizon are also visible. In Figure 4.15(c), the Rhib sequence shows a large cluster of false positives resulting from the large white caps in the foreground. The false positives arising from the wake of a moving vessel are shown in Figure 4.15(d). A false negative is also visible in the smaller boat at the top of the frame that is not detected. We point out that the false negatives and false positives that the detector generates are consistent with the works of others [9, 28].



(a) Rhib.

(b) Detections for Rhib.

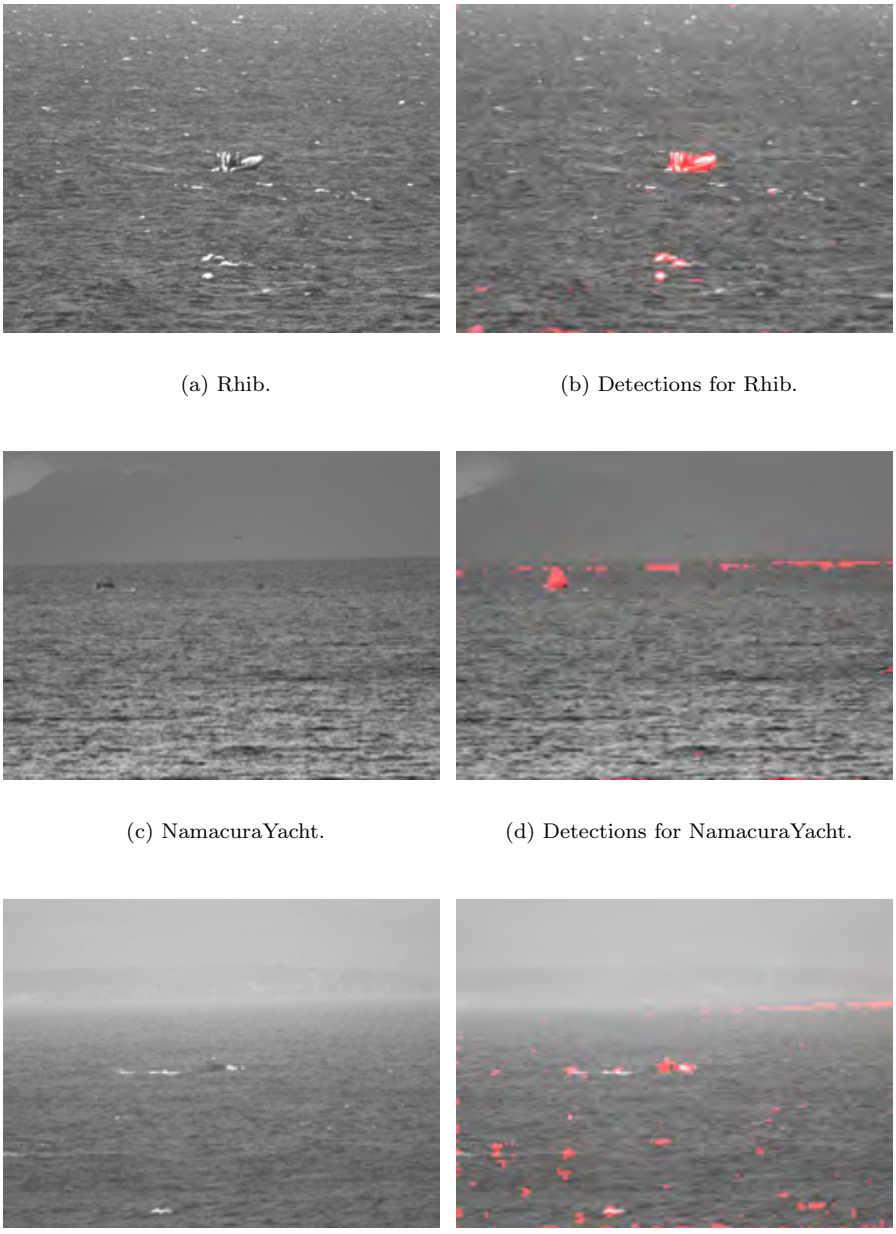
(c) NamacuraYacht.

(d) Detections for NamacuraYacht.

(e) NamacuraRough.

(f) Detections for NamacuraRough.

Figure 4.11: Detections using the GMM.



(a) Rhib.

(b) Detections for Rhib.

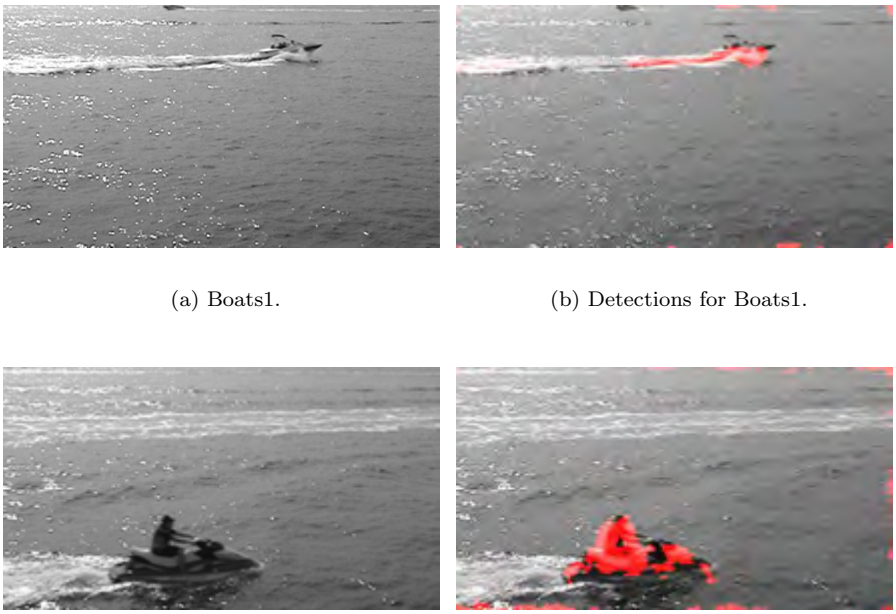
(c) NamacuraYacht.

(d) Detections for NamacuraYacht.

(e) NamacuraRough.

(f) Detections for NamacuraRough.

Figure 4.12: Detections using the STM.



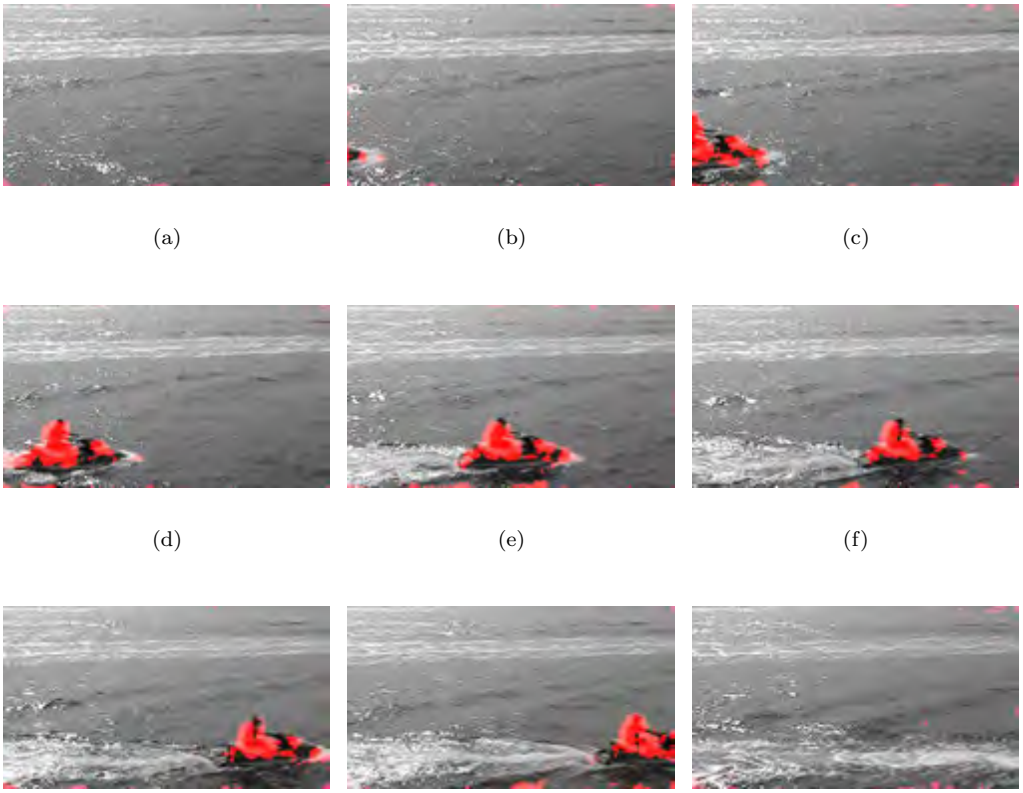
(a) Boats1.

(b) Detections for Boats1.

(c) Boats2.

(d) Detections for Boats2.

Figure 4.13: Detections using the GMM for Boats1 and Boats2. Similar results were achieved for the STM.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.14: Robustness of the model on the Boats2 video using the GMM texture model: (a) – (i) show a sequence of frames of a jet ski entering and leaving the scene. The wake that the water craft creates is not falsely detected.

There are several reasons why small targets are not detected:

- A small target may contain insufficient discriminating texture samples. It is likely the texture window is too large for a small target and its feature histogram is dominated by the ocean LBPs.
- The small target has poor textural characteristics and its LBPs are similar to the ocean.
- Due to the super pixel representation, a small target may be missed or split across texture windows. If it is split across texture windows, there may be too few discriminating samples for detection.

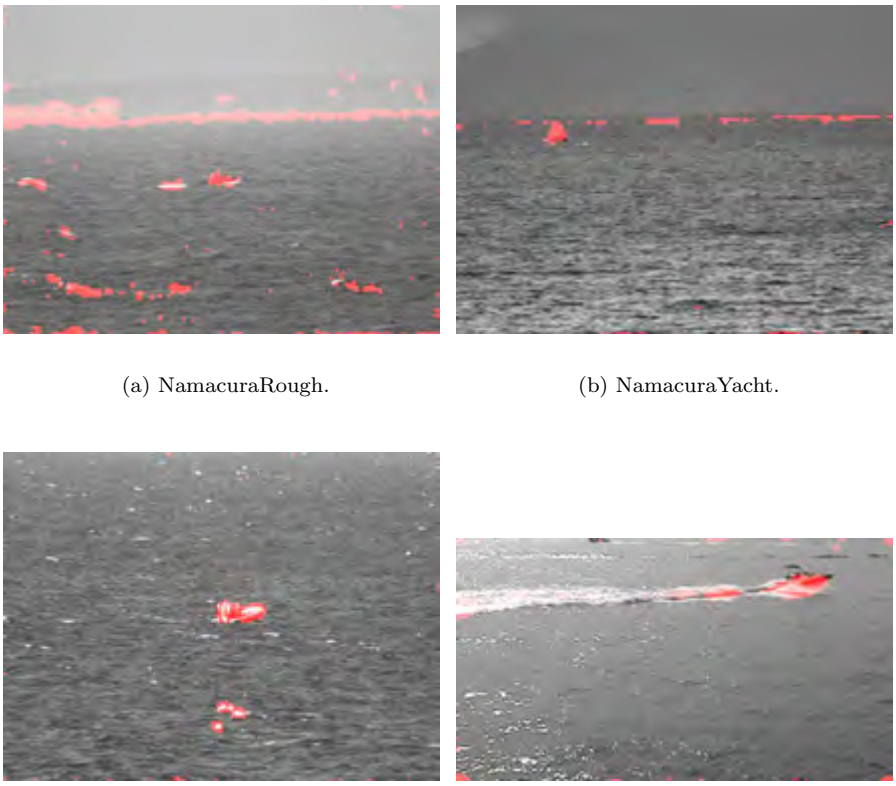
To summarize, the primary causes of false positives and false negatives are:

- Small targets.
- White caps on the ocean.
- The wake caused by a moving target.
- The texture boundary at the horizon line.
- A lack of textural/salient details on the target for separation from the background.

4.9 Comparison to Existing Methods

We compared our texture models to some of the comparative results presented by Chan et al. that use the Boats1 and Boats2 sequences [44]. In their paper, the various models rely on the static camera assumption. Thus, these methods immediately have an advantage because they have registered pixels, and model the spatial and temporal characteristic of the patterns. The following methods are compared to the proposed GMM and STM:

- The Stauffer-Grimson (SG) method [37] with $\alpha = 0.01$.
- A simple PCA background model with 10 components and 7×7 observation patches.
- The method of Zhong and Sclaroff that uses a Kalman filter for segmenting foreground objects from dynamic textured background [43].
- The dynamic texture mixture models, Dynamic Texture Mixture with 1 component (DTM1) and Dynamic Texture Mixture with 3 components (DTM3), of Chan et al. with their parameters $\alpha = 0.16$, $\beta = 0.08$ and a covariance regularization parameter of $\sigma = 10$ [44].



(a) NamacuraRough. (b) NamacuraYacht.

(c) Rhib. (d) Boats1.

Figure 4.15: Examples of false detections.

The AUC and the FPR are summarized in Table 4.3 with the best performance figures shown in bold. The TPRs that we compared against, 0.90 for Boats1 and 0.55 for Boats2, were the only ones available for comparison in Chan et al. [44]. The methods that we compare against are all temporal models — they are generally pixel-wise models with parameters computed from past observations. Our results show that the region based model has comparable results to the state of the art in spite of the absence of registered pixels and a temporal component. The LBP patched based approach is robust and performs well on these test sequences. The proposed heuristics are suitable assumptions for constructing a background model for the ocean.

The DTM3 performs the best on the Boats1 video, having the highest AUC and the lowest FPR. The STM and GMM have similar performance to the other methods. For example, the GMM and STM have better AUC than the method of Zhong and Sclaroff, and SG, but they also have slightly higher FPRs compared to these two methods. In the Boats1 sequence, the high FPR of our system is caused by the wake created by the target; the low AUC is caused by the small target that is difficult to detect. The STM has the highest AUC in the Boats2 sequence followed by the GMM. Both methods have some of the lowest FPRs for Boats2, performing better than all the other methods.

Although the dynamic texture models offer the best performance in general, performing very well on both test sequences, these results show that the performance of the proposed approaches are at least comparable to the state of the art and in some cases better than those methods. We point out, once again, that our models are region based and do not require registered pixels for pixel-wise models.

4.10 Conclusion

The experimental results presented in this chapter demonstrate the effectiveness of the proposed texture models for maritime target detection. Although the model is region-based and has no temporal dynamics, the LBP feature distributions are effective for describing the ocean background and detecting potential targets. The region based approach circumvents the need to perform registration of image frames for more common pixel-wise temporal models such as mixture of Gaussians. The region-based model herein would be useful as one component of a multi-modal feature analysis approach for maritime target detection.

A qualitative analysis of the data clustering method showed that it was effective in partitioning the homogeneous texture classes in a set of Brodatz mosaics. This partitioning is crucial to the initialization of the texture models and it also demonstrates that the LBP feature descriptors capture the discriminating characteristics of visual texture. On a single real world image, captured with a long focal length lens, the clustered texture classes corresponded to regions ordered by distance from the camera. This result is useful for object detection and tracking in

Table 4.3: Comparison to state of the art.

Sequence		SG	PCA	[43]	DTM1	DTM3	GMM	STM
Boats1	AUC	0.9516	0.9698	0.9493	0.9886	**0.9910**	0.9590	0.9612
Boats1	FPR (TPR=0.90)	0.1030	0.0865	0.1067	0.0120	**0.0061**	0.1040	0.1122
Boats2	AUC	0.8952	0.8136	0.7966	0.8916	0.9266	0.9666	**0.9733**
Boats2	FPR (TPR=0.55)	0.0261	0.1527	0.1495	0.0318	0.0088	0.0091	**0.0066**

long range surveillance of maritime scenes. The texture models were also shown to summarize the most common and frequent texture appearances, in a maritime image, corresponding to the ocean class; less common appearances that are typical of maritime targets against a regular texture background were shown to be more salient than the background class. Thus, the texture models describe the ocean appearance in a meaningful way. Horizon detection was achieved using the LBP descriptors to detect changes across texture boundaries. An advantage of this approach is that the boundaries are found based only on texture and structure rather than intensity.

The performance of the foreground detection was demonstrated using ROC curves and both models were shown to have similar performance. In almost all test cases, the AUC was greater than 0.85 with a FPR of less than 2% at 50% TPR. The lowest AUC of 0.7722 was reported on the NamacuraRough sequence using the GMM. The models were compared to existing state of the art methods using two publicly available maritime test sequences. The proposed models offer similar levels of performance compared to some of the state of the art; on the Boats2 sequence the STM and GMM performed the best. The dynamic texture models provided the best performance overall on two test sequences, having low FPRs and high AUCs, but they are still pixel-wise models that will not work on video captured from moving cameras. Thus, in this regard our models are favourable.

The texture models were shown to perform poorly across texture boundaries and textureless regions. However, these are known problems in texture analysis and can be addressed in future work. The common false positives, such as white caps, the wake of a target and specular reflection, are typical in maritime scenes. As with most methods using regional descriptors, small targets are difficult to characterize and detect. Further, targets with patterns similar to the texture background will not be detected. The detector is computationally efficient and runs at 8.5 and 6.0 frames-per-second for STM and GMM respectively with 15000 regular super pixels.

The results presented in this chapter are very promising for maritime target detection. A new way of viewing the problem has been presented. Rather than attempting to implement pixel wise models we have shown experimentally that a region based model works just as well as it can exploit the regional texture patterns to create a description of the ocean.

Chapter 5

Graph Cut

5.1 Introduction

Variations in image data cause poor spatial coherence in foreground detection. The sources of these variations include image appearance changes, sensor noise and environmental effects. Some examples are:

- White caps on the ocean. Although these are part of the ocean appearance, our models cannot describe all white caps.
- Texture variations on the ocean caused by the position of the sun, wind or water motion.
- The variations of pixels close to the horizon texture boundary.

Pixels that exhibit these types of variations deviate from the general appearance that our texture models describe. Given the noisy observed data, as described above, one wishes to infer the class label of a pixel where the label denotes membership to either target or background. This chapter discusses GC optimization to improve foreground detection.

Taking our cue from Boykov and Kolmogorov [74], the labelling problem can be formulated as the minimization of the energy function of the labels $\mathbf{l}$,

$$E(\mathbf{l}) = E_{data}(\mathbf{l}) + E_{smooth}(\mathbf{l}). \tag{5.1}$$

This energy function ensures that object boundaries are preserved and there is strong agreement between $\mathbf{l}$ and the observed data. In the equation above, E_{data} measures the agreement between $\mathbf{l}$ and the observed data while E_{smooth} measures the extent to which $\mathbf{l}$ is smooth. For these types of vision problems, a crucial aspect is the specification of the data and smoothing functions.

In this work, binary image labelling of the following energy function is considered for n sites:

$$E(\mathbf{l}) = \sum_{i=0}^{n-1} U_i(l_i) + \sum_{i=0}^{n-1} \sum_{j \in \mathcal{N}_i} P_{i,j}(l_i, l_j), \tag{5.2}$$

where U_i is the data term and $P_{i,j}$ the pairwise energy term that enforces smoothness in the labelling. The data term, analogous to E_{data}, is the cost of assigning class label l_i to pixel i and it is computed using the background texture models. Smoothness on the pixel grid is enforced using pairwise affinities $P_{i,j}$, analogous to E_{smooth}. This energy function can be minimized with a GC.

Graph cuts are well known and very popular in the computer vision community. In the next two sections the GC method and its relation to MRFs, from a probabilistic perspective, are discussed. Thereafter, the implementation details of the energy functions are presented. The chapter ends with a presentation of the experimental results, followed by the conclusion.

5.2 Graph Cut

Graph cuts and their application to computer vision problems have gained interest in the last decade or so through the work of Boykov et al. [74, 75]. Their contributions study GCs within the scope of vision and they also present an efficient method for fast energy minimization. In the binary image labelling problem, the global minimum for the energy function in Equation 5.2 can be found using the principle of maximum flow (or minimum cut) in a graph. It can also be extended to multi-label problems, which are not the focus here. We discuss GCs with respect to binary image labelling.

A directed graph $\mathbf{G} = (V, E)$ contains vertices V connected by edges E. Each edge in E has a non-negative capacity. For an edge connecting vertices i and j the capacity is c_{ij}. For the maximum flow problem, there are two additional special vertices called the source (s) and sink (t) that are connected to all the other nodes.

In computer vision problems the vertices often represent pixels. The edges between the vertices are the pairwise potentials. There also exists a directed edge from source to vertex and one from vertex to sink. By the Ford-Fulkerson theorem, the maximum flow in a graph is equal to the minimum cut in the graph [76]. For the minimum cut, one cuts the edge between source and vertex or the edge between vertex and sink. Both edges are never cut because this will incur a greater cost. A minimum cut is a possible labelling of the image if we say that a pixel is labelled 0 when the edge from the vertex to its sink is cut, or it is labelled 1 when the edge from source to its vertex is cut. The cut partitions the image into disjoint sets S and T such that $s \in S$ and $t \in T$. The total cost of a cut is the sum of boundary edges (p, q) such that

$p \in S$ and $q \in T$.

The maximum flow is analogous to the maximum amount of water that can flow from the source to the sink if the source was a tank of water and the edges were pipes. There are 2 main methods for solving the maximum flow problem: augmenting path [76] and push-relabel approaches [77].

5.2.1 Augmenting Path Methods

An augmenting path is an $s - t$ path with available capacity. The main component of the augmenting path approach is the residual graph G_f. It stores information about the residual capacity of the graph. This is used to detect when edges in a network are saturated. The standard iterative approach is summarized:

- Find shortest path $s - t$ along non-saturated edges of G_f.
- If a path is found, augment it by pushing the maximum possible flow d_f so that at least one edge is saturated.
- The residual capacities along the $s - t$ path in G_f are reduced by d_f while residual capacities of reverse edges are increased by d_f.

Reverse edges are used to redirect flow from an edge. Each augmentation increases the total flow from s to t. The maximum flow is reached when any $s - t$ path crosses a saturated edge in G_f. The original algorithm makes no mention of how an $s - t$ path should be found.

5.2.2 Push-relabel Techniques

The push-relabel methods maintain a notion of "preflow": the flow into a node v exceeds the flow out of v. A flow is present when the flow into v is equal to the flow out of v. A height function that determines which vertex pair is selected for a push operation is also defined.

This technique assumes that each node in the graph G has a label that reflects its height. The height determines the direction of flow. The push function pushes flow downhill from a node to one of lower height. The relabel function changes the height of a node to the minimum value such that a valid out-edge is created. The basic method is:

- All nodes are initialized with height zero.
- Each edge from the source is filled to capacity.

- Examine all nodes except s and t.
- Flow from a filled node is pushed downhill (overflowing).
- If an overflowing node is at the same level or lower than nodes it can push flow to, it is raised just enough to push flow to these nodes.
- If the sink is not reachable from an overflowing node, excess water is sent back to the source.

When the flow stops, the max-flow has been found.

5.2.3 Method of Boykov et al.

The method that we use to find the maximum flow of a graph is the well-known approach of Boykov et al. that improves the performance of the standard augmenting path method [74]. In their seminal paper, they presented a new way to speed up augmenting paths to find the maximum flow. The authors pointed out that most augmenting path techniques required a breadth-first search tree for finding an $s - t$ path. Furthermore, in most of these methods the search tree was rebuilt several times and this contributed to poor performance. They proposed the following approach:

- Two search trees are built: one from the source and one from the sink.
- These two trees are re-used and never rebuilt.

The authors stated that it was not guaranteed that the shortest path would be found using their algorithm. However, their results showed that the method outperformed the standard variants on typical vision problems.

5.3 Markov Random Fields

One may take a probabilistic perspective and look at a graph cut as the optimization of a MRF. A Markov Random Field (MRF) is an undirected graphical model of a set of random variables that have a Markov property. We consider it from the perspective of the image labelling problem where the probability of a label for each pixel is conditioned on the labels of its neighbours. The information provided by the label of a single pixel is weak; however, by preferring particular spatial configurations of the labels, an optimal labelling may be extracted. When these spatial configurations are pairwise, the MAP solution for the MRF is found using graph cut techniques.

An undirected graph $\mathbf{G} = (V, E)$ contains vertices V connected by edges E. The MRF has the following components:

- n sites corresponding to the pixels. These are the vertices $i \in V$ in the graph $\mathbf{G}$.
- A set of random variables $\mathbf{l} = \{l_0, l_1, \cdots, l_{n-1}\}$ corresponding to each site. The label of pixel i is denoted by l_i.
- A set of neighbours $\mathcal{N}_i$ at each of the sites.

A MRF must satisfy the Markov property

$$Pr(l_i|l_{V/i}) = Pr(l_i|l_{\mathcal{N}_i}), \tag{5.3}$$

which says that it is conditionally independent of all the other sites given its neighbours. The joint probability of the variables is the product

$$Pr(\mathbf{l}) = \frac{1}{Z}\prod_{i=0}^{n-1} Pr(l_i|l_{\mathcal{N}_i}), \tag{5.4}$$

where Z is the partition function. Equation 5.4 can be described by pairwise potential functions γ_{ij} such that

$$Pr(l_i|l_{\mathcal{N}_i}) = \frac{1}{Z_i}\prod_{j\in\mathcal{N}_i} \gamma_{ij}(l_i, l_j), \tag{5.5}$$

where Z_i is a partition function. Then the joint distribution is

$$Pr(\mathbf{l}) = \frac{1}{Z}\prod_{i=0}^{n-1}\prod_{j\in\mathcal{N}_i} \gamma_{ij}(l_i, l_j). \tag{5.6}$$

It is common to write the above as a cost function using the Gibbs distribution:

$$Pr(\mathbf{l}) = \frac{1}{Z}\exp\left(-\sum_{i=0}^{n-1}\sum_{j\in\mathcal{N}_i} \phi_{ij}(l_i, l_j)\right), \tag{5.7}$$

where $\phi_{ij}(\bullet) = -log(\gamma_{ij}(\bullet))$ is the cost function.

Given an observed image $\mathbf{x} = (x_0, x_1, \cdots, x_{n-1})$, the posterior probability over the image labels $\mathbf{l}$ is computed using Bayes' rule

$$\begin{aligned} Pr(\mathbf{l}|\mathbf{x}) &= \frac{Pr(\mathbf{x}|\mathbf{l})Pr(\mathbf{l})}{Pr(\mathbf{x})} \\ &= \frac{\prod_{i=0}^{n-1} Pr(x_i|l_i)Pr(l_0\cdots l_{n-1})}{Pr(x_0\cdots x_{n-1})}. \end{aligned} \tag{5.8}$$

Dropping the numerator, the maximum *a posteriori* estimate in the log domain is

$$\begin{aligned}
\hat{l}_{0,\cdots,n-1} &= \underset{l_0,\cdots,n-1}{\operatorname{argmax}} \left[\prod_{i=0}^{n-1} Pr(x_i|l_i) Pr(l_0 \cdots l_{n-1}) \right] \\
&= \underset{l_0,\cdots,n-1}{\operatorname{argmax}} \left[\sum_{i=0}^{n-1} \log[Pr(x_i|l_i)] + \log[Pr(l_0 \cdots l_{n-1})] \right] \\
&= \underset{l_0,\cdots,n-1}{\operatorname{argmax}} \left[\sum_{i=0}^{n-1} \log[Pr(x_i|l_i)] + \sum_{i=0}^{n-1} \log[Pr(l_i|l_{\mathcal{N}_i})] \right] \\
&= \underset{l_0,\cdots,n-1}{\operatorname{argmax}} \left[\sum_{i=0}^{n-1} \log[Pr(x_i|l_i)] + \sum_{i=0}^{n-1} \sum_{j \in \mathcal{N}_i} \log(\gamma_{ij}(l_i, l_j)) \right], \qquad (5.9)
\end{aligned}$$

where we have now written the prior using the Markov property and pairwise potential functions. We can write this in terms of the negative log-likelihood:

$$\begin{aligned}
\hat{l}_{0,\cdots,n-1} &= \underset{l_0,\cdots,n-1}{\operatorname{argmin}} \left[\sum_{i=0}^{n-1} -\log[Pr(x_i|l_i)] - \sum_{i=0}^{n-1} \sum_{j \in \mathcal{N}_i} \log(\gamma_{ij}(l_i, l_j)) \right] \\
&= \underset{l_0,\cdots,n-1}{\operatorname{argmin}} \left[\sum_{i=0}^{n-1} -\log[Pr(x_i|l_i)] + \sum_{i=0}^{n-1} \sum_{j \in \mathcal{N}_i} \phi_{ij}(l_i, l_j) \right]. \qquad (5.10)
\end{aligned}$$

This equation is the same as Equation 5.2 which we repeat here for completeness:

$$E(\mathbf{l}) = \sum_{i=0}^{n-1} U_i(l_i) + \sum_{i=0}^{n-1} \sum_{j \in \mathcal{N}_i} P_{i,j}(l_i, l_j).$$

From this, we note:

- The prior $Pr(\mathbf{l})$ obeys the Markov property. This is enforced through the pairwise potentials and it can be used to enforce particular configurations of labels. A common approach is to favour smooth labelling.
- The unary potential is the likelihood function that measures the agreement between the observed data and the data model for the assigned label.

5.4 Potential Functions and Inference

Thus far, we have presented image labelling from the view of GCs and MRFs. Here we show how the background texture models and smoothness priors are used to improve foreground detection. The labelling that we optimize consists of binary variables $l_i \in \{0, 1\}$. The graph is

set up in the following fashion to compute the maximum flow [49]:

- The capacities of edges from a pixel i to the source and sink are set by the unary potentials $U_i(1)$ and $U_i(0)$ respectively.
- Edges between pixels are set by $P_{i,j}(0,1)$ and $P_{i,j}(1,0)$ and reflect the cost when i is connected to the source and j to the sink, and vice versa. For a general cost structure, refer to Prince [49].

When the pairwise costs are negative, a simple way to adjust them when computing the maximum flow is to add the same constant to each capacity so that they are non-negative. This does not affect the solution for labelling. In our particular problem, the potential functions are set up as follows, where τ is the detection threshold:

- For the unary data term of site i, we use the distance D to the closest texture component to compute the weight to a terminal node. D is computed using the Mahalanobis distance (D_{MAH}) for the GMM and the chi-square distance (D_{CHIS}) for the STM. If $D < \tau$, site i is connected to the sink (background class) with weight $\frac{D}{\tau}$; else it is connected to the source (object class) with weight $\frac{D}{\tau}$ where τ is simply a scaling factor that we use to normalize the unary terms.
- For the pairwise potentials, if $j \in \mathcal{N}_i$, we connect site i to site j with the real-valued weight λ. Our method uses an 8-neighbourhood system and λ controls the level of smoothing in the optimization.

The data potential functions assign weights to nodes based on their distance from the classification threshold. Thus, outliers of the model have much higher weights than inliers.

If one were to take the probabilistic interpretation of a MRF, the logarithm of the Gaussian function in the GMM resolves to the Mahalanobis distance plus a constant. Thus, the weight to a terminal node is determined by the distance to the mean of the Gaussian function and that is what we have above. In the case of the STM, our interpretation is that of a straightforward maximum flow problem with data weights determined by D_{CHIS}. The inference for our model is performed using the Boykov-Kolmogorov maximum flow library[1]. A performance decrease of about 15% on the frame rate reported in the previous chapter was observed.

5.5 Experiments

The main experiments in this chapter are designed to show that through the smoothing prior the GC optimization in general reduces the number of false positives in the detections. In some

[1] `http://vision.csd.uwo.ca/code/`

cases, it can also increase the number of true positives. Two sets of results are presented:

- An analysis of the ROC performance and the FPR of the methods.
- A comparison to existing methods showing the improvement provided by the GC.

The parameter settings in Chapter 4 remain unchanged:

- Number of model components: $K = 4$.
- Minimum component weight: $\pi_{min} = 0.075$.
- Texture window size: 32×32.
- Learning rate: $\alpha = 0.05$
- The weighting parameter γ was varied from 1.0 to 4.0 to generate the ROC curves. Note that the weight to a terminal node is thus $\frac{D}{\gamma\tau}$;
- The parameter controlling the smoothness in the GC is $\lambda = 0.25$.

Some results using the GC are shown in Figures 5.1 and 5.2. It is easy to see that the optimization improves on the initial image labelling:

- The false positives that arise near the horizon are removed.
- Small image regions that are false positives are removed.
- The labelling of the targets improves in most cases because holes in the target are filled.

The ROC performance and comparison to existing methods are discussed next.

5.5.1 ROC Performance on Maritime Sequences

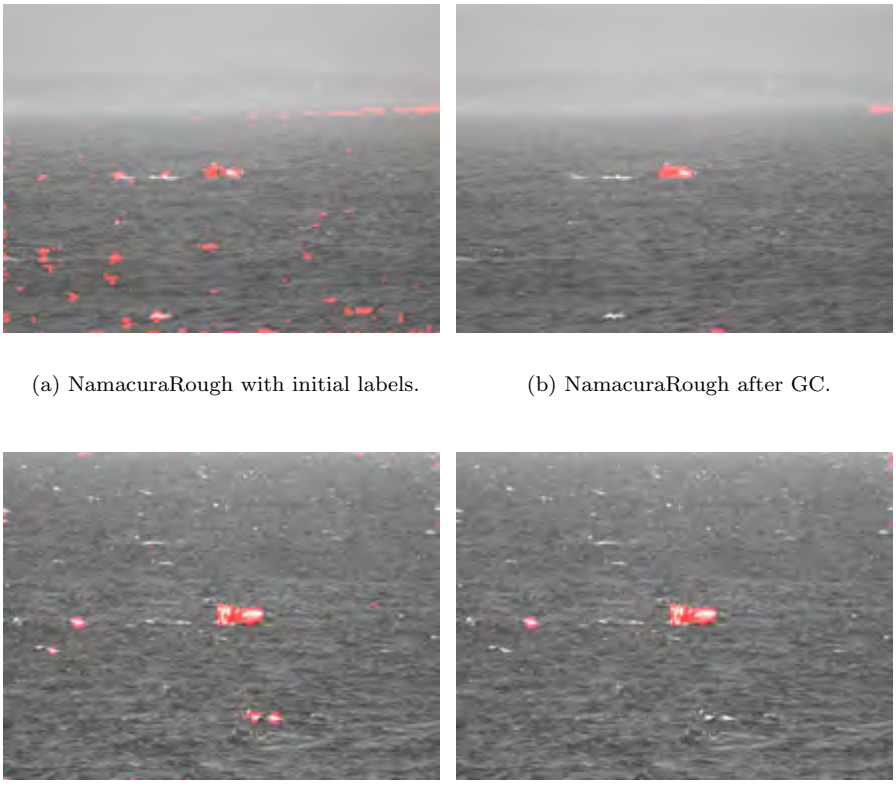
Table 5.1 compares the ROC AUC both with and without the graph cut optimization. For each texture model, GMM and STM, the highest AUC is highlighted in bold. With the exception of the one result from the NamacuraYacht sequence the GC improves the AUC for all cases. The ROC AUCs are all greater than 0.8. The system FPR performance at 50% TPR is shown in Table 5.2. Once again the best FPR for each texture model is shown in bold. The GC optimization produces the lowest FPR for all test videos.

Table 5.1: ROC AUC.

Sequence	**GMM**	**STM**	**GMM+GC**	**STM+GC**
Rhib	0.8827	0.8718	**0.8978**	**0.9071**
NamacuraYacht	**0.9208**	0.9088	0.9178	**0.9109**
NamacuraRough	0.7722	0.7882	**0.8364**	**0.8395**
Boats1	0.9590	0.9612	**0.9621**	**0.9634**
Boats2	0.9666	0.9733	**0.9902**	**0.9906**

Table 5.2: FPR at 50% TPR.

Sequence	**GMM**	**STM**	**GMM+GC**	**STM+GC**
Rhib	0.0033	0.0044	**0.0015**	**0.0008**
NamacuraYacht	0.0083	0.0159	**0.0008**	**0.0035**
NamacuraRough	0.1056	0.0843	**0.0821**	**0.0602**
Boats1	0.0147	0.0052	**0.0113**	**0.0033**
Boats2	0.0071	0.0051	**0.0032**	**0.0017**



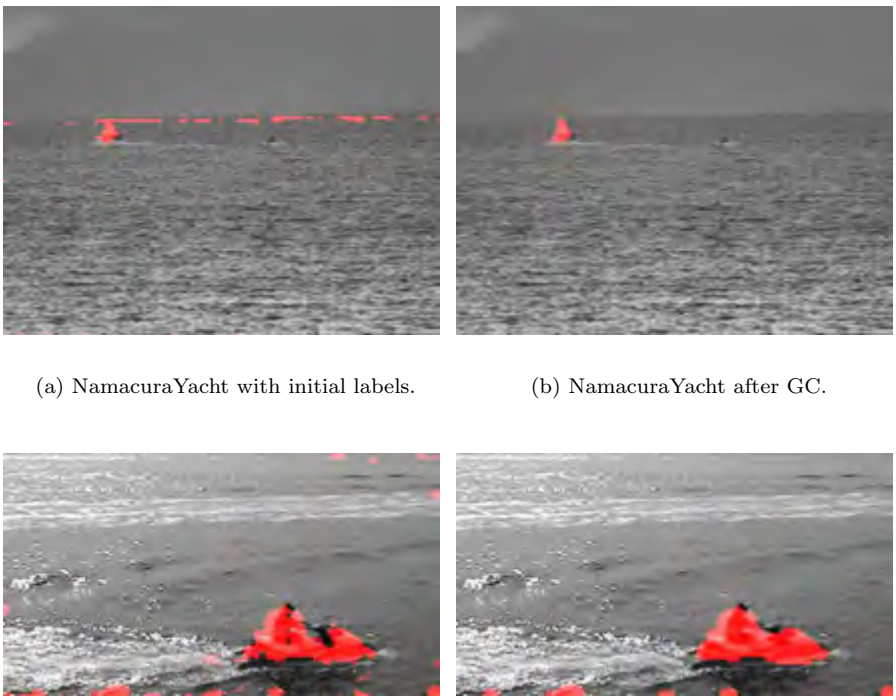
(a) NamacuraRough with initial labels.

(b) NamacuraRough after GC.

(c) Rhib with initial labels.

(d) Rhib after GC.

Figure 5.1: Foreground detection using a GC.



(a) NamacuraYacht with initial labels.

(b) NamacuraYacht after GC.

(c) Boats2 with initial labels.

(d) Boats2 after GC.

Figure 5.2: Foreground detection using a GC.

5.5.2 Comparison to Existing Methods

The comparative results that were presented in Chapter 4 are compared to our GC results. Table 5.3 shows the previous comparison without the GC optimization. We concluded earlier that:

- The dynamic texture models offer the best detection performance, in general, for maritime targets in our test sequences.
- Our model, although not the best performer overall, is comparable to the other state of the art methods and in some cases offers better performance.

The comparison with optimized image labelling is shown in Table 5.4. On the Boats1 video the dynamic texture models still perform the best, followed by PCA. However, we also see that the performance figures of our methods are now very close to the other state of the art methods. The GMM provides the third best result after PCA on the Boats1 sequence. The STM performs the best on the Boats2 sequence, having the highest ROC AUC and the lowest FPR. On this same sequence the GMM performs second best, followed by the dynamic texture models. The new comparative results show that the proposed models are better than the state of the art for some video sequences and, in general, can offer similar performance to most existing methods for this type of problem.

5.6 Conclusion

The graph cut optimization that was presented in this chapter provides a principled way to improve maritime target detection using the background texture models. It was shown through experimental analysis that the GC optimization improves the AUC of the ROC curve for the test sequences. Under most scenarios the FPR of the system is improved; under some scenarios, holes in targets are filled and this leads to an improvement in the TPR. Comparison of the graph cut results with those of existing methods show that the proposed technique is competitive and offers similar performance to the state of the art.

Table 5.3: Previous comparison to state of the art.

Sequence		SG	PCA	[43]	DTM1	DTM3	GMM	STM
Boats1	AUC	0.9516	0.9698	0.9493	0.9886	**0.9910**	0.9590	0.9612
Boats1	FPR(TPR=0.90)	0.1030	0.0865	0.1067	0.0120	**0.0061**	0.1040	0.1122
Boats2	AUC	0.8952	0.8136	0.7966	0.8916	0.9266	0.9666	**0.9733**
Boats2	FPR(TPR=0.55)	0.0261	0.1527	0.1495	0.0318	0.0088	0.0091	**0.0066**

Table 5.4: Comparison of GC to state of the art.

Sequence		SG	PCA	[43]	DTM1	DTM3	GMM+GC	STM+GC
Boats1	AUC	0.9516	0.9698	0.9493	0.9886	**0.9910**	0.9621	0.9634
Boats1	FPR(TPR=0.90)	0.1030	0.0865	0.1067	0.0120	**0.0061**	0.0991	0.1039
Boats2	AUC	0.8952	0.8136	0.7966	0.8916	0.9266	0.9902	**0.9906**
Boats2	FPR(TPR=0.55)	0.0261	0.1527	0.1495	0.0318	0.0088	0.0039	**0.0021**

Chapter 6

Feature Tracking

6.1 Introduction

In the previous chapters, region-based texture models were proposed for describing the appearance of the ocean in a maritime video scene. Outliers of these background models were labelled potential targets. The texture models describe the spatial grey level information in a scene and do not consider temporal dependencies in the video. It was mentioned in an earlier chapter that the registration of image frames from maritime video is a difficult and complex problem due to the dynamic nature of the pixel appearances. However, in a maritime scene containing targets, the targets exhibit locally stable appearances that one may exploit for detecting objects that persist over a number of video frames. Under most scenarios it is a fair assumption that salient features of the ocean texture will persist for much shorter intervals, due to the dynamic ocean appearance, compared to features on maritime targets.

The idea that is described above falls within the concept of TBD, a well-known approach in the radar literature [2]. TBD is designed to work in a cluttered environment with the core idea being the accumulation of evidence over time for an observation until it is known that the observation originates from background or target. This is beneficial for maritime tracking in a cluttered environment with dynamic background appearances. In the case of a maritime scene, one may monitor the lifespan of a track with the view that rigid objects will have longer lifespans than ocean features. Feature detection and tracking is proposed as a feasible approach for TBD in maritime scenes. TBD is used to label feature points in a video frame as a precursor to operations such as target extraction and construction of model distributions.

We propose using the length of a track as the confidence score for a track. A minimum track

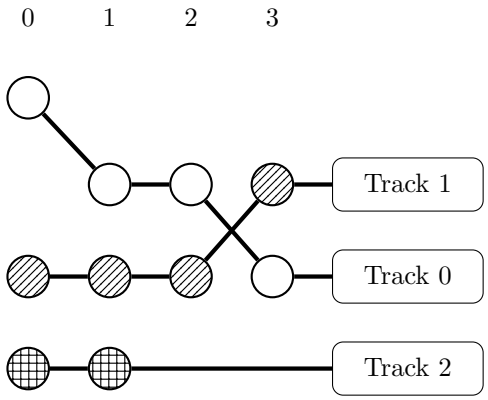


Figure 6.1: The TBD concept.

length π is specified for the detection threshold. Then,

$$l_i = \begin{cases} 1, & length(f_i) \geq \pi \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

where l_i is the label of feature f_i and $length(f_i)$ is the track length of feature f_i. Figure 6.1 shows the TBD concept for $t = 0$ to $t = 3$. Tracks 0 and 1 have observations (denoted by the circles) for each point in time, whereas Track 2 has only two measurements. If one were to set π=3, the tracks with lengths greater than 3 would be marked target; the other tracks would be marked background. In this case, Track 0 and Track 1 are potential targets. In the remainder of the chapter, the feature tracking method is presented followed by the experimental results. The chapter ends with concluding comments.

6.2 Method

The feature tracking method that is implemented has the following components:

- Feature point detection.
- Descriptor computation.
- Tracking of features.

Features are detected in each video frame I_t, at time t, and their corresponding descriptors are computed. Feature tracking is used because the video frames are captured from the same viewpoint and, apart from camera shake, the geometric and photometric changes between frames are expected to be small. Thus, features in frame I_{t-1} are tracked in I_t.

6.2.1 Feature Detection

The Harris corner detector is used to detect interest points because it is fast to compute and the image patches surrounding the corners are generally good for tracking [79]. The method computes a score for each pixel in an image based on directional changes. A corner point has strong directional variations. Harris and Stephen used the sum of squared differences of an image patch centred at (x, y) and its shifted version to determine the corner score for a shift of (u, v):

$$S(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2. \tag{6.2}$$

Using a Taylor expansion and some manipulation, one gets

$$\begin{aligned} S(u,v) &\approx \begin{bmatrix} u & v \end{bmatrix} \left(\sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u & v \end{bmatrix} A \begin{bmatrix} u \\ v \end{bmatrix}. \end{aligned} \tag{6.3}$$

The matrix A is the Harris matrix and its eigenvalues, λ_1 and λ_2, indicate the presence of an interest point at location (x, y) in the image. A corner is detected when both λ_1 and λ_2 have large positive values. The weights $w(x, y)$ are a Gaussian weighted window, for example. Precise computation of the eigenvalues is computationally expensive and the following corner score function S is proposed where κ is tunable for sensitivity:

$$S = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 \tag{6.4}$$

$$= \det(A) - \kappa(\text{trace}(A))^2. \tag{6.5}$$

In this equation $\det(\bullet)$ and $\text{trace}(\bullet)$ are the determinant and trace of a square matrix respectively. Corner scores greater than a predefined threshold are selected as detected features. Local non-maximal suppression is performed to select the final set of features. Tomasi and Kanade [81], and Shi and Tomasi [82] proposed modifications to the Harris corner detector for detecting good features to track. Their technique computes the actual eigenvalues.

6.2.2 Feature Descriptor

A descriptor is computed for each interest point found by the corner detector. The descriptors are important for the matching process because they describe the image appearance in the region of the corner. Local features between frames generally vary due to changes in brightness, scale and orientation. Even if one compensates for this, there can be other appearance changes that make it difficult to match or track features. Several image descriptors have been developed for robustness. The feature descriptor that we use is the SIFT descriptor [78]. In the original

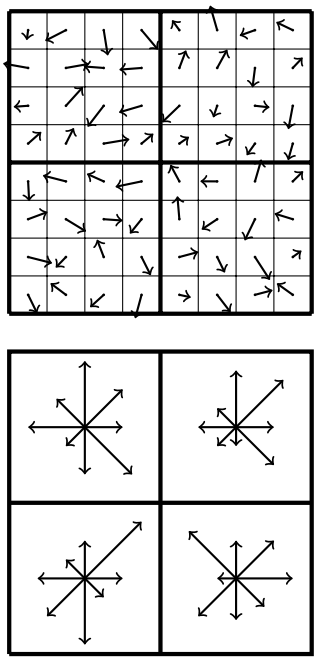


Figure 6.2: SIFT-type descriptor for 2×2 cells and 8 orientations.

SIFT descriptor, gradient direction as well as magnitude is computed in a 16×16 window around a feature point. This window is then split into 4×4 cells, and within each cell an 8-dimensional histogram of gradient orientations is computed. A histogram bin is weighted by the magnitude of the gradient and the distance of the gradient position from the feature point location. Points close to the feature point have greater contributions. The histogram is a 128-dimensional vector and it is normalized to unit length; the resulting values are clipped to 0.2 and the vector is re-normalized to unit length. The normalization and clipping make the descriptor insensitive to photometric variations.

The one deviation from the definition above is the size of the descriptor window — a 32×32 window is used in this work. A SIFT-type descriptor is shown in Figure 6.2 for 2×2 cells and 8 orientations.

6.2.3 Feature Tracking

The feature tracking algorithm maintains a list F_t of features that are being tracked in the image. It does the following to manage the list:

- **Feature tracking:** Features from the previous frame F_{t-1} are tracked in the current frame F_t and they are updated.

- **Feature deletion:** Features that cannot be matched are deleted.
- **Feature initialization:** New features are initialized and added to the tracking list.

We have $F_t = \{f_0, f_1, \cdots, f_{n-1}\}$ for n time-varying features where $f_i = \{x, y, vx, vy, \mathbf{d}, l, m\}$ such that:

- (x, y) is the feature location in the image plane.
- (vx, vy) is the velocity of the tracked feature.
- $\mathbf{d}$ is the feature descriptor.
- l is the feature label.
- m is the track length of the feature.

Feature tracking is implemented using a hierarchical block matching strategy. We refer to this process as tracking because the patch matching only occurs in regions around keypoints from the previous frame. In other words, the position of a tracked keypoint in the current frame is dependent on its position in the previous frame.

A hierarchical search method is preferred because it speeds up the matching process and detects large displacements as well. The hierarchical scheme is implemented using an image pyramid with levels sub-sampled by a factor of two along each dimension as one moves up the pyramid. The lowest level is the full resolution image. The block matching can be done at sub-pixel locations if desired using bilinear interpolation to determine pixel values.

A hierarchical search method starts the patch matching process at the top of the pyramid and moves downwards towards the full resolution image — this is a coarse to fine matching process. A match found at location (x, y) in level L is propagated downwards and the starting point at level $L - 1$ becomes $(2x, 2y)$. Thus, rather than doing a full search at $L = 0$, one may do multiple small searches at each level so that a much smaller number of total computations is expended. Once the feature displacements have been estimated, their velocities are computed and stored. For consecutive frames, it is expected that the appearance of an image patch does not change much and that the motion an image patch undergoes corresponds to translation. Thus, to save on computations we only extract SIFT descriptors for comparison to tracked features when the flow has been computed at the lowest scale i.e. at the full resolution image. For flow computation, the image intensity patch is extracted and the sum of squared deviations is computed for the error metric.

We now explain the tracking algorithm. Let us assume that at time $t - 1$ a list of tracked features F_{t-1} has been updated and is available. For the video frame at time t, denoted I_t, do the following:

- Extract all corners and their descriptors from frame I_t and store them in C_t.
- Perform block matching to estimate the displacements of F_{t-1} from I_{t-1} to I_t.
- Once the locations in F_{t-1} have been matched, SIFT-types descriptors $\mathbf{s}_i$ are extracted at the matched locations in I_t and compared to the corresponding descriptors in F_{t-1} using the histogram intersection. If the match score is greater than a predefined threshold, the feature's state is updated using the new position and descriptor. The descriptor $\mathbf{d}_i$ for a feature is updated

$$\mathbf{d}_i^{new} = 0.9\ \mathbf{d}_i^{old} + 0.1\ \mathbf{s}_i \tag{6.6}$$

  and the feature location is replaced with the new matched location. The track length m_i is incremented for a feature that has been updated. If the match score is below a predefined threshold, the feature is deleted. The feature label l_i is determined using Equation (6.1). All active features are placed in F_t.
- New features are initialized and added to F_t in the following manner. Each feature in C_t is examined by searching around its location to check if it contains any of the feature locations in F_t. If its neighbourhood does not have other features, it is added to F_t as a new feature. If there is a feature in F_t that is in its neighbourhood, it is discarded.

Under some conditions a feature may be temporarily occluded, and this makes it difficult to track because descriptors cannot be matched. To handle these situations, the feature is propagated for a few frames until a match occurs. The estimated velocity is used to predict the location of the feature. This approach does not work when there is adverse motion or appearance changes that make recovery difficult.

6.3 Experiments

The performance of TBD for detecting maritime targets in video is measured using ROC curves that are calculated for the test sequences. The following strategy was used to generate the ROC curves:

- Given a set of detections in a video frame, those that intersect with targets in the associated ground truth are positives. The remaining detections are negatives.
- Using the TBD approach, targets are labelled by the algorithm and the true positives and false positives are determined.

The ROC curves are an intuitive indicator of the number of stable interest points detected on a target. For example, the 50% TPR shows one that half of the feature points on a target are

stable for the given FPR. It is assumed that there is a fair coverage of feature points on the target. We noted that the flat low contrast regions above the horizon line in NamacuraRough and NamacuraYacht contained zero corner points. Thus, horizon detection was not needed for these performance figures.

For the experiments in this chapter, a four-level image pyramid was created for the feature matching. At each scale, a 5×5 neighbourhood is searched to match image regions. At the lowest level this corresponds to a search area of 33×33 in the original frame. We found that this was sufficient for our test sequences. The patch matching for computation of the flow is performed on 15×15 patches at all levels of the pyramid. We found that this patch size made the flow computation stable. The feature tracking parameters were set up as follows:

- Region size for local non-maximal suppression in the Harris corner detector: 11×11 window.
- Minimum track length score for foreground detection: $\pi \in \{1, 2, 4, 6, 8, 10\}$ is used to generate the ROC curves.
- Search window for adding new feature points: 11×11.
- Corner detector sensitivity parameter: $\kappa = 0.04$.
- Minimum corner strength for feature detector: S=0.0005. The 8-bit video data was normalized to $[0, 1]$ and the Sobel operator was used to compute image gradients. The strongest 200 features were selected for tracking.
- Minimum match score: 0.75.
- Descriptor window size: 32×32.

The minimum corner strength value affects the tracking performance: a too small value introduces features that cannot be tracked well, as pointed out by Shi and Tomasi [82]. This increases the processing time as well because a large number of features are processed. Weak features also create more false negatives and false positives. Most of these problems are avoided by selecting the strongest 200 features. Details of the software implementation and the experimental results are presented next.

6.3.1 Implementation and Computational Performance

The feature tracker was implemented using CUDA on Ubuntu 14.04. A GTX860M GPU was used for most of the image processing. This laptop GPU has the following hardware specification:

- 640 shader units at 1029 MHz — these are 5 streaming multiprocessors with 128 arithmetic logic units (cores or stream processors) per unit.
- 4GB memory (128-bit interface and 80GB/sec bandwidth).

CUDA is a computing platform and programming architecture developed by the technology company NVIDIA. NVIDIA is one of the leading manufacturers of GPUs. CUDA enables significant increases in computing power through the use of multi-threading cores in graphics cards. In the CUDA architecture, a GPU chip consists of a group of blocks. This is known as a grid. The grid has no synchronization between blocks. A block is a set of threads that can execute in any order — threads in a block can cooperate. The GPU chip is structured so that it has a collection of multiprocessors. Each multiprocessor handles one or more blocks in a grid. Blocks are not divided across different multiprocessors. A multiprocessor is further divided into stream processors that handle one or more threads in a block.

A CUDA kernel executes on a set of data blocks (analogous to the physical GPU blocks) with threads allocated to each block for data processing. A video frame is processed by allocating a portion of the frame to each block. Thus, for example, one can allocate an image tile of size N to a block and have N threads process the tile (one thread per pixel). It is important to remember that each architecture has a compute capability that defines the GPU capabilities (such as maximum number of threads and blocks that can execute concurrently on a multiprocessor). In the general case, a chunk of data is processed by splitting it up into blocks and allocating processing threads to each block.

The GPU implementation of the feature tracking method detects and tracks 200 features at 28 frames-per-second for 1360×1024 8-bit video; for the same frame rate, 290 features are tracked on 720×400 video. At this rate a single feature is tracked in less than a millisecond. There are a large number of computations that are carried out on each video frame — they all execute on the GPU:

- Four level image pyramid construction.
- Harris corner detection that includes computation of image derivatives, summation of derivatives over a 5×5 weighted window and local non-maximal suppression over an 11×11 region for each detected corner.
- Block matching across the image pyramid at the detected feature point locations.
- Computation, comparison and updating of 32×32 descriptors.
- Updating of the tracks and the feature list.

Table 6.1: AUC and FPR at 50% TPR.

Sequence	AUC	FPR
Rhib	0.8067	0.0374
NamacuraYacht	0.9784	0.0002
NamacuraRough	0.6265	0.3198
Boats1	0.8636	0.0136
Boats2	0.9066	0.0042

6.3.2 ROC Performance

Figure 6.3 shows the ROC curves for TBD while the AUC and FPR are summarized in Table 6.1. Similar to the performance of the background texture models, it is observed that most of the ROC curves have the characteristic shape for a good classifier. At 50% TPR the detector has less than 4% false positives for all test cases except for the challenging NamacuraRough sequence that has close to 32% false positives.

The NamacuraRough sequence is very challenging for TBD for several reasons and it highlights the conditions that are difficult for the method:

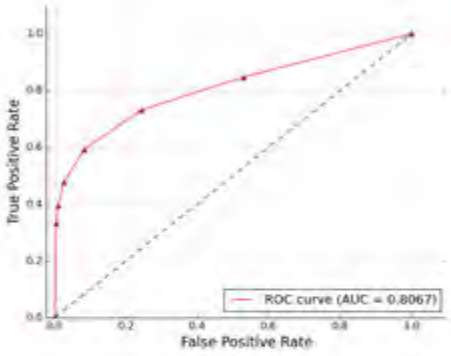
- The target is partially occluded for most of the video. Thus, it is difficult to maintain tracks with long lifespans. Lowering the track score π increases the FPR.
- The video contains severe camera shake at times and this makes tracking difficult if the image shift, relative to the previous frame, is not within the search area in the coarse to fine block matching strategy.

In spite of these challenges, the method is still able to achieve stable tracking of some parts of the vessel as shown by the green crosses in Figure 6.6(c). The red crosses are feature points labelled background using TBD.

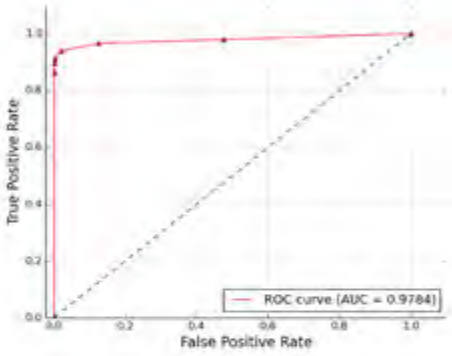
6.3.3 Detection Results

Some detection results are shown in Figures 6.4 to 6.6, where green points are foreground and red are background ones. There are a low number of false positives for most of the test sequences; the foreground feature points are also well clustered on the targets. A qualitative analysis of the results showed that the dynamic appearance of the ocean is advantageous to the proposed method because corners detected on the ocean texture do not persist for a long time due to elements such as:

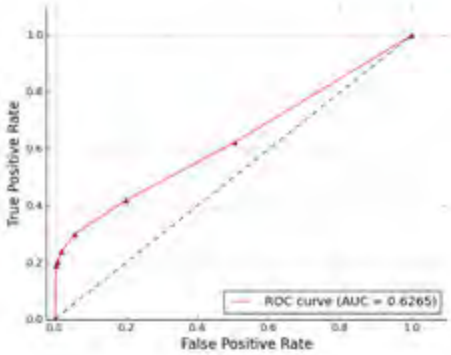
- Surf appearing and disappearing.



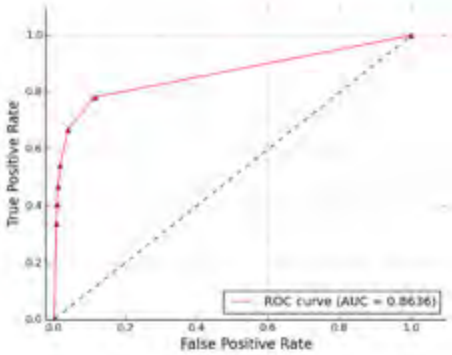
(a) Rhib.



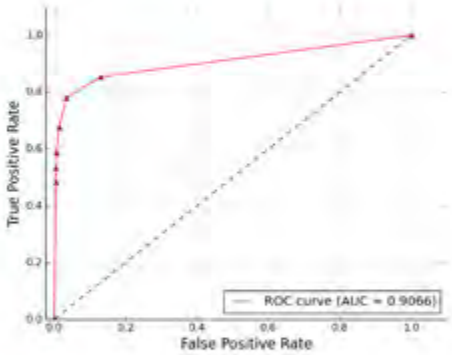
(b) NamacuraYacht.



(c) NamacuraRough.



(d) Boats1.



(e) Boats2.

Figure 6.3: TBD ROC performance.

- The motion of the waves that causes corners and textures to appear and disappear periodically.
- Specular reflections on the ocean that also appear and disappear periodically.

In some cases, the appearance change of an image region is not smooth like a rigid object and, hence, the feature appearance model cannot track this change. Thus, the tracking of these types of features fails and it works to the advantage of TBD.

The feature detection and tracking method is able to detect and track features even in the presence of camera shake and out-of-plane object motion such as in the Rhib sequence. Features are also detected on small targets, such as in the Boats1 sequence in Figure 6.5. The Namacura vessel is not detected in the NamacuraYacht sequence because it has low contrast against the ocean background. It must be highlighted that all feature points detected on a target may not be labelled target due to several factors such as occlusion, temporal appearance changes and accumulation of error in feature tracks (the drift problem). These problems are difficult to address as a whole. However, if at least 50% of the detected corner points on a target are temporally persistent, one may be able to track the whole object robustly.

6.4 Conclusion

This chapter presented feature detection and tracking as a means of locating key points on maritime targets. The method itself is very simple and uses the length of a track as a measure of persistence of an object in a scene. The results, both quantitative and qualitative, show that TBD is feasible for detecting features on maritime targets at 50% TPR where the FPR for most of our test sequences was less than 4%. The method suffers when the input video has severe camera shake or the target undergoes a large proportion of occlusions. The labels that are assigned to feature points can be used to construct more descriptive background and foreground appearance models in an unsupervised fashion.

This work has proposed two approaches for detecting maritime targets. The region-based texture models are advantageous when dense outputs are required – they generate a label for every pixel in a frame. These models do not require registered image frames because they do not model the temporal dynamics of the ocean. However, their computing time is almost four times slower than TBD and they have difficulty detecting small targets. The TBD method is significantly faster than the texture modelling approaches and this is likely due to its full GPU implementation. However, it generates sparse points and requires time-varying data that can be tracked. Unlike the texture models proposed in this thesis, TBD is also suitable for detecting corners on small targets.

Figure 6.4: TBD results for Rhib. Red features are background and green are foreground.

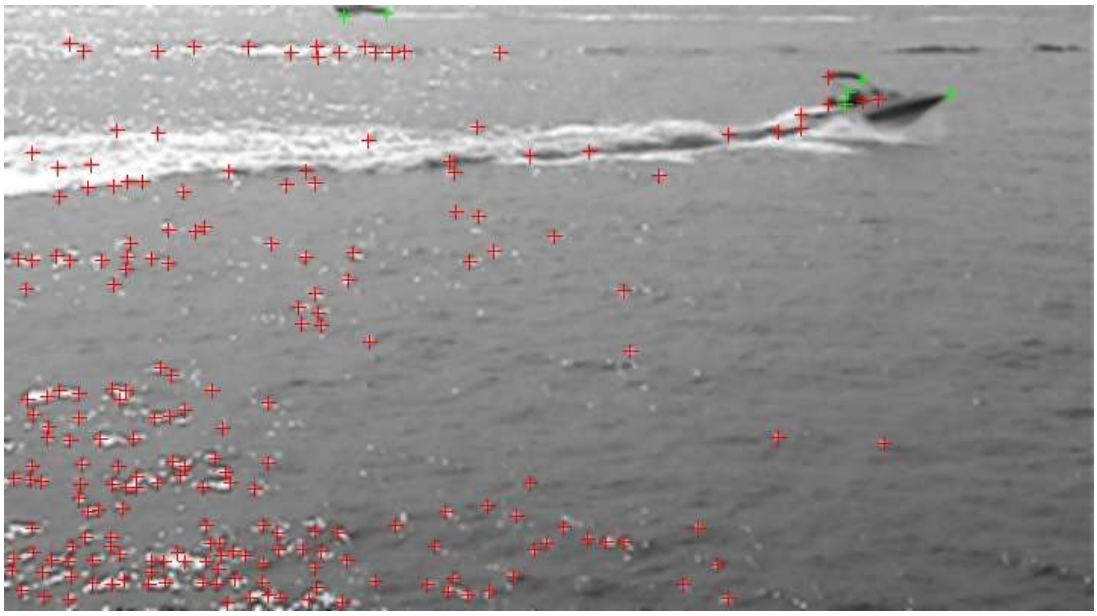


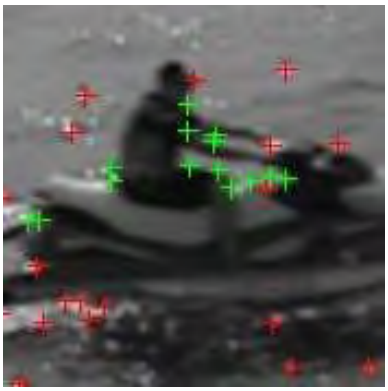
Figure 6.5: TBD results for Boats1. Red features are background and green are foreground.

(a) Rhib.

(b) NamacuraYacht.

(c) NamacuraRough.

(d) Boats1.



(e) Boats2.

Figure 6.6: TBD results. Red features are background and green are foreground.

Chapter 7

Conclusions

In this thesis, several new ideas have been presented for detecting maritime targets from video data. Three ideas are put forward for robust unsupervised detection of foreground objects:

- Region-based texture representations of the ocean background such that outliers of these models are potential maritime targets.
- Graph cut optimization, for spatially coherent image labelling, that improves detection performance when using the region-based texture model.
- Feature tracking for TBD as a means of detecting stable features, corresponding to maritime targets, on the ocean.

These ideas are summarized in the thesis statement presented in Chapter 1:

Robust unsupervised detection of a maritime target from a static or moving camera can be achieved using a region-based texture model, within a graph-cut framework, and feature tracking.

The thesis statement is proved by the findings of this work and they are summarized herein.

7.1 Summary of Results

Chapter 1 presented the introduction to this work, where the problem was described and the thesis statement was introduced together with the objectives of this study. The limitations and assumptions of the research were also highlighted - this work is primarily concerned with detection of one or two maritime targets of reasonable size and it is assumed that they are

surrounded by an ocean background. The main problem that this work addresses is unsupervised detection of a maritime target from monochrome video acquired using a static or moving camera. There is limited research in this area and the proposed methods in this thesis address those limitations.

Chapter 2 explored background information on the application area and prior art in video-based maritime target detection and tracking. Related work in background modelling, and feature detection and tracking was also presented. The significance of the proposed work was highlighted by contrasting it with the limitations in current approaches. Thus, three main novelties were detailed:

- Targets are detected in grey scale video without a dependence on colour or thermal data as is commonly done.
- The method is unsupervised and does not make use of any training samples with known labels.
- The method is not limited by the motion of the cameras.

The background texture model method is related to the work of Voles [9] and several variations and improvements on his research are listed.

A new approach for background modelling of a maritime video is proposed in Chapter 3. The model is based on the assumption that ocean texture is the dominant class in a maritime video frame that contains a small number of targets. This characteristic is exploited to build a texture model where the components that describe the most amount of image data are likely to be part of the ocean class. Outliers of this model are potential maritime targets. The texture appearance is described using LBP descriptors - histogram texture distributions - and two texture models are proposed: the GMM and STM. Both models are defined for k components that are initialized using an information theoretic clustering approach for histogram data. The clustering is initialized using the k-means++ technique. The GMM is updated online using an approximation of the EM algorithm; the STM is updated online using a low pass filter on the texture distributions. The models are developed to be fast and efficient in software implementation.

A number of experimental results for the background texture models were presented in Chapter 4. Five test sequences were used to generate foreground detection performance metrics, of which two sequences were used for comparison to current state of the art in background modelling. The different aspects of the texture models were shown to be effective for tasks such as texture feature clustering, saliency and horizon detection. The GMM and STM were shown to offer similar levels of foreground detection performance. However, the STM was more efficient computationally, running at 8.5 frames-per-second compared to the GMM's 6.0

frames-per-second, after model initialization. The foreground detection performance metrics were computed from ROC curves and for almost all test videos:

- The ROC curves had the characteristic shape of a good classifier.
- An AUC of 0.85 or greater was observed.
- At 50% TPR the FPR was less than 2%.

Slightly worse performance was observed for just one video sequence, NamacuraRough, and this is deemed a very difficult test case. The results also showed that false detections, such as target wakes and white caps on the ocean, were similar to those reported by other authors. Small targets are difficult to detect in general. This can be addressed by increasing the detection threshold. However, this increases the FPR and it provides poor foreground-background separation. In comparison to the state of the art, which are all pixel-wise models, our methods perform comparatively well. However, our method is still preferred because these models will fail on test sequences acquired with moving cameras.

The graph cut optimization presented in Chapter 5 minimizes an energy function on the image labelling so that smooth solutions are preferred. The texture models are used to compute the data terms in the energy function. Pairwise potentials enforce spatial coherence in local neighbourhoods. The optimization improved the previous ROC AUC for all test cases to over 0.8 and it also reduced the FPR for all test sequences. Compared to the state of the art, the graph cut optimization improves on the original results. On the Boats1 sequence, the GMM had the third best result after the dynamic texture models and PCA. On the Boats2 sequence our methods perform the best, better even than the dynamic texture models. These results show that the proposed approach offers similar performance to the state of the art for maritime sequences.

In Chapter 6, keypoint features on maritime targets were discovered using TBD. This was implemented with feature tracking. The TBD concept computed a track score for foreground confidence based on the length of a track. It was proposed that maritime targets will persist in a scene whereas features on the ocean will have much shorter lifespans. Thus the track score was used to detect potential foreground features. The results showed that at 50% TPR the FPR was less than 4% for all test sequences except NamacuraRough. The NamacuraRough sequence contained several instances of target occlusion and this made detection difficult. However, it was shown qualitatively that TBD was still able to detect a small number of stable features on the target in this sequence. For the remaining sequences, the detected foreground feature points were well clustered on the targets. The method also worked well for small targets, in contrast to the texture models. The feature tracking implementation executed at 28 frames-per-second on 1360×1024 video for approximately 200 keypoints, using a coarse to fine block matching strategy on Harris corners and SIFT descriptors for the feature appearances.

7.2 Assessment of the Objectives

In Chapter 1, four objectives of this work were listed. They are now assessed.

7.2.1 Objective 1

To present unsupervised methods for detecting maritime targets in grey scale video captured by static or moving cameras.

This objective was met. The methods that were proposed are all unsupervised and rely on simple scene assumptions. They were tested on sequences captured by a moving camera, as well as on two sequences acquired with a static camera, with very promising results.

7.2.2 Objective 2

To present methods that are fast and efficient for real-time applications.

This objective was satisfied. The initialization of the texture models is not real-time. However, once the models are constructed the feature extraction, model update and foreground detection are fast and efficient. A reasonable processing frame-rate was reported for the GMM and STM. This performance figure dropped by approximately 15% when the graph cut inference was performed. Real-time processing of 28 frames-per-second was achieved by the feature tracking on 1360×1024 video.

7.2.3 Objective 3

To provide an empirical evaluation of the proposed methods on real-world data.

In order to achieve this objective, the proposed methods were tested on five real-world test sequences that are publicly available. Three sequences were acquired with a long range surveillance lens mounted on a moving camera. The remaining two videos were acquired with a static camera. These test scenarios contain targets of different sizes at different distances from the camera. The empirical evaluation consisted of a large number of experiments on these data sequences where the primary results were based on an analysis of ROC curves for detector performance.

7.2.4 Objective 4

To provide a comparison of the proposed methods to existing techniques.

This objective was met for the texture models and GC experiments. The foreground detection results for two video sequences, Boats1 and Boats2, were compared to several existing methods for modelling dynamic backgrounds. Although these methods used pixel-wise models, it was found that our methods offer similar levels of performance. The GC-optimized foreground detection, using both the STM and GMM, performed the best on the Boats2 sequence. On the Boats1 sequence, the proposed texture models performed the best after the dynamic textures and PCA techniques.

7.3 Future Work

There are several avenues arising from this research that should be explored in future work:

- **Improving the texture model:** The models in this work assume that the ocean texture components are described by a large fraction of the feature samples. There are situations where this assumption may be violated and this should be examined by incorporating additional information into the model maintenance algorithm. One may consider some form of supervised learning of a one-class classifier for the ocean texture.
- **Rejection of false detections:** The foreground detection method using the GMM or STM is very promising. The results can be enhanced by considering ways to reject false foreground detections such as white caps, wave crests and horizon texture boundaries. This can be accomplished using the idea suggested above i.e. supervised classification. In the case of horizon boundaries, additional computations should be performed to determine if a texture window is positioned across a texture boundary and the necessary steps should be taken to refine the LBP feature histogram.
- **Incorporation of different features:** The incorporation of additional types of descriptors into the texture models is expected to improve the detection performance. For example, one may consider using grey scale intensity jointly with the LBP for textureless regions. A meaningful study should consider the implications of using additional types of texture features, such as Gabor or textons, as well as center surround saliency measures.
- **Incorporation of additional cues into the GC:** The current GC energy function contains unary terms that are computed using the texture model. Additional unary terms may be included for cues such as grey scale intensity or center-surround saliency. Priors for target shape and position will benefit foreground detection.

- **Extending the TBD concept:** The proposed TBD method uses track length to compute the track score. More powerful scoring functions may take into consideration information such as appearance and motion for improved detection.
- **Improved feature tracking**: The feature tracking can be improved by storing the appearances of detected objects so that they may be used to re-detect a feature when a track is lost. This is particularly useful when occlusion or appearance changes occur.

Bibliography

[1] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 5, pp. 909–926, 2008.

[2] S. Blackman and R. Popoli, Design and analysis of modern tracking systems. Artech House, 1999.

[3] R. Moreira, N. Ebecken, A. Alves, F. Livernet, and A. Campillo-Navetti, "A survey on video detection and tracking of maritime vessels," IJRRAS, vol. 20, pp. 37–50, 2014.

[4] H.-J. Lee, L.-F. Huang, and Z. Chen, "Multi-frame ship detection and tracking in an infrared image sequence," Pattern Recognition, vol. 23, no. 7, pp. 785–798, 1990.

[5] J. G. Sanderson, M. K. Teal, and T. J. Ellis, "Characterisation of a complex maritime scene using Fourier space analysis to identify small craft," in Seventh International Conference on Image Processing and Its Applications, vol. 2, pp. 803–807, 1999.

[6] A. Smith and M. Teal, "Identification and tracking of maritime objects in near-infrared image sequences for collision avoidance," in Seventh International Conference on Image Processing and Its Applications, vol. 1, pp. 250–254, 1999.

[7] P. Voles, M. Teal, and J. Sanderson, "Target identification in a complex maritime scene," in IEE Colloquium on Motion Analysis and Tracking, p. 4, IET, 1999.

[8] P. Voles, A. Smith, and M. K. Teal, "Nautical scene segmentation using variable size image windows and feature space reclustering," in European Conference on Computer Vision, pp. 324–335, Springer, 2000.

[9] P. Voles, Feature-based object tracking in maritime scenes. PhD thesis, Bournemouth University, 2005.

[10] D. Socek, D. Culibrk, O. Marques, H. Kalva, and B. Furht, "A hybrid color-based foreground object detection method for automated marine surveillance," in Advanced Concepts for Intelligent Vision Systems, pp. 340–347, Springer, 2005.

[11] C. Bibby and I. D. Reid, "Visual tracking at sea," in International Conference on Robotics and Automation, pp. 1841–1846, 2005.

[12] C. Bibby and I. D. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in European Conference on Computer Vision, 2008.

[13] S. Fefilatyev, "Detection of marine vehicles in images and video of open sea," Master's thesis, University of South Florida, 2008.

[14] S. Fefilatyev and D. Goldgof, "Detection and tracking of marine vehicles in video," in International Conference on Pattern Recognition, pp. 1–4, IEEE, 2008.

[15] S. Fefilatyev, D. Goldgof, and C. Lembke, "Tracking ships from fast moving camera through image registration," in International Conference on Pattern Recognition, pp. 3500–3503, IEEE, 2010.

[16] S. Fefilatyev, D. Goldgof, M. Shreve, and C. Lembke, "Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system," Ocean Engineering, vol. 54, pp. 1–12, 2012.

[17] H. Bouma, D.-J. J. de Lange, S. P. van den Broek, R. A. Kemp, and P. B. Schwering, "Automatic detection of small surface targets with electro-optical sensors in a harbor environment," in SPIE Europe Security and Defence, pp. 711402–711402, International Society for Optics and Photonics, 2008.

[18] M. D. R. Sullivan and M. Shah, "Visual surveillance in maritime port facilities," in SPIE Defense and Security Symposium, pp. 697811–697811, International Society for Optics and Photonics, 2008.

[19] P. Westall, J. J. Ford, P. O'Shea, and S. Hrabar, "Evaluation of machine vision techniques for aerial search of humans in maritime environments," in Digital Image Computing: Techniques and Applications (DICTA), 2008, pp. 176–183, IEEE, 2008.

[20] P. Westall, P. O'Shea, J. J. Ford, and S. Hrabar, "Improved maritime target tracker using colour fusion," in International Conference on High Performance Computing and Simulation, pp. 230–236, IEEE, 2009.

[21] D. Bloisi and L. Iocchi, "ARGOS - a video surveillance system for boat traffic monitoring in venice," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23, no. 07, pp. 1477–1502, 2009.

[22] D. Bloisi, L. Iocchi, M. Fiorini, and G. Graziano, "Automatic maritime surveillance with visual target detection," in Proc. of the International Defense and Homeland Security Simulation Workshop (DHSS), pp. 141–145, 2011.

[23] H. Wei, H. Nguyen, P. Ramu, C. Raju, X. Liu, and J. Yadegar, "Automated intelligent video surveillance system for ships," in SPIE Conference Series, vol. 7306, International Society for Optics and Photonics, 2009.

[24] K. M. Gupta, D. W. Aha, R. Hartley, and P. G. Moore, "Adaptive maritime video surveillance," in SPIE Defense, Security and Sensing, pp. 734609–734609, International Society for Optics and Photonics, 2009.

[25] W. Kruger and Z. Orlov, "Robust layer-based boat detection and multi-target-tracking in maritime environments," in International Waterside Security Conference, pp. 1–7, IEEE, 2010.

[26] N. Pires, J. Guinet, and E. Dusch, "ASV: An innovative automatic system for maritime surveillance," Navigation, vol. 58, no. 232, 2010.

[27] R. Wijnhoven, K. van Rens, E. G. T. Jaspers, and P. H. N. de With, "Online learning for ship detection in maritime surveillance," in 31st Symposium on Information Theory in the Benelux, pp. 73–80, May 2010.

[28] Z. L. Szpak and J. R. Tapamo, "Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set," Expert Systems with Applications, vol. 38, no. 6, pp. 6669 – 6680, 2011.

[29] D. Frost and J.-R. Tapamo, "Detection and tracking of moving objects in a maritime environment using level set with shape priors," EURASIP Journal on Image and Video Processing, vol. 2013, no. 1, pp. 1–16, 2013.

[30] X. Bao, S. Zinger, R. Wijnhoven, and P. H. N. de With, "Water region detection supporting ship identification in port surveillance," in Advanced Concepts for Intelligent Vision Systems, pp. 444–454, Springer, 2012.

[31] X. Bao, S. Zinger, R. Wijnhoven, and P. H. N. de With, "Robust moving ship detection using context-based motion analysis and occlusion handling," in Sixth International Conference on Machine Vision, pp. 90670F–90670F, International Society for Optics and Photonics, 2013.

[32] P. Kaimakis and N. Tsapatsoulis, "Background modeling methods for visual detection of maritime targets," in Proceedings of the 4th ACM/IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream, pp. 67–76, ACM, 2013.

[33] K. Makantasis, A. Doulamis, and N. Doulamis, "Vision-based maritime surveillance system using fused visual attention maps and online adaptable tracker," in 14th International Workshop on Image Analysis for Multimedia Interactive Services, pp. 1–4, IEEE, 2013.

[34] M. Dawkins, Z. Sun, A. Basharat, A. Perera, and A. Hoogs, "Tracking nautical objects in real-time via layered saliency detection," in SPIE Defense and Security, pp. 908903–908903, International Society for Optics and Photonics, 2014.

[35] I. Bechar, T. Lelore, F. Bouchara, V. Guis, and M. Grimaldi, "Object segmentation from a dynamic background using a pixelwise rigidity criterion and application to maritime target recognition," in International Conference on Image Processing, pp. 363–367, IEEE, 2014.

[36] W.-C. Hu, C.-Y. Yang, and D.-Y. Huang, "Robust real-time ship detection and tracking for visual surveillance of cage aquaculture," Journal of Visual Communication and Image Representation, vol. 22, no. 6, pp. 543–556, 2011.

[37] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 747–757, 2000.

[38] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 657–662, 2006.

[39] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pp. 1778–1792, 2005.

[40] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in International Conference on Computer Vision, pp. 1219–1225, IEEE, 2009.

[41] Z. Yin and R. Collins, "Moving object localization in thermal imagery by forward-backward MHI," in Computer Vision and Pattern Recognition Workshop, pp. 133–133, IEEE, 2006.

[42] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," International Journal of Computer Vision, vol. 51, pp. 91–109, February 2003.

[43] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in International Conference on Computer Vision, pp. 44–50, 2003.

[44] A. B. Chan, V. Mahadevan, and N. Vasconcelos, "Generalized Stauffer-Grimson background subtraction for dynamic scenes," Machine Vision and Applications, vol. 22, no. 5, pp. 751–766, 2011.

[45] J. Huang, X. Huang, and D. Metaxas, "Optimization and learning for registration of moving dynamic textures," in International Conference on Computer Vision, pp. 1–8, IEEE, 2007.

[46] R. Vidal and A. Ravichandran, "Optical flow estimation and segmentation of multiple moving dynamic textures," in Computer Vision and Pattern Recognition, vol. 2, pp. 516–521, IEEE, 2005.

[47] C. M. Bishop, Pattern recognition and machine learning. Springer, 2006.

[48] R. Duda, P. Hart, and D. Stork, Pattern classification. John Wiley and Sons, 2001.

[49] S. J. Prince, Computer vision: models, learning, and inference. Cambridge University Press, 2012.

[50] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, pp. 1097–1105, 2012.

[52] D. M. Hawkins, Identification of outliers, vol. 11. Springer, 1980.

[53] Sonka, M. and Hlavac, V. and Boyle, R., Image processing, analysis and machine vison. PWS Publishing Company, 1999.

[54] R. Szeliski, Computer vision: algorithms and applications. Springer, 2011.

[55] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, 2002.

[56] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," IEEE Transactions on Image Processing, vol. 19, no. 6, pp. 1657–1663, 2010.

[57] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 5, no. 5, pp. 363–387, 2012.

[58] A. Jepson, D. Fleet, and T. El Maraghi, "Robust online appearance models for visual tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pp. 1296–1311, October 2003.

[59] T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," Pattern Recognition, vol. 32, no. 3, pp. 477–486, 1999.

[60] A. Ypma and R. P. Duin, "Support objects for domain approximation," in ICANN 98, pp. 719–724, Springer, 1998.

[61] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297, 1967.

[62] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[63] I. S. Dhillon, S. Mallela, and R. Kumar, "A divisive information theoretic feature clustering algorithm for text classification," The Journal of Machine Learning Research, vol. 3, pp. 1265–1287, 2003.

[64] R. Gonzalez and R. Woods, Digital image processing. Addison-Wesley Publishing Company, 2002.

[65] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in A Practical Approach to Microarray Data Analysis, pp. 91–109, Springer, 2003.

[66] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," International Journal of Computer Vision, vol. 77, no. 1, pp. 125–141, 2008.

[67] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in International Conference on Computer Vision, pp. 456–463, IEEE, 2003.

[68] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels," tech. rep., 2010.

[69] J. Chang, D. Wei, and J. Fisher, "A video representation using temporal superpixels," in Computer Vision and Pattern Recognition, 2013.

[70] C. Scharfenberger, A. Wong, K. Fergani, J. S. Zelek, and D. A. Clausi, "Statistical textural distinctiveness for salient region detection in natural images," in Computer Vision and Pattern Recognition, pp. 979–986, IEEE, 2013.

[71] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-guided flight stability and control for micro air vehicles," Advanced Robotics, vol. 17, no. 7, pp. 617–640, 2003.

[72] T. G. McGee, R. Sengupta, and K. Hedrick, "Obstacle detection for small autonomous aircraft using sky segmentation," in International Conference on Robotics and Automation, pp. 4679–4684, IEEE, 2005.

[73] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, pp. 381–395, 1981.

[74] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124–1137, 2004.

[75] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222–1239, 2001.

[76] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," Canadian journal of Mathematics, vol. 8, no. 3, pp. 399–404, 1956.

[77] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," Journal of the ACM (JACM), vol. 35, no. 4, pp. 921–940, 1988.

[78] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[79] C. Harris and M. Stephens, "A combined corner and edge detector," in Fourth Alvey Vision Conference, vol. 15, p. 50, Citeseer, 1988.

[80] B. D. Lucas, T. Kanade, et al., "An iterative image registration technique with an application to stereo vision.," in IJCAI, vol. 81, pp. 674–679, 1981.

[81] C. Tomasi and T. Kanade, "Detection and tracking of point features," tech. rep., School of Computer Science, Carnegie Mellon University, 1991.

[82] J. Shi and C. Tomasi, "Good features to track," in Computer Vision and Pattern Recognition, pp. 593–600, IEEE, 1994.