Robust Automatic Object Detection in a Maritime Environment

Polynomial background estimation and the reduction of false detections by means of classification

M. Hartemink

Master of Science Thesis

TU Delft Delft University of Technology

Media and Knowledge Engineering

Robust Automatic Object Detection in a Maritime Environment

Polynomial background estimation and the reduction of false detections by means of classification

Master of Science Thesis

For the degree of Master of Science in Computer Science - Media and Knowledge Engineering at Delft University of Technology

M. Hartemink

September 11, 2012

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Delft University of Technology

The work in this literature review was supported by the parties shown. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Intelligent Systems (insy)

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) for acceptance a thesis entitled

Robust Automatic Object Detection in a Maritime Environment

by

M. Hartemink

in partial fulfillment of the requirements for the degree of

Master of Science Computer Science - Media and Knowledge Engineering

Dated: September 11, 2012

Supervisor(s):

prof.dr.ir. M.J.T. Reinders

Dr.ir. D.M.J. Tax

Reader(s):

Dr.ir. H.A. Van Der Meiden

Dr.ir. F. Bolderheij

Abstract

Robust automatic detection of surface and air objects in a maritime environment is a problem that is of growing importance to the Royal Netherlands Navy (RNLN). Due to a shift in the field of operation from the open oceans towards the littoral waters, the RNLN is forced to operate in complex environments with cluttered backgrounds and the presence of many small vessels and a wide range of other objects. Traditional radar systems are not optimal in these circumstances due to their minimum detection range, lack of sensitivity to small, non-metallic, objects and poor classification power. Complementation by Electro-Optical (EO) camera systems is therefore desired, which resulted in the start of the development of a detection algorithm based on polynomial background estimation. Automated object detection in the maritime environment is a complex problem however, due to various complicating factors. These factors include the highly dynamic background, camera motion, the variety in possible objects and their appearance, and the diversity in meteorological as well as environmental circumstances. Although the developed detection algorithm is quite well capable of detecting the objects, it also produces an extensive amount of false detections. This study investigates whether these false detections can be eliminated, while maintaining the true detections, by means of classification of the detections as either target or background.

To this end, the initial detection algorithm is optimised to detect as much objects as possible in a carefully constructed dataset of eight hundred Visible Light (VL) images. The resulting detections from the optimised algorithm are used accordingly to train and test various basic classifiers, using a set of features found in the literature. The best performing classifier is selected and the performance of this classifier, and the two-stage detection algorithm as a whole, is subsequently further analysed by means of various tests involving the features used, the evaluation procedure and the fusion of detection results. Results show that especially the features as well as the clustering procedure for detected pixels are important parameters with respect to a good performance of the algorithm.

This works shows that the linear discriminant classifier is best to use with the problem among the classifiers considered. Moreover, it is demonstrated that including features of histogram equalized boxes in combination with features of the entire image increased the performance the most, that determining the features on a slightly larger area than the originally detected area is beneficial and that fusion of detections after classification can be used to optimise

the detector output. Although the developed classification approach is capable of eliminating many false detections and to retain a majority of the true detections, further research is required. Suggested are separate classifiers for the sea- and sky part, inclusion of the time dimension, optimisation of the operating point of the classifier and preprocessing steps.

Table of Contents

Acknowledgements

This thesis report forms the end of what has proven to be an interesting as well as challenging journey. After two years of studying, hard work, but also lots of fun, my time at Delft University of Technology is almost at an end. Without the help and support of the people around me, I would not have made it to this point. First of all I would like to express my gratitude to my supervisors, Dr.ir. D.M.J. Tax and Dr.ir. H.A. Van Der Meiden, for their assistance and guidance during the practical work and writing of this thesis report. Furthermore I would like to thank the co-workers at CAMS-Force Vision, MSc. Dirk Doornenbal, Drs. Niels Mol, MSc. Jasper Priem and Ir. Mark Zijlstra, for their help and the wonderful time I had working at their department. Finally I would like to thank my friends and family for their encouragement along the way and their support throughout my studies in both prosperous as well as hard times.

Delft, University of Technology — M. Hartemink
September 11, 2012

"What you see, yet cannot see over, is as good as infinite."
— *Thomas Carlyle*

Chapter 1

Introduction

Digital video cameras are installed and used in increasing quantities everywhere around us and have become part of our everyday life. In modern society we cannot walk or drive around any more without being captured by some security or surveillance device. Arguably, security and surveillance tasks - such as security of important objects or buildings, traffic monitoring, and crowd surveillance in public areas such as, shopping malls and public transport facilities - are the most common applications of video cameras today. Automated object detection is a critical first step within these applications and has become an important research area in computer vision.

Also within the Royal Netherlands Navy (RNLN) the use of camera systems for security and surveillance purposes has risen and is gaining importance. Radar systems are widely deployed aboard military, as well as commercial, ships and are traditionally the main sensor system for object detection. Unfortunately a radar has limitations on the minimum detectable range, a lack of sensitivity to small, non-metallic, targets and poor classification power. Complementation of these radar systems by a real-time video surveillance system is therefore desired. This has led to fact that Electro-Optical (EO) sensors, such as Visible Light (VL)- and Infrared (IR) cameras, are incorporated in the sensor suites of various ships of the RNLN. At the moment, however, there is a lack of supporting software capable of processing the information originating from the EO camera systems installed aboard the ships. Consequently, the crew is forced to 'process' the images themselves, creating an undesired situation since it requires full attention of an operator who must continuously scan the video stream for possible targets. This results in fatigue of the operator, which in turn increases the probability of errors and leaves less personal available for other important tasks aboard the ship. Hence, an algorithm capable of performing the initial detection of possible targets within the images would be of great help to the ship's crew. This resulted in the aim of the RNLN to obtain knowledge about automatic object detection in the maritime environment and the development of a robust detection algorithm.

1-1 Problem Description

Within a maritime scene there can occur various types of objects for which it is expected that they are detected by an automatic detection algorithm. Examples not solely include surface objects, such as all kinds of vessels, jet-skis, rubber boats, canoes, swimmers, buoys, rocks, etcetera, but also air objects like helicopters, air planes, birds, chaff/flares and so on. These objects are of interest since they might pose a threat to the assets of the RNLN or its allies, or may compromise safe navigation. Whether an object really poses a threat is determined later on during an identification stage, see section 2-1, and is not yet an issue during the stage in which the objects are detected. Furthermore, detection of these objects is important in order to obtain an as detailed and complete operational picture as possible. On the other hand there are also a lot of 'objects' for which it is not expected that they are returned by a detection algorithm, because they are considered to belong to the background. Examples include wake, wave crests, foam, glare, clouds, dunes, coast lines and the structures ashore, etcetera. So, a detection algorithm is only expected to detect objects within a scene that might pose a threat and should disregard objects and phenomena belonging to the background.

Although automatic detection of potentially threatening objects in a maritime environment might seem not that hard at first instance, it turns out to be fairly complicated if one looks in more detail into it. There are a number of factors that complicate the automatic detection process, these factors include:

- **Camera motion**
  First of all, unlike ashore, the camera is not in a fixed position but installed on board a (moving) vessel and may therefore suffer from movement and vibrations due to the moving and/or shaking platform. A detection algorithm should be able to deal with these movements and vibrations.
- **Variety of objects and their appearance**
  The RNLN is expected to operate in a wide range of operational conditions and must be cautious for an extensive range of possible objects. Surface objects not only include various vessels such as: large oil tankers and container ships, small fishing ships, fast rubber boats such as a Rigid Hull Inflatable Boat (RHIB) used by drug smugglers, cabin boats, jet-skis, sailing vessels, and so on, but also non-vessel objects like buoys, rocks and sea mines. Surface objects however are only part of the problem, since also objects in the air like air planes and helicopters are of interest. Furthermore, the orientation, speed, and range of the objects with respect to the camera may take any possible form. Objects may be close to- or far away from the camera and may be seen from all possible angles. In sum this results in an almost infinite variety of possible objects in combination with their occurrence.
- **Highly dynamic background**
  Since the problem involves a maritime environment, the background is highly dynamic and far from stationary due to waves, wake, moving clouds, and illumination changes.
- **Meteorological circumstances**
  The meteorological circumstances can be very different making automated detection more complicated. Rain- and snowfall, fog, clouds, glare caused by the sun and variation in the sea-state are examples of such circumstances.

- **Geographical locations and direction of the camera**
  Also the environment plays an important role in the detection process. A coastline may or may not be present, a coastline may be urban or not, there might be a horizon present in the image or not, the colour of the sea may differ, etcetera. All these aspects might obstruct the automated detection process.

Due to these factors automatic object detection in maritime environment is far too complex for conventional detection methods that are applicable to common indoor, outdoor and traffic scenes. Most often these techniques are either not well capable of detecting the possibly threatening objects or in disregarding the background which results in a large amount of false detections. A basic background extraction technique using a fixed background template as is often used in e.g. traffic monitoring or vehicle detection, will not be suitable in a maritime environment due to the highly dynamic background and the possible camera motion. Even more advanced methods, i.e. methods that use a single Gaussian or a mixture of Gaussians to model the pixel intensity and/or colour, have severe drawbacks in the form of a trade-off problem and are not very suitable to the problem either. Therefore, new, more sophisticated methods, or extensions to existing methods, are required in the maritime domain.

At the Centre for Automation of Mission-Critical Systems (CAMS) - Force Vision, part of the Dutch Department of Defence (DoD), a start has been made with the development of an object detection algorithm based on polynomial background estimation. Although this type of object detection is in theory capable of detecting a wide range of possible targets in various circumstances, it also produces a large amount of false detections. Examples of the output of the algorithm are shown in Figure 1-1. In this figure the objects of interest, which are expected to be detected by the algorithm, are manually annotated with a blue ground truth bounding box. The red bounding boxes are the output resulting from the detection algorithm. Since only a few of the red boxes coincide with a ground truth bounding box, there are only a few true detections and all the other boxes of the systems output are considered false detections. As can be seen the amount of false detections is substantial, and as a consequence, without further improvement in terms of false detection reduction, this object detection approach would be unusable in practice. In previous research, [1], it was found however that classification techniques might be a solution to the false detection problem.

The main research question in this thesis is therefore:

> *Given initial detections resulting from a basic detection algorithm, can a system that learns from examples be used to eliminate false detections while maintaining true detections?*

Ultimately, the insights acquired in this project will have to contribute in the development of a generic and robust automatic object detection algorithm for the maritime domain that can be used with the EO sensors aboard the ships of the RNLN.

1-2 Research Objective, Scope and Methodology

For a robust and generic object detection scheme, the system should ideally have the following properties in order to be of practical use:

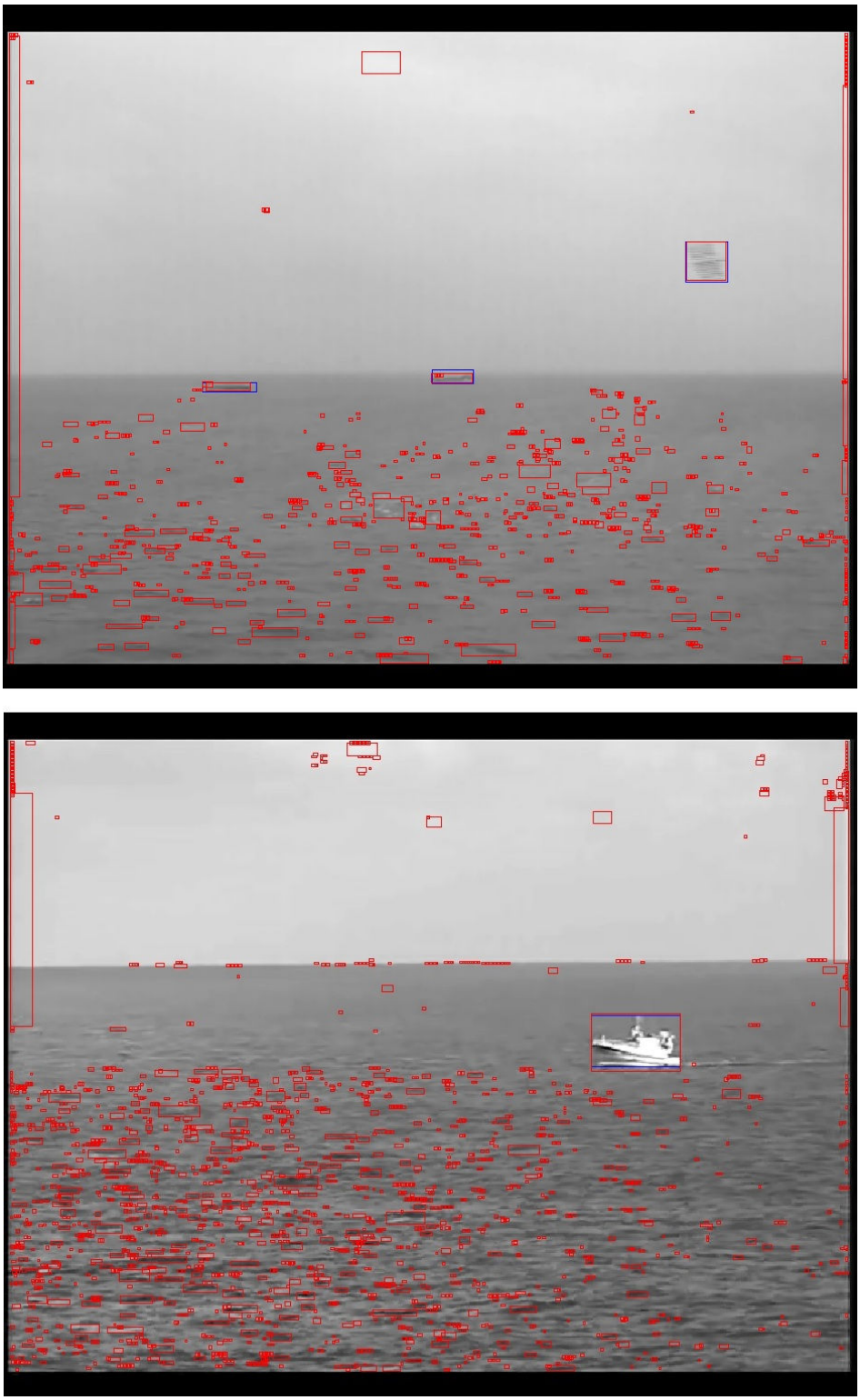


Figure 1-1: The false detection problem: the output of the detection algorithm based on polynomial background estimation (red boxes) contains a large amount of false detections. A detection is marked as a false detection if the bounding box does not coincide with a manually annotated ground truth bounding box (blue box) which encloses an object of interest.

1. It must detect objects of interest within various maritime scenes containing complex backgrounds;
2. It must use as little scene-related assumptions and other prior knowledge as possible;
3. It must produce no missed detections and no false detections;
4. It must operate at a reasonable frame rate and therefore be fast and efficient.

In a previously performed literature study, which will be extensively discussed in the third chapter of this report, it is found that detection algorithms based on polynomial background estimation or gradient filtering show high potential for the maritime domain. These methods can provide a solid base for robust object detection and satisfy the first, second and last requirement as described above. Since at CAMS - Force Vision a polynomial background estimation detection algorithm has been developed, it is chosen to use this approach as a starting point for the work in this project.

As stated the main goal of this thesis project is to examine whether a learning based approach can be used to effectively eliminate false detections while maintaining true detections. The focus within this project will therefore be mainly on the second part of the third requirement, in the form the development of a classification algorithm that classifies the initial detections as either background or target. Of course a classification step combined with the detection step should also comply to the fourth requirement, but this is initially of less importance since at the moment more importance is given to the functionality of the learning based approach rather than to its speed. Even so, by introducing a system that learns from examples the second requirement is compromised, since examples are scene related and can be considered as prior knowledge. In order to reduce this dependence to a minimum, the dataset to be used should contain as much variety in objects and environments as possible.

To guideline the research process and to achieve a satisfying result in the end, the research process is split in four stages.

1. **Construction of a representative dataset.**
   First stage in the research process is to construct a sufficiently large and representative dataset that covers a wide range of scenarios and possible objects that can be encountered in the operational deployment of the RNLN. This is an important and critical step, which must ensure that the detection algorithm and learning approach are actually usable in real life and not only in a particular set of circumstances. In the second research stage this dataset is required for optimisation of the object detection algorithm and in the third stage for the extraction of target and background examples.
2. **Optimisation of the object detection algorithm.**
   As a starting point an object detection algorithm based on polynomial background estimation is used, which is available at CAMS-Force Vision. This initial algorithm will be optimised to detect as many objects as possible within the constructed dataset in order to maximize the detection capabilities (first part of the third requirement; no missed detections) of this detection approach. Since later on the false detections will be eliminated, the amount of false detections is not yet an issue here and it is important to obtain as much 'target' detection samples as possible for the third research stage. Note

that it is not a goal here to improve the detection capabilities of the existing algorithm, but merely to exploit its capabilities to the fullest extent! Result of this stage is a performance baseline of the initial detection algorithm and the availability of a lot of target and background samples which are required in the third research stage.

3. **Development of the classifier.**
   The third stage is dedicated to the development of a classification algorithm which must classify the output of the detection algorithm as either target or background. As a starting point a set of features that showed promising results in existing literature will be used to train various classifiers. Once a classification step has been developed it can be tested together with the detection algorithm. First the detection algorithm, with the parameters found in the second stage, will be applied to the dataset and the initial detection results are classified as either target or background by the trained classifier which is obtained in this stage. The performance of the new two-stage detection scheme can know be evaluated and compared with the baseline. Based on the performance, the best performing classifier is selected for further analysis in the final research stage.

4. **Performance analysis.**
   The emphasis of this work will lie in the final stage, in which the performance of the classification step will be analysed by investigating the influence of various parameters with respect to both the performance of the classifier as well as the object detection system as a whole. Primary goal of this stage is to gather knowledge about the performance of the new two-stage detection process and to identify critical variables of the classification system. Each adjustment will be compared with respect to the baseline obtained in step two and the performance of the initial two-stage detection scheme obtained in stage three.

Of the four steps as described above, the first two steps can be considered as preparatory work for the actual research in the third and fourth stage.

1-3 Overview of Chapters

The remaining chapters of this thesis report are structured as follows. The report continues with an elaborate problem background which underlines the project value in chapter 2. Subsequently, in the third chapter, previously performed research which comprises an evaluation of object detection approaches will be discussed, because it provides the origin and foundation of this project. Chapters 4 to 6 will follow the research strategy as described in the previous section. First the preparatory work is discussed in chapter 4, which concerns the compilation of the dataset and the optimisation of the detection algorithm. Chapter 5 accordingly, is devoted to development of the classification step and the initial results obtained with the new two stage detection approach. In chapter 6 the performance of the classification step will be analysed in order to identify the critical parameters of this approach. Finally we will end with the conclusions and recommendations in chapter 7 of this report.

Chapter 2

Problem Background

Automatic detection of surface and air objects in a maritime environment is a complex problem and one that is of growing importance to the Royal Netherlands Navy (RNLN). During the last decades RNLN field operations have shifted from the open ocean, or so called 'blue water' operations, to coastal areas, or 'brown water' operations. This means that in modern warfare scenarios naval ships must be able to operate not only on the open sea, but also in coastal areas such as bays and narrow straits. Excellent examples of the latter type of operations are the current anti-piracy operations in the Somalia area and the anti-drug operations continuing in the Caribbean. These complex environments, with cluttered littoral backgrounds and many civilian ships, may contain asymmetric threats from targets such as a Rigid Hull Inflatable Boat (RHIB), cabin boats, fishing boats, power boats and other small vessels.

Besides the shift in the field of operation, also naval sensing has changed. Due to advances in sensor technology and the change in the field of operation, sensor management has become increasingly knowledge-intensive. At the same time, the RNLN is faced with a decrease in the amount of available knowledge, both quantitatively and qualitatively, as a result of budget cuts and the policy to reduce crew numbers. This growing discrepancy fuels the need for sensor management automation. To be able to achieve the desired mission results, the tasks of the ship's crew will have to be automated and where this is not possible the crew will have to be sufficiently supported with the deployment of its on-board sensor, weapon and command systems.

Traditionally naval sensing has been primarily radar based. Although radar technology is extremely advanced nowadays, they are not optimal in many scenarios - especially coastal ones - due to their minimum detection range, lack of sensitivity to small, non-metallic, targets and poor classification power. In order to account for this weak spot, Visible Light (VL) and Infrared (IR) sensors are installed and integrated in the sensor suites of the frigates in addition to the classical radar-based sensor suite. Examples of such sensors are the MIRADOR, see Section 2-2, on board the Air Defence and Command Frigate (LCF) of the Zeven Provinciën class and the gatekeeper on board the Oceangoing Patrol Vessel (OPV) of the Holland class. These Electro-Optical (EO) sensor systems can provide valuable complementary information and can contribute to the situational awareness by means of object detection, tracking and

classification. Due to the lack of supporting software however, the information of the EO sensors is now relayed to the operator without further processing. As a consequence, an operator will have to perform detection and classification tasks manually which is both time consuming and might cause operator fatigue. This, combined with the earlier-mentioned crew reduction and the aim to automate processes where possible, has resulted in the demand for an automatic object detection algorithm. In the near future also the submarines of the Walrus class will be equipped with an optronic mast that can be used to make a surface scan with its EO sensors, which underlines the importance for the development of an automatic object detection algorithm.

2-1 The Common Operational Picture (COP) Compilation Process

Achieving situational awareness is one of the most important goals in any military mission and forms the base of the Command and Control (C2) process. To achieve situational awareness all available information, originating from various sensors, is fused into one COP of the area of operation, see Figure 2-1. During the common operational picture compilation, the area is searched for objects and if a COP has already been built, the presence of objects has to be reconfirmed and the information concerning the objects has to be updated. Such detection and reconfirmation is performed by sensors on board the ship in question as well as by sensors of other platforms. Based on the common operational picture and regulations, such as for instance the rules of engagement, it is decided what action needs to be taken. As can be seen in Figure 2-1, the common operational picture forms the heart of the C2 process and allows to make informed and right decisions. It therefore speaks for itself that it is of great importance for the COP to be as informative, detailed and complete as possible.

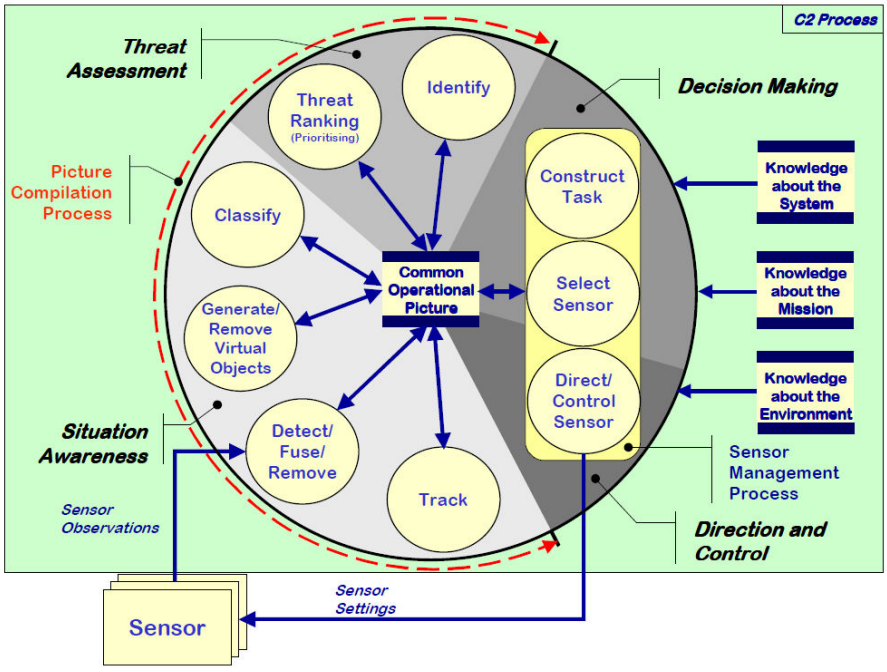


Figure 2-1: The command and control process, [2].

Basically the picture compilation process can be divided into four tasks, which are ordered according to the amount of prior knowledge they require.

1. Search and Detect
   First an object needs to be detected. Therefore, the environment should be continuously scanned for possible objects and if an object has already been detected its presence must be reconfirmed. As outlined earlier, EO sensors are especially a valuable addition for the detection of small, non-metallic, objects that are situated within the field of view of the vessel.

2. Track
   Once an object has been detected, it can be tracked with the available sensors. From this tracking, valuable object properties can be derived, such as its speed, position, heading, etc. This information can help to recognize, classify and identify the object later on.

3. Classification
   During the classification stage the type of object is determined. Classification can be done at a global level, e.g. is the object military or civil, or at a more detailed level e.g. is the object a RHIB or a cabin boat. Preferably the object is classified in as much detail as possible. VL and IR information provides a powerful cue in classification. Images of objects can be matched with known types of vessels/objects and with infrared profiles. Furthermore, EO sensors give a far more conclusive classification result than can be obtained with a radar system, since the Radar Cross-Section (RCS) is not unique to an object and depends strongly on the angle at which the object is seen. Results obtained during tracking of the object in the previous stage can be of great help in making a global classification and narrowing down possibilities of what the object can be. If, for example, a surface object is detected and during tracking it turns out to move with a velocity of 40 knots and has recently made a sharp turn, it is unlikely that the object is a large oil tanker or container ship. This way a rough classification could be deduced.

4. Identification
   During the identification phase the object is categorized in one of the identity categories. These categories are: friendly, neutral, unknown, suspect and hostile. Initially, a detected object is categorized as unknown. Based on additional available information, i.e. tracking behaviour, classification results, Identification Friend or Foe (IFF) information, the COP and prior intelligence information, the identification is updated into one of the other categories if possible and necessary.

Both VL and IR sensors are a valuable addition to the operational picture compilation process since can be used in all four stages. As already has been stated, they are capable of detecting and tracking objects which cannot be detected e.g. by radar systems and are especially well suited to the detailed classification and identification of objects. A robust object detection algorithm is therefore of great importance. However, due to the complicating factors as mentioned in the introduction and the area of operation of the RNLN we are currently facing a fairly complicated problem.

2-2 The MIRADOR Electro-Optical Sensor System

This section gives a brief introduction to the mirador EO sensor system, because it is the RNLN's aim to use the detection algorithm with this system at first. Later on it is intended to be used with the optronic mast aboard the submarines of the Walrus class as well. However, since the optronic mast with its sensors is not yet available we will focus on the mirador system aboard the LCFs. The MIRADOR electro-optical sensor system, see Figure 2-2, comprises three visible light TV cameras, an infrared camera and a laser range finder.

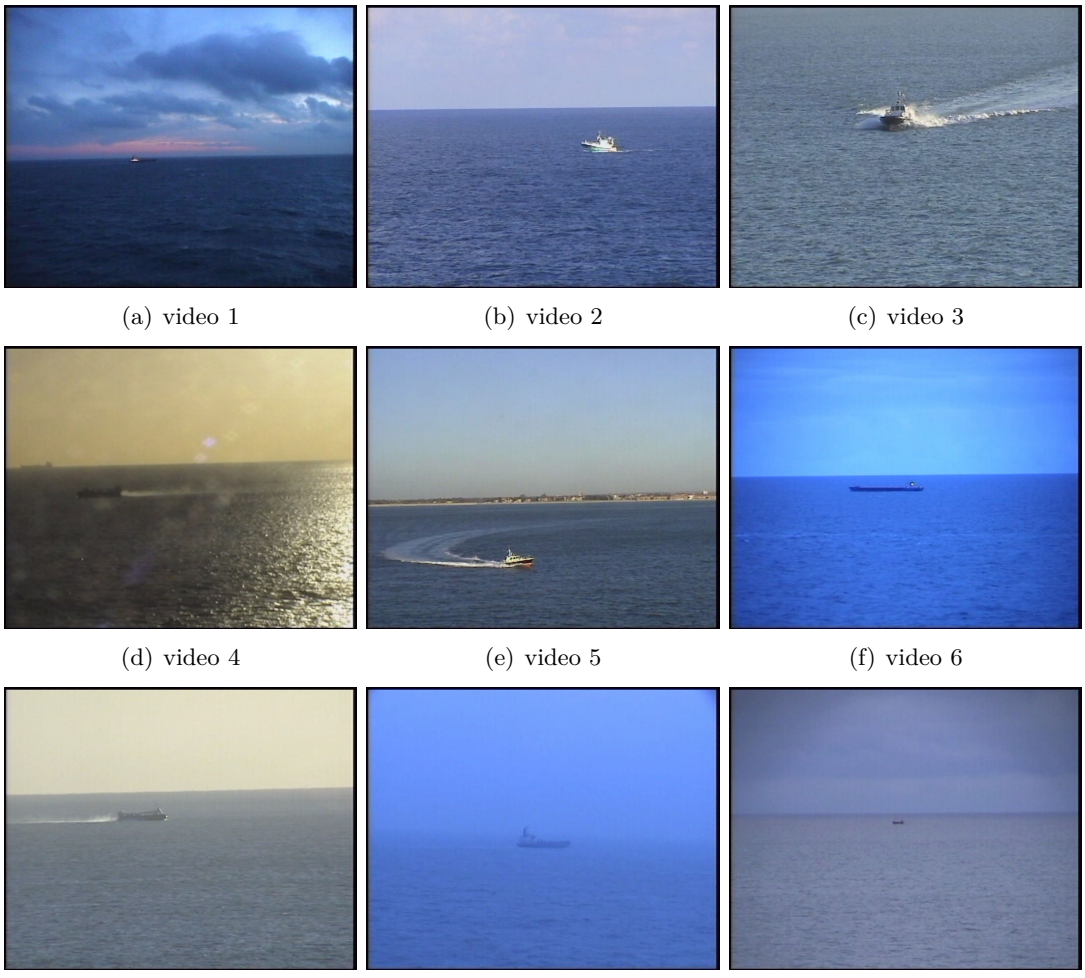
Of the three visible light cameras, one is a colour TV camera that can be used during daylight for surveillance. The second camera is a long range TV camera, with a fixed field of view that can be used to track surface and air targets during daylight and the third one is a low light-level TV camera that can be used during dusk for surveillance. Together with the infrared camera the mirador comprises four cameras, however only two of them can be operated at the same time. As can be seen in Figure 2-2, all cameras are installed in the same cabinet which is self-stabilizing. Since the cameras cannot be moved independently they all have the same viewing direction.

Figure 2-2: EO sensor system MIRADOR, [3].

2-3 Available Data

At the Centre for Automation of Mission-Critical Systems (CAMS) - Force Vision, which is the software company for the Dutch Department of Defence (DoD) and especially the RNLN, many hours of VL as well as IR videos are available. These videos were recorded with the MIRADOR on Her Netherlands Majesty (HNLMS) Tromp in the fall of 2007 and contain various types of objects recorded in different meteorological and environmental circumstances. The size of the images originating from the colour/low light-level TV camera is 720x576 pixels and the size of the images originating from the infrared TV camera is 512x512 pixels. From the available data it should be possible to create a challenging dataset that covers a wide range of possible operational circumstances. The compilation of the final dataset from the available data will be discussed in Chapter 4.

To illustrate what kind of footage is available, a couple of frames from different videos captured with the colour TV camera are shown in Figure 2-3.



(a) video 1 (b) video 2 (c) video 3

(d) video 4 (e) video 5 (f) video 6

(g) video 7 (h) video 8 (i) video 9

Figure 2-3: Impression of the available visible light images

Chapter 3

Existing Detection Approaches

In previous research [1], various existing literature with respect to object detection has been reviewed in order to identify techniques that show high potential for the maritime domain as the Royal Netherlands Navy (RNLN) encounters it. In this work several object detection approaches are evaluated, in which a distinction is made between static and adaptive approaches, and is based on the same requirements as reported Section 1-2. This chapter provides a synopsis of the techniques discussed in the afore mentioned research, including their advantages and disadvantages, and ends with its main conclusion which provides the starting point of the research in this project.

3-1 Static Approaches

Static object detection approaches are generic approaches that do not require any prior knowledge and are applicable on single frames without the need of using previous frames. Most techniques are directly applicable although some require a training phase before they can be applied. Detection approaches discussed in this category are gradient based methods, non-adaptive background estimation methods, target-background classification and Maximum Average Correlation Height (MACH) filter template matching.

3-1-1 Gradient Based Approaches

Gradient based methods most often firstly apply some kind of smoothing filter in order to reduce the amount of noise and secondly apply some kind of gradient filter, such as the Sobel or Prewitt filter, to detect edges within the image ([4],[5],[6]). After the image is filtered with the gradient filter, the result is thresholded and it is tried to connect the edges by means of dilation which must fill the gaps that are likely to occur. After dilation, the connected edges are segmented in several Region(s) Of Interest (ROI) which are represented by bounding boxes. The process as described above is illustrated in Figure 3-1.

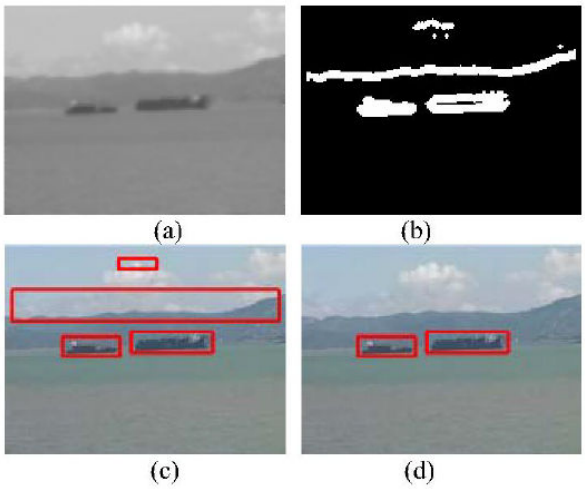


Figure 3-1: (a) Example of filtered image. (b) The extracted edges. (c) segmentation in ROI's. (d) Moving maritime object detection result, [4].

Advantages of gradient based methods are that they are simple, fast, highly generic and capable of detecting both stationary as well as moving objects. Downside of this approach is that there also occur lots of false detections. After generation of the bounding boxes enclosing the regions of interest in most literature there often follows a step that must reduce the amount of false detections. In the example of Figure 3-1 (d) it is determined by means of temporal differencing which boxes are stationary and which boxes are moving. The boxes that are labelled as non-moving boxes are eliminated [4]. Besides temporal differencing also multi-histogram matching is used to distinguish between boxes that are actually an object and which boxes are background [6].

3-1-2 Static Background Estimation Approaches

In non-adaptive background estimation approaches, it is tried to model the intensity of the background by means of polynomials or a regression model. In both cases the estimated background image is subtracted from the original image and the resulting difference/residue image is thresholded subsequently to identify possible objects, see Equation (3-2). The assumption is that the background can be modelled well, while objects present in the image cannot. This discrepancy results in a large difference or high residue.

In the work of [7] a regression model is used to obtain an estimate of the background intensity values. In this work first a horizon detection step is executed in order to isolate the sea part for modelling. If no horizon is detected, the entire image is used for modelling of the sea background. At low resolution, water regions are rather homogeneous as waves and other sea clutter are smoothed over. Hence, the intensity of the water pixels can be estimated using a regression model with respect to the pixels's coordinates, (x, y), which results in an estimation of the background, $\tilde{I}$:

$$\tilde{I}(x, y) = ax + by + c \tag{3-1}$$

To obtain an accurate estimate of a, b and c robustly to non-water objects that might appear below the horizon line, the weighted least squares error function is iteratively minimized:

$$\min_{a,b,c} \sum_{(x_i,y_i)\in\Omega} w_i \left(\underbrace{I(x_i,y_i) - \tilde{I}(x_i,y_i)}_{r_i} \right)^2 \tag{3-2}$$

where the summation is taken over Ω, the set of all pixels (x_i, y_i) below the estimated horizon line in the original image I. Initially, the weights w_i are set equal. At the next iterations, w_i are updated as follows:

$$w_i = \begin{cases} (1 - r_i^2)^2 & |r_i| \leq T \\ 0 & |r_i| \geq T \end{cases} \tag{3-3}$$

where r_i is the residue at pixel (x_i, y_i), and T is a pre-defined threshold. In this way, the non-water pixels which have a high residue will not contribute to the resulting estimates of the model parameters. To reduce the presence of sea clutter the residue image, see Equation(3-2), is filtered with a smoothing filter and potential objects are detected by thresholding the smoothed residue image.

Besides a linear model also a higher order model may be used to obtain an estimate of the background. Since this approach is used in this work as well, it is not discussed in this section. Instead, a full description of this approach is given in Section 4-2 of this report.

Advantages of non-adaptive background estimation approaches are that they are simple, fast, highly generic and capable of detecting both stationary and moving objects. A huge drawback of these approaches, however, is that they cause a lot of false detections. Furthermore, it is expected that these approaches will have a lack of sensitivity to low-contrast scenarios as possible objects will almost blend into to the background in terms of intensity difference.

3-1-3 Target-Background Classification

The techniques discussed so far, directly use the intensity difference between the object and the surrounding background in order to detect the objects in the image. To detect the objects, either the gradient is used or the background is estimated and compared with the intensities of the input image. A novel approach, which is pattern recognition based rather than image processing based, described in [8], is the target-background classification approach.

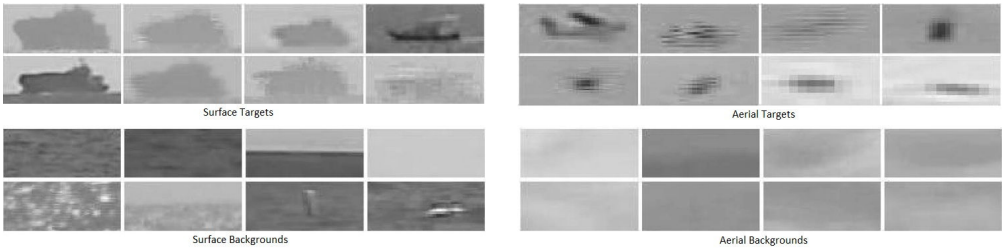


Figure 3-2: Examples of target and background images, [8].

In the target-background classification approach regions of interest which might be target or background, as shown in Figure 3-2, are classified as either target or background by a classifier which is trained beforehand using comparable images. Reported test results with a Support Vector Machine (SVM) classifier are very high and in the order of 99.6-99.8%. Unfortunately, the region(s) of interest is/are not determined automatically but must be provided by hand. This method is therefore not directly suitable for automatic object detection, but might very well be used to eliminate false detections or to track earlier detected objects. Techniques used in this approach will extensively be discussed in Chapter 5 of this report.

3-1-4 MACH Filter Template Matching

The last static approach discussed is the MACH filter template matching approach. In the work of [9] this approach is used for a visual surveillance system at a maritime port facility. The idea of MACH filtering is to create a general template for each class of possible objects and to correlate all these object templates with the input image to determine whether an object of the corresponding class is present or not. First a dataset of training images is constructed containing several images of the most common vessel types occurring in the area of the port, see Figure 3-3.

Figure 3-3: Examples of the set of common vessels, these include: container ships (a), speed boats (b), tanker ships (c), tugboats (d), cruise ships (e) and fishing boats (f), [9].

Given a series of instances of a class, a MACH filter combines the training images into a single composite template by optimizing four performance metrics: the Average Correlation Height (ACH), the Average Correlation Energy (ACE), the Average Similarity Measure (ASM), and the Output Noise Variance (ONV). This process results in a two-dimensional template that may express the general shape or appearance of an object. This process is shown in Figure 3-4.

Once the templates are obtained the objects in frames of the surveillance videos can be detected by performing a cross-correlation in the frequency domain. The highest peak corresponds to the most likely location of the object in the frame. Test results have shown that this approach is quite successful in the maritime port setting. However, this approach is infeasible for use in the various operational scenarios of the RNLN. First of all, the amount of different objects that might be present is enormous, requiring a huge dataset with training images. Secondly, this approach is very sensitive to object size and appearance. The method only works if the viewing angle of the objects is always approximately the same, as well as their

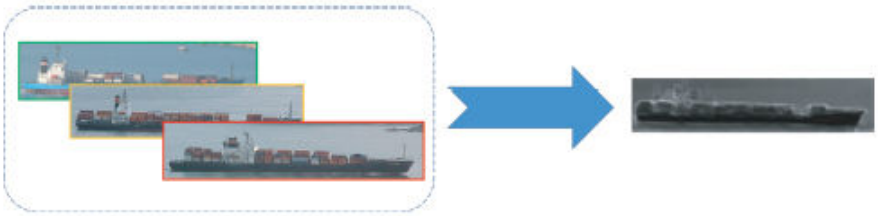


Figure 3-4: MACH filters combine a collection of training images (left) into a single composite template (right) by optimizing a set of metrics, [9].

size in the images. These constraints do not hold for the RNLN scenarios. If it would be tried to use this method, this would mean that for each size and/or appearance a template should be created, which would result in a massive dataset and processing times that will be too high, making real time application impossible. All in all, it is concluded that this approach is not suitable for the dynamic scenarios in which the Dutch navy has to operate.

3-2 Adaptive Approaches

Opposed to the static background estimation techniques discussed in the previous section are the non-recursive and recursive background estimation techniques. A lot of these techniques can be found in literature concerning visual applications, such as urban traffic monitoring and visual detection of pedestrians at public locations. A comprehensive overview and discussion of these techniques is provided in [10]. The techniques discussed in this section differ from the static background estimation techniques in the way that the scene is modelled over a time series instead of independently per frame.

3-2-1 Non-Recursive Background Estimation Techniques

A non-recursive background estimation technique uses a sliding-window approach for background estimation. It stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each pixel within the buffer. A couple of possible techniques to use the temporal variation are listed and discussed below.

Frame differencing
Frame differencing, or also temporal differencing, is the simplest form of non-recursive background estimation. It uses the frame at time $t-1$ as the background model for the frame at time t. This technique can be used to detect moving objects, but is very sensitive to noise, variations in illumination and camera movements. Also this method would fail to detect an object if it would stop moving. Since it only uses a single previous frame, pixels inside a large uniformly coloured object may not be identified as a detection as well. This problem however can be accounted for with a proper clustering algorithm of the detected pixels.

Average filtering
In average filtering the average of all frames in the buffer is taken as an estimate of the background. If the scene is static the average will be similar to the current frame, except where motion occurs. However, this technique is not robust to scenes with many moving

objects, particularly if they move slowly. Furthermore, this method is not robust to drastic changes in the scene within the frames of the buffer such as changing lighting conditions or movement of the background. Exponential weighting of the frames might reduce the influence of these phenomena, but does not overcome the problem completely. Another obvious problem with this technique is that pixels coming from both background and foreground are used to update the background model. A solution is that only pixels not identified as moving objects are used to update the background.

Median filtering
Median filtering defines the background to be the median at each pixel location of all the frames in the buffer. It is hereby assumed that the pixel stays in the background for at least more than half of the frames in the buffer. This approach has also been extended to colour images by replacing the median with the medoid. Weakness of this approach is the assumption that a pixel stays in the background for more than half of the frames in the buffer. There are many scenarios in which this would not be the case.

Minimum-Maximum filtering
In minimum-maximum filtering, three values are estimated for each pixel during a training sequence without foreground objects: minimum intensity, maximum intensity and the maximum intensity difference between consecutive frames. These values are calculated over several frames and are periodically updated for background regions. As an estimation of the background the previous frame is used. Several options are possible to define detections, the intensity is outside the range spanned by the minimum and maximum value determined in the learning phase, the current intensity difference with the previous frame is larger than the maximum intensity difference stored, or combinations of both. Although this technique is capable of detecting both moving as well as stationary objects, it requires a learning phase without foreground objects which might be hard or impossible.

Linear predictive filter
A linear predictive filter, such as a Wiener or Kalman filter, can be used to compute the background estimate by applying the filter on the pixels in the buffer to predict a pixel's current value from a linear combination of its previous values. Pixels whose prediction error is several times worse than the expected error are classified as foreground pixels. The filter coefficients are estimated at each frame time based on the sample covariance, making this technique difficult to apply in real-time. Also autoregression and principal component analysis can be used to predict the frame to be observed.

In general, non-recursive techniques are highly adaptive as they do not depend on the history beyond the frames stored in the buffer. On the other hand, the storage requirements can be significant if a large buffer size is needed to cope with slow-moving objects and will be hard to detect with non-recursive background estimation techniques. Even so, objects that do not move cannot be detected at all. A serious constraint, since for our application it is often required to detect not only moving objects but also those that do not. Other disadvantages are that some techniques require a learning phase without foreground objects and that all techniques have serious performance exacerbation if the background is not stationary, or nominal stationary, in consecutive frames of the video. A huge drawback, since the camera is installed aboard a platform which is most of the time moving and where also the environment itself is highly dynamic (waves, lighting changes, etc.). Therefore, it is expected that these

methods are not suitable for the Dutch navy, due to low performance caused by the fact that the practical constraints will not be satisfied.

3-2-2 Recursive Background Estimation Techniques

Recursive techniques, opposed to non-recursive techniques, do not maintain a buffer for background estimation. Instead, they recursively update either a single or multiple background model(s) based on each input frame. As a result input frames from distant past could have an effect on the current background model. Compared with non-recursive techniques, recursive techniques require less storage space, but errors in the background model will remain in the model for some time. Most schemes include exponential weighting in order to decrease the importance of distant frames while increasing the importance of recent frames. Also positive decision feedback is included in most schemes to only use background pixels for updating. Some popular recursive techniques are listed and discussed below:

Approximated median filter
Instead of the non-recursive median filtering method, the median can also be recursively estimated. A running estimate of the median is incremented by one if the input pixel is larger than the estimate and decreased by one if smaller. This estimate eventually converges to a value for which half of the input pixels are larger than this value and the other half is smaller. In other words, the value converges to the median. Drawback of the approximated median is that it slowly adapts to large changes in that it needs many frames to learn the background region if an object starts moving after being stationary.

Single Gaussian
In the single Gaussian approach, the background is estimated by assuming that the intensity distribution of each pixel can be modelled as a Gaussian distribution. To this end, for each pixel its mean value, μ, and variance, σ^2, are determined and updated. This basic Gaussian model can adapt to slow changes in the scene (i.e. global illumination changes) by recursively updating the model with new input frames. Besides intensity, this method can also easily be extended to colour images to model the pixels colour components as single Gaussian distributions. Downsides of this approach are that the background might be multi-modal in which case a single Gaussian would not be sufficient, and that it is not robust to sudden changes in the environment (e.g. wavering trees, sudden illumination changes, white foamy wave crests, etc.). Furthermore, objects need to be at least slowly moving to be detectable with this method. As stated earlier, this is a downside since in our scenario we are also interested in stationary objects.

Mixture of Gaussians
In case the background of the scene contains many non-static objects such as tree branches and bushes whose movement depends on the wind, a single Gaussian assumption for the Probability Density Function (PDF) of the pixel intensity would not hold. The pixel intensity values vary significantly with time due to the motion. Instead of a single Gaussian, a Mixture of Gaussians (MoG) is used to model such type of variations. In a MoG approach the intensity of a pixel can be modelled by a mixture of K separate Gaussian distributions. If a pixel's intensity distribution is modelled this way, every pixel value is compared against the existing set of models at that location to find a match. A match is defined as a pixel that falls within, for example 2, standard deviations of a distribution. The parameters and weight for the

model are updated based on a learning factor. If there is no match, the least likely model is discarded and replaced by a new Gaussian with statistics initialized by the current pixel value. The models that account for some predefined fraction of the recent data are marked as background and the rest as foreground.

Also MoG approaches have drawbacks. Besides that these approaches are computationally intensive and their parameters require careful tuning, they are not robust to sudden changes in global illumination. Furthermore, this type of approach requires a training stage without foreground objects, which is hard or impossible in the scenarios encountered by the RNLN, since we cannot control the environment. Advantage of this method is that it is robust to some extent against changes in the background. In this respect, MoG approaches are facing trade-off problems depending on the learning rate to adapt to these changes. For a low learning rate, it produces a wide and inaccurate model which will have low detection sensitivity and poor background adaptation properties. A high learning rate on the other hand will result in quick background adaptation, but stationary or slow moving objects will be absorbed into the background model, which causes false negatives. Therefore, a compromising set of parameters has to be identified in order for this approach to perform well.

In the work of [11], the single Gaussian approach is adopted and tested in a maritime surveillance scenario with a stationary pan, tilt, zoom camera. In the maritime environment, the most prevalent background is the sea or ocean. Therefore, the choice of an appropriate background subtraction method depends greatly on the PDF of the ocean pixels. The authors first argue that based on examination of histograms of randomly selected ocean pixels the majority of ocean pixels are unimodally distributed. This hypothesis is confirmed with Hartigan's Dip test using 10 different ocean pixel observations over 800 frames of each of the sequences. Although not all tests allowed to reject the hypothesis, the authors initially adopted the single Gaussian approach.

Conclusions are that the single Gaussian approach can be used for the detection of objects, and small objects, but that in case the contrast between the ocean and target becomes poor this method will fail to detect the motion of the object and therefore the object itself. Lowering the threshold for the foreground/background decision would be a solution but would produce a significant amount of false alarms. Furthermore, glint and white foam of waves are causing many false detections with this method. An attempt to model the scenes with a mixture of Gaussian's did not improve the results significantly. As a reason the authors mention that the camera did not look at the same scene for an extended amount of time resulting in a too short time span to build the bimodal distribution.

In sum it is concluded that the recursive approaches discussed in this section are also not the ideal solution to robust object detection for the navy. Some approaches discussed are reasonably capable of dealing with changes in the background, which is a necessity, but these methods still require the camera to be stationary, or nominal stationary, and are not well capable of detecting slow moving objects and stationary objects. Although there exist background subtraction techniques for moving cameras ([12],[13]), these techniques are unfortunately unusable due to the fact that the background is required to be static and the objects of interest to be moving.

3-3 Conclusion

In the end it is concluded that the gradient and non-adaptive background estimation methods show the highest potential for object detection in the maritime domain since they satisfy most of the requirements. They are fast and straight-forward, do not require any prior knowledge, are applicable to both Visible Light (VL) and Infrared (IR) images and are capable of dealing with camera motion as well as both stationary and moving objects. Unfortunately, these methods produce large amounts of false detections. This problem is acknowledged and marked for further research in [1]. To this end classification of the initial detections by means of a classifier is suggested.

Chapter 4

Preparatory Work

In this chapter the required preparatory work for the research within this thesis project will be discussed. First topic is the construction of a solid and representative dataset that should form the base for the rest of the research. Secondly the detection algorithm based on polynomial background estimation and its principles will be discussed, followed by an overview of metrics that will be used to evaluate the performance of the object detection algorithm. Finally this chapter ends with the tests, and their results, that are performed to optimise the detection algorithm with respect to the amount of objects it can detect.

4-1 Construction of the Dataset

Before the existing detection algorithm can be optimised and a system that learns from examples can be developed, a dataset with images is required for testing and the extraction of target/background examples. The construction of this dataset is an important step which must ensure that the detection- and classification algorithm are generally applicable and not only in a particular set of circumstances. Therefore, the dataset must be sufficiently large and it must cover as many different operational scenarios and targets as possible. For the construction of the dataset the available recordings of the mirador sensor, as discussed in Section 2-3, are used. In total 11 DVD's with 435 PAL-videos are available. Since the majority of these videos concern Visible Light (VL) videos, it is chosen to construct a VL dataset and to disregard the infrared videos.

Since all testing- and programming work will be executed in Matlab®, the VL videos are first converted from DVD format (.vob) to .avi files. A necessary step, because Matlab does not support .vob files. From the .avi videos, useful frames can be extracted to include in the dataset. Manually scanning all the videos for useful frames would be too time consuming however. In order to get an overview of the contents of the videos more quickly and to use as much of the available material as possible, from each video a hundred frames are extracted linearly over the duration of the video. Result is an initial dataset containing nearly forty thousand images. This initial amount is subsequently narrowed down to a final, representative,

dataset of eight hundred images. During selection of the images for the final dataset there is paid attention to the following properties:

- **Variation in type of objects and their size**
  It is tried to capture as much different objects that are of interest for detection. Included in the final dataset are various military and civil vessels, varying from frigates, submarines and replenishment ships to oil tankers, container ships, ferries, rubber boats, cabin boats, power boats, sailing vessels, etcetera. Besides vessels also many other objects are present, such as buoys, smoke markers, birds and rocks. Furthermore, the size, appearance and position of the objects within the images are very different. Sometimes the object of interest is not even completely within the image but only partly. Finally, the amount of objects that are of interest in an image is not kept fixed, but instead may vary from zero, to one or multiple objects as reported in Table 4-1. All objects of interest for detection are manually annotated with a Bounding Box (BB) that encloses an object of interest properly, and which is referred to as the Ground Truth (GT) in the remainder of this report.

Table 4-1: The amount of objects that may occur in an image versus the amount of images in the final dataset that contain these numbers of objects.

# of Objects	# of Images
0	134
1	435
2	159
3	39
4	16
5	15
6	2

- **Variation in meteorological and geographical circumstances**
  It is tried to capture as many different meteorological and geographical circumstances as possible. Included are images with low contrast due to fog, images captured during rainfall and with rain drops on the lens, in sunny weather causing lots of glare, in cloudy weather, etcetera. Some images are even captured during dusk, so visibility is low. Furthermore, attention has been paid to the geographical circumstances and/or direction of the camera. Images may be captured on the open ocean, and contain a horizon or not, may be captured in coastal areas, and contain a coastline or not or may even be captured in a narrow passageway to a harbour. All circumstances that are representative for the field of operation of the Royal Netherlands Navy (RNLN) and must therefore be included in the final dataset.

The result is a representative dataset with respect to the operational circumstances of the RNLN and with a large amount of variation. This variation, in objects as well as meteorological and geographical circumstances, is illustrated in both Figure 2-3 and Figure 4-1. Furthermore, the general properties of the final dataset are listed in Table 4-2. Due to its size and variation, the final dataset provides a solid base for the research in the rest of this project.

Table 4-2: General properties of the final dataset.

Description	Value
Total Number of Images	800
Number of Colour Images (colour TV camera)	668
Number of Grey-Scale Images (low-light level TV camera)	132
Total Number of Objects	1021
Image Dimensions (WxH, pixels)	720x576
Size of Smallest Object (in pixels)	15
Dimensions of Smallest Object (WxH, pixels)	3x5
Size of Biggest Object (in pixels)	164480
Dimensions of Biggest Object (WxH, pixels)	640x257

Figure 4-1: Impression of the final dataset and the variety in circumstances and objects

4-2 Detection Principles

In [5], a detection algorithm based on polynomial background estimation is described, and its principles will be used in this work as well. This detection approach operates on a gray-scale input image, $I(x, y)$, and tries to estimate the background by means of polynomials. To this end, a polynomial is fitted to the intensity values of each column of the image:

$$\tilde{I}(x) = \sum_{n=0}^{N_{max}} a_n x^n \tag{4-1}$$

where N_{max} is the order of the polynomial and $n = 0, \ldots, N_{max}$, a_n are the polynomial coefficients and $\tilde{I}(x)$ are the estimated intensity values by the polynomial. Result is an estimated background image, $\tilde{I}(x, y)$, which comprises the estimated intensity values of the polynomial fits per column. Since it is likely that the background covers most pixels, it is expected that objects cannot be covered well by the polynomials and that they will differ from the estimated intensity values. Objects can therefore be detected by subtraction of the original image with the estimated background image:

$$I_{diff}(x, y) = |I(x, y) - \tilde{I}(x, y)| \tag{4-2}$$

and by thresholding the resulting difference image $I_{diff}(x, y)$.

For the threshold the mean, $\mu(y)$, and standard deviation, $\sigma(y)$, are determined per row of the absolute difference image. Using these statistical parameters, an upper and lower threshold, T_{high} and T_{low}, are defined as follows:

$$T_{high}(y) = \mu\Big(I_{diff}(y)\Big) + \alpha \cdot \sigma\Big(I_{diff}(y)\Big) \tag{4-3a}$$

$$T_{low}(y) = \mu\Big(I_{diff}(y)\Big) - \alpha \cdot \sigma\Big(I_{diff}(y)\Big) \tag{4-3b}$$

where: α is a predefined threshold parameter.

In case a difference value, $I_{diff}(x, y)$,exceeds one of the thresholds as defined above, detections occur:

$$D(x, y) \Rightarrow I_{diff}(x, y) > T_{high}(y) \vee I_{diff}(x, y) < T_{high}(y) \tag{4-4}$$

Result is a logical matrix, $D(x, y)$, which indicates for each pixel in the image whether the pixel is marked as a detection or not. Finally the detected pixels in vicinity of $\pm\delta$ of each other are clustered into bounding boxes enclosing the detected area. Result is a five step detection algorithm, as visualized in Figure 4-2. A full description of the practical implementation of this algorithm is included in Appendix A of this report.

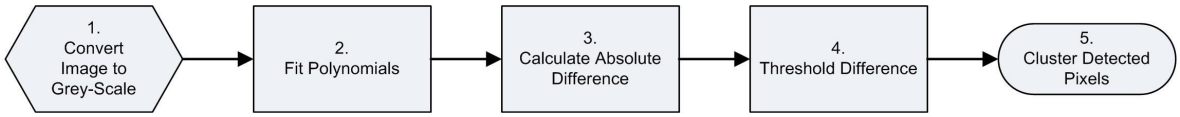


Figure 4-2: Work flow of the detection algorithm based on polynomial background estimation.

4-3 Performance Evaluation of the Detection Algorithm

Before optimisation of the object detection algorithm, the evaluation measure has to be defined. In all literature reviewed prior to this work ([14], [15], [5], [16], [9], [17], [8], [18], [19], [20], [21], [22], [11]), the corresponding detection algorithms are evaluated by examining the spatial overlap between the manually annotated Ground Truth (GT), which encloses an object of interest properly, and the Bounding Boxes (BBs) produced by the detection algorithm, see Figure 4-3.

Figure 4-3: Illustration of a ground truth (GT, blue rectangle) and a bounding box produced by a detection algorithm (BB, red rectangle).

In this work the most commonly used metrics are adopted, which are discussed in the remainder of this section. Here a distinction is made between basic metrics, which are determined per frame, and global metrics which are determined over series of images. Here the basic metrics are used to calculate the global metrics. Advantages of the metrics used are that they are also common in the field of radar, which makes it possible to compare the performance of a camera surveillance system with that of a radar, and that they are common in the field of classification as well, which allows to stick to the same terminology throughout this work.

4-3-1 Basic Metrics

In nearly all pieces of literature at least four basic metrics based on the spatial overlap between the BB and the GT are used for performance evaluation in a single frame. These are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN):

- **True Positive (TP):** Both the GT and the systems output agree on the presence of an object. The BB of the systems output coincides/intersects with the GT of an object.
- **True Negative (TN):** Both GT and systems output agree on the absence of an object.
- **False Positive (FP):** Some, or all, pixels in the BB of the systems output are not covered by the GT of an object.

- **False Negative (FN):** There exists a GT, but the systems output does not provide a BB that coincides/intersects with this GT.

The performance metrics above are useful to describe the performance of the algorithm in terms of how many objects the algorithm has successfully detected, how many objects are missed and how many false detections occur in a single frame. The definitions above however, cannot be used directly. If, for example, a BB just overlaps one pixel with a GT it would already count as a true positive, which would not make sense. An additional constraint is therefore needed to overcome this problem. In the literature such a constraint differs. Some use the simple constraint that the centroid of the BB must be inside the GT, such as in[19]. This is quite a poor constraint however, since it only works reasonably well if the GT and overlapping BB are of similar size. For very small bounding boxes (i.e. a few pixels) the problem remains. A better constraint, used in [5], exploits the spatial overlap between the BB and GT. Here the area of the intersection between the BB and the GT divided by the area of the GT must exceed a certain threshold before the result is counted as a TP. Unfortunately also this constraint has a severe limitation, since it allows a BB to be much bigger than the GT if it covers a sufficiently large area of the GT. This is undesired and therefore this constraint is rejected as well. The best and most commonly used constraint that has been found, is the constraint as reported in [18] and [23]. Here the intersection between the GT and the BB, divided by the union of the two bounding boxes, must exceed a predefined threshold:

$$\text{Overlap Ratio} = \frac{GT \cap BB}{GT \cup BB} > \text{Overlap Threshold (OT)} \tag{4-5}$$

This constraint sets both a limit on the area of the ground truth which must minimally be detected, as well as a limit on the minimum and maximum size of the bounding box relative to the ground truth that is considered. Therefore, this constraint is considered best to use, and adopted in this work as well.

If the detection outcome is compared with the ground truth, there are several possibilities with respect to the spatial overlap of the ground truth with the bounding boxes resulting from the algorithm. These possibilities are shown in Figure 4-4 and will be used to illustrate how the basic performance measures are determined.

From this figure it follows that the basic performance measures are only clearly defined for the situations as depicted in sub figures (a) till (d). If there is a GT and a detection which have an overlap ratio that is large enough, the detection would count as a true positive (a). However, if the overlap ratio is not large enough the detection would count as a false positive. Even so, if there is a detection but no ground truth that coincides with it, this detection would count as a false positive (b). If there is no ground truth (object) present in the image and if there are no detections resulting from the algorithm, this would count as a true negative (c). Finally, if there is a ground truth, but no detection or a detection with an overlap ratio that is not large enough, this situation would count as a false negative (d). In these situations the definitions of the basic performance measures are unambiguous. However, there might also occur situations in which the definitions are ambiguous and it should be elucidated how the definitions are applied in those situations. These situations occur when a single ground truth is covered by multiple detections (e), or in which multiple ground truths are covered by a single detection (e).

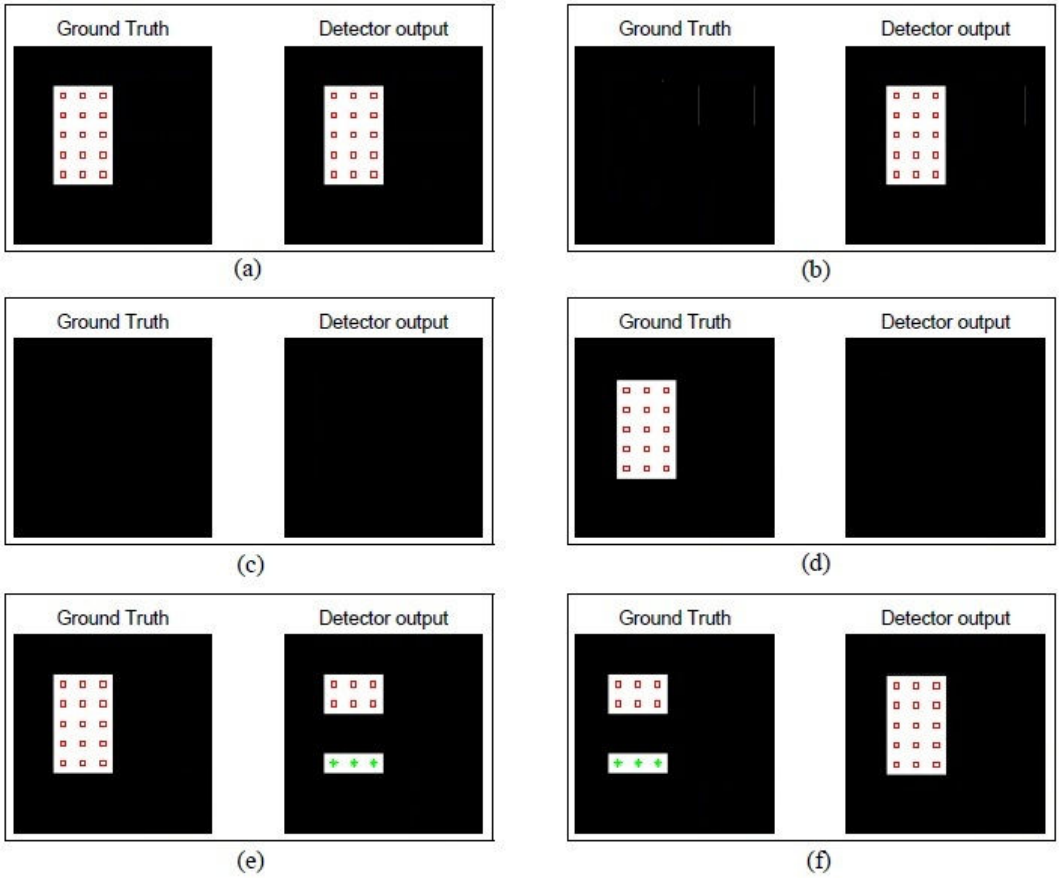


Figure 4-4: Different possible detection outcomes: (a) TP (b) FP (c) TN (d) FN (e) ?? (f) ??. Original image taken from [18] and edited for correct illustration in this work.

In this work these situations are handled as follows. In the first case, it is checked whether there are bounding boxes with an overlap ratio that exceed the predefined threshold. If this is the case, the detection with the largest overlap ratio will count as a true positive. Since ideally an object is covered by a single bounding box, it is chosen to count all the other detections as false positives. So, if in example it is assumed that the detection with the red squares in (e) has a sufficiently large overlap ratio, this detection would count as a true positive and the detection with the green pluses as a false positive, even if this detection possesses a large enough overlap ratio. In case multiple ground truths are covered by a single detection, it is checked for each ground truth whether the overlap ratio exceeds the predefined threshold. Depending on the result, the detection would count for each ground truth object as a true or false positive. So for the example provided in (f) the detection might either count as two true positives, a true positive and false positive, or two false positives, depending on the Overlap Threshold (OT).

4-3-2 Global Performance Metrics

The four basic metrics as defined in the previous subsection will be used to calculate a couple of performance metrics that provide information about the global performance of the algorithm

with respect to multiple images. These metrics are used and described in the work of [19] as well. In the following definitions the basic performance metrics are not defined per image, but as the total number of occurrences of the corresponding metric. So i.e. TP is now defined as the sum of all true positives that occur in the images considered. Using these sums as definitions, a set of five metrics will be used to evaluate the performance of the algorithm with respect to all the images in the dataset. These metrics are:

$$\text{Precision (PR)} = \frac{TP}{TP + FP} \tag{4-6}$$

$$\text{Recall (RC)} = \frac{TP}{TP + FN} \tag{4-7}$$

$$\text{Specificity (SP)} = \frac{TN}{TN + FP} \tag{4-8}$$

$$\text{Accuracy (AC)} = \frac{TP + TN}{TP + FP + FN + TN} \tag{4-9}$$

$$\text{Negative Predictive Value (NPV)} = \frac{TN}{TN + FN} \tag{4-10}$$

Although the metrics above are quite self explaining by their names, it will be indicated for each metric what information it provides with respect to the performance of the detection algorithm. First of all the precision yields the fraction of all detections that are marked as true positive, and can therefore be seen as the probability that a detection resulting from the algorithm covers an object. Secondly, recall yields the fraction of objects that are correctly detected, and can therefore be seen as the probability that an object is detected by the algorithm. Third, contrary to the recall, the specificity relates to the ability of the algorithm to correctly identify the absence of objects. Furthermore, the accuracy indicates what proportion of the detection results are actually correct results and the negative predictive value finally indicates what fraction of the negative detection results are indeed reflecting the absence of an object. As stated in Section 1-2 the detection algorithm should ideally produce no false negatives and no false positives. The performance metrics should therefore ideally be equal to one, but in reality as close to one as possible.

A companying graphical performance metric, which is also often used in the literature that has been reviewed ([14], [17], [24]) is the precision-recall curve. In a precision-recall curve the precision is plotted against the recall for different settings of the algorithm. Precision-recall curves are a powerful tool to investigate the influence of variables on the performance and for the comparison of different algorithms, since multiple curves can be plot in one graph. In this work points that lie on these curves are used to illustrate the detection capabilities during the various stages of this research, such that the performance can easily be compared.

4-4 Optimisation of the Detection Algorithm

Before starting with the development of a classification scheme that will complement the object detection algorithm, care must be taken to ensure that the initial detection algorithm detects as many objects as possible in order to start with the best basis possible for the

classification step. Therefore, the detection algorithm will be optimised with respect to the recall. The precision is ignored at the moment, because the detection algorithm must ensure that, ideally, no objects are missed. Later on the classification step will have to ensure the improvement of the precision by eliminating, ideally, all false positives.

Within the detection algorithm as described in Section 4-2, there are five parameters that may have an influence on the performance: the direction in which the polynomials are fit (per row or per column), the order of the polynomials, N_{max}, the direction in which the thresholds are determined, the detection threshold parameter α and the clusterings parameter δ. Besides the influence of these parameters, also the overlap threshold used in the performance evaluation, see Equation (4-5), will have a huge influence. Therefore, this parameter must be taken into account as well. Using these six parameters, the object detection algorithm based on polynomial background estimation is optimised in a couple of steps. These steps, and the results obtained in each step, will be discussed in chronological order in the remainder of this section.

4-4-1 Optimal Detection- and Clustering Threshold

The first step in the optimisation process is to determine the best values for the detection threshold parameter (α) and the clustering parameter (δ) for different directions of the polynomial fit and threshold. Both parameters can be applied per row (horizontally) or per column (vertically). This yields a total of four possibilities: Vertical Fit, Horizontal Threshold (VFHT), Vertical Fit, Vertical Threshold (VFVT), Horizontal Fit, Horizontal Threshold (HFHT), and Horizontal Fit, Vertical Threshold (HFVT). Since in [5] it is found that third order polynomials are optimal to use, it is decided to copy this value and therefore N_{max} is set to 3 in all of the tests performed throughout this research. In order to determine which values for the detection threshold parameter (α) and clusterings parameter (δ) result in the highest recall, per combination of polynomial fit- and threshold direction, a series of tests is performed. During these tests the values for α and δ are varied from 1.50 to 3.00 in steps of 0.25 and from 1 to 5 in steps of 1 respectively. In total this results in 4*7*5 is 120 tests. For the performance evaluation of the detection results an OT of 0.1 is used. Reason for this low value of 0.1 for the overlap threshold is that it is not important to an operator that an object is completely enclosed if detected. If approximately one third of both the width and height are covered if a detection falls within the ground truth, this is assumed to be fine based upon talks with some operators, and results in an overlap threshold of 0.1. Table 4-3 shows the best results obtained during these tests with respect to the recall for each combination of polynomial fit and threshold direction.

From this table it follows that the best recall is obtained if the polynomials are fitted per row, the detection threshold is determined per row and if α and δ are set to 2.00 and 1 respectively. If these settings and an overlap threshold of 0.1 are used, 82.7% of the objects within the dataset are detected. Besides the best performance with respect to the recall, these settings also provide the best overlapping detection bounding boxes with, on average, an Overlap Ratio (OR) of 0.518. The precision on the other hand is astonishingly low and worst of all setups with a value of 0.001. This means that of all detections only 0.1% is a true positive. In other words, on average there are per true positive approximately a thousand false positives. A result that definetely must be improved further beforehand, since the true

Table 4-3: The best detection results with respect to the recall resulting from the tests with different combinations of the polynomial fit- and threshold direction using an overlap threshold of 0.1

	α	δ	**Precision**	**Recall**	**Specificity**	**Accuracy**	**NPV**	**Av. OR**
VFHT	2.00	2	0.002	0.803	0.000	0.002	0.000	0.451
VFVT	2.50	2	0.005	0.703	0.000	0.005	0.000	0.410
HFHT	2.00	1	0.001	0.827	0.000	0.001	0.000	0.518
HFVT	2.00	1	0.001	0.807	0.000	0.001	0.000	0.410

positive detections are 'drowned' in false positives. The result is a very low accuracy and the specificity and the negative predictive value are zero. This means that although no objects are present in an image, the algorithm returns at least one or more detections anyway. During visual inspection of the results its is noticed that a lot of detections are only a few pixels in size. It seems therefore sensible to somewhat improve the results by setting limits on the minimum- as well as maximum allowable detection size. Later on in this section this improvement will be discussed further.

4-4-2 Optimal Direction for the Polynomial Fit and Detection Threshold

In the previous optimisation step it was found that the best recall is obtained when the polynomials are fitted per row and if the detection thresholds are determined per row as well. Before continuing with a Horizontal Fit, Horizontal Threshold (HFHT) set-up it must be checked whether this combination is best in general, or perhaps only if an overlap threshold of 0.1 is used. In order to do so, the best detection results as reported in Table 4-3 are several times re-evaluated using an overlap threshold varying from 0 to 1 in steps of 0.01. The outcome of these re-evaluations are shown in Figure 4-5, where the overlap threshold is plotted against the corresponding recall.

From Figure 4-5 it clearly follows that this set-up is optimal, since for nearly all possible values of the Overlap Threshold (OT) the resulting recall is bigger than the corresponding recall of any of the other set-ups. Based on this outcome, and the results obtained in the previous step as reported in Table 4-3, it is decided to use the detection algorithm in a HFHT setup, with the detection threshold, $\alpha = 2.00$ and the clusterings threshold $\delta = 1$, as the basis detection algorithm throughout the rest of this research. For the performance evaluation an overlap ratio of 0.1 will be used (see 4-4-1).

4-4-3 Minimum- and Maximum Detection Size

As stated in subsection 4-4-1, the basis detection algorithm produces way to much false positives. Before a classification scheme is developed which must classify the detection output, it is sensible to eliminate a part of the false positives beforehand. It is sensible to do this by introducing a constraint on the minimum- and maximum area that a detection BB may have to survive. First of all, setting a minimum detection size seems to be a necessity for the correct operation of a classification scheme. A few pixels do simply not provide enough information for correct classification, no matter how good the features used might be. If an area consisting

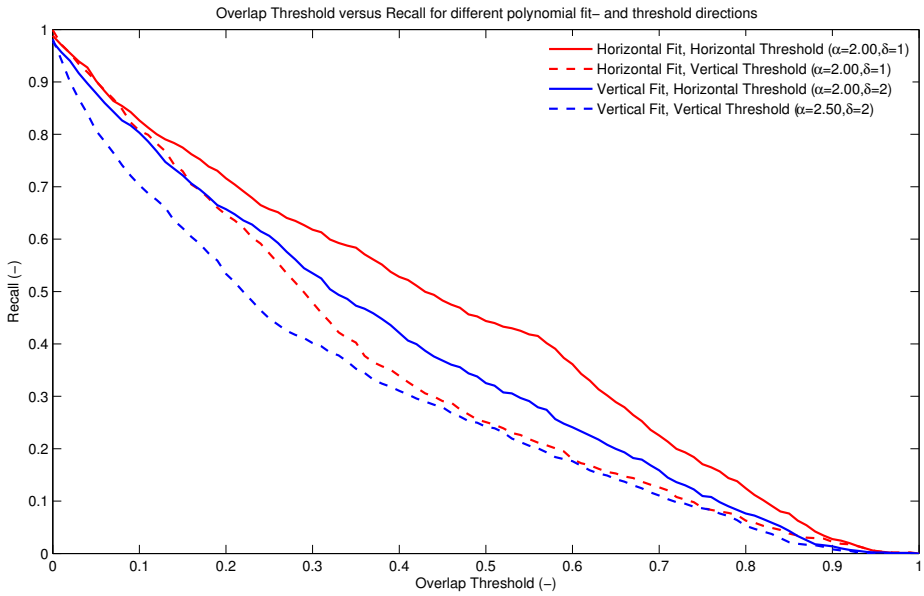
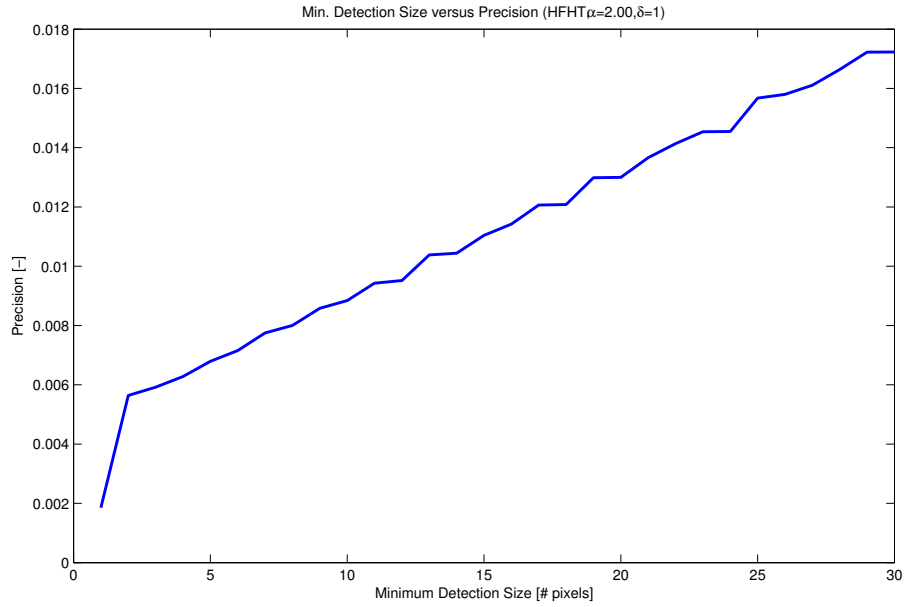


Figure 4-5: Overlap Threshold versus Recall for different polynomial fit- and detection threshold directions

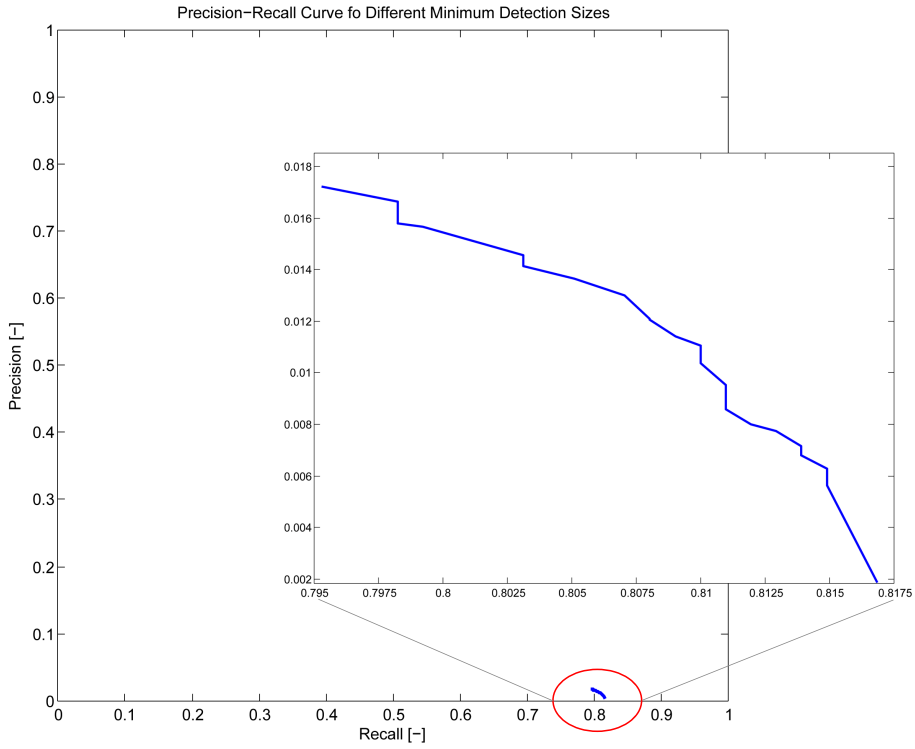
of only a few pixels has to be classified as either target or background, it is expected that this would be practically impossible and would cause a lot of wrongly classified areas. Even for a human it would be impossible to tell, looking at only a few pixels, whether they belong to an object or the background. Furthermore, a lower limit is necessary in order to maintain real-time, or near real-time, processing capabilities. At the moment there are so many false positives that, if not reduced, the system requirement of (near) real-time application is likely to be compromised. On the other hand, also setting an upper limit for the detection size is sensible. It is unlikely that objects will appear very large in the image during surveillance, but this would be more likely if there is zoomed on a specific object. Therefore, very large detected areas are most likely background and can be removed. It is reasoned that an upper limit of 15% of the total image size, which equals 0.15*720*576 = 62208 pixels, will be an acceptable upper limit.

In order to determine an acceptable lower limit, the lower limit is varied from 1 to 30 in combination with the fixed upper limit of 62208 pixels. The effect on the precision when these limits are used is shown in Figure 4-6. In this figure the effect on the recall can be derived from the precision-recall plot by looking up the recall that corresponds with a certain value of the precision.

Based on sub figure 4-6(a) it can be concluded that eliminating small detections has a significant, positive, effect on the precision. Especially the elimination of single, isolated, pixels causes a jump in the precision as can be seen at the transition of a minimum detection size of 1 to 2. Furthermore, it turns out that using an upper limit of 0.15 times the image size provides an improvement in the precision as well. The precision at a minimum detection size of 1 equals 0.002, which is nearly twice as high as the 0.001 when no upper limit is used, see Table 4-3. Furthermore, if the corresponding precision-recall curve is inspected, it turns out



(a) Minimum detection size versus precision



(b) Precision-recall curve for minimum detection size

Figure 4-6: The influence of the minimum allowed detection size on the precision (a), and on the precision/recall (b). The maximum allowed detection size is set to 15% of the image size: 62208 pixels

that around a precision of 0.01 there is an optimum between the gain in precision and the loss in recall. This precision corresponds with a minimum detection size of 15 pixels. Since this is also the smallest object present in the dataset and seems to be a reasonable area from which on successful classification should be possible, it is chosen to use this as the lower limit for the detection size.

The baseline detection algorithm

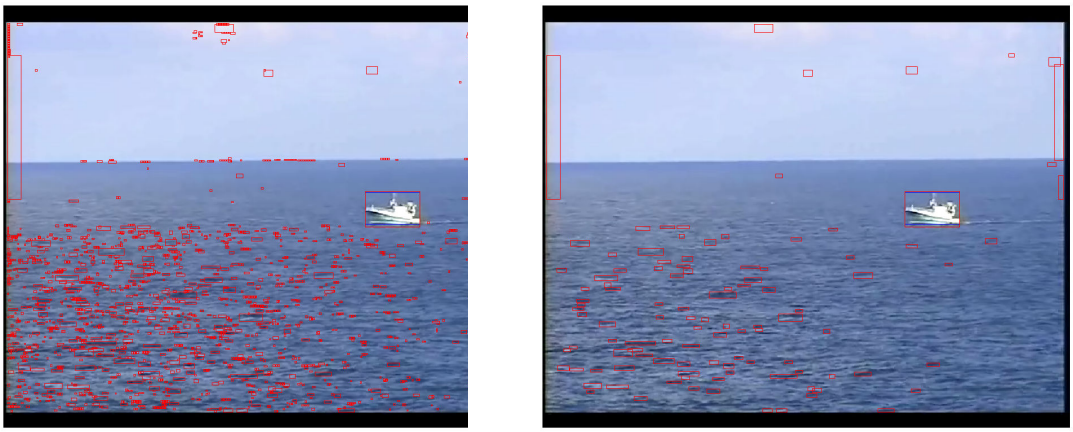
During the rest of the research the detection algorithm will therefore be used in a HFHT set-up, with $\alpha = 2.00$, $\delta = 1$, and a minimum and maximum detection size of 15 and 62208 pixels respectively. The baseline performance of the detection algorithm using these settings, and an overlap threshold of 0.1 during the performance evaluation, is shown in Table 4-4.

Table 4-4: Performance of the baseline detection algorithm.

	α	δ	**Precision**	**Recall**	**Specificity**	**Accuracy**	**NPV**	**Av. OR**
Baseline	2.00	1	0.011	0.810	0.000	0.011	0.000	0.522

Performance comparison

The effect of introducing a minimum- and maximum size on the output of the detection algorithm is shown in Figure 4-7. As can be seen the effect is quite substantial since a lot of false positive detections are eliminated. The substantial reduction is even better visible in the precision-recall plot as shown in Figure 4-8. The original detection algorithm, so if all detection sizes are allowed, has a precision and recall of 0.001 and 0.827 respectively. The baseline detection algorithm as will be used throughout the rest of this work on the other hand, has a precision and recall of 0.011 and 0.810 respectively. This holds an improvement of the precision of approximately a factor ten - or in other words one tenth of all initial false positives have been eliminated - while only about 1.7% of the true positive detections are lost, which is considered an acceptable loss. Despite the significant improvement, the precision is still very low and not even near an acceptable value for practical use. Later on, during classification of the output as either target or background, it is aimed to improve the precision even further while preserving the best obtainable recall of 0.810. This implies that the performance, after classification of the detection output, should move along the vertical line through the point of the baseline algorithm. Ideally the performance of the two stage detection algorithm should reach the top of the line at a precision of one.



(a) Original detection output

(b) Detection output after introduction of a minimum- and maximum detection size

Figure 4-7: Graphical illustration of the effect of introducing a minimum- and maximum detection size on the detection output.

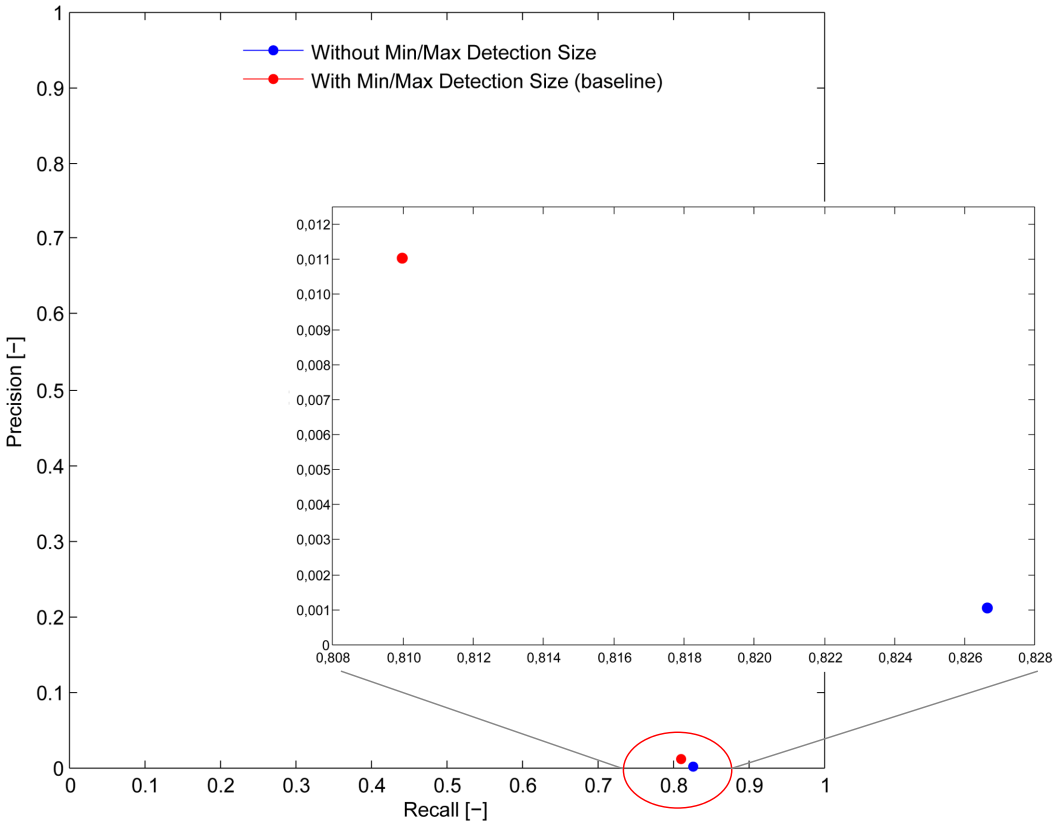


Figure 4-8: Precision-Recall plot of the detection algorithm after introducing a minimum- and maximum detection size.

Chapter 5

Development of the Classifier

This chapter describes the development of a classifier which should classify the resulting Bounding Boxes (BBs) from the detection algorithm as either 'target' or 'background'. Since in the literature an existing set of features has been found that showed promising results, it is decided to use this set as a starting point to determine which classifier is best to use. This results in an initial classifier whose performance is analysed in detail in Chapter 6. First, the features and classifiers used in this work are discussed in the first section. The performance metrics that are used to evaluate the performance of the classifiers are discussed in the second section, followed by a description of the feature dataset required for training and testing of the classifiers in the third section. Finally, the fourth and last section of this chapter is dedicated to the tests and results that are used to determine which of the considered classifiers is best to use.

5-1 Classification Principles

As in any classification problem, there are two critical components with respect to the performance: 1) the features that are used, and 2) the type of classifier that is used. In case of the latter component there is a variety of classifiers to choose from and tests should point out which provide the best performance. More difficult however is the former component. There are many, many features one can think of in most problems. Some provide a strong distinction between the classes while others do hardly contribute. Unfortunately, it is almost impossible to tell beforehand which features are useful, and in practice it comes down to trial and error to identify them.

In case of classification of bounding boxes there even is, besides the possibilities in features, an additional problem: the BBs are not fixed in size but differ. Therefore, features determined per pixel, e.g. pixel intensity or first and second derivative of the pixel intensity, cannot be used directly to compare BBs of different sizes due to the fact that this would result in a different amount of features per box. A solution that allows to compare bounding boxes of different sizes is therefore required. In the work of [8] a solution to this problem is described,

as well as a set of features that can be used. In the remainder of this section, this solution, together with the features, is described in detail followed by a discussion of the classifiers that will be considered.

5-1-1 Features

In [8] local covariance descriptors of subregions of the image are used for target detection and tracking purposes. The covariance descriptors of a subregion, which cover both spatial and statistical information and their correlations, are used as features in the form of a local covariance matrix. This local covariance matrix is used accordingly to classify whether a subregion belongs to the target class or background class. As will become clear, an advantage of using the local covariance matrix as feature input for the classifier is that it does not depend on the size of the subregion. This means that regions with different sizes can be classified without problems and therefore this approach is adopted in this work as well.

Computation of the local covariance descriptors

The authors of [8] have chosen to use local variance descriptors due to their low computational complexity, robustness to partial occlusion and the opportunity to compare subregions of different sizes. Furthermore, they also enable to add or remove features in a simple manner, which makes the approach flexible for different targets such as surface- or air targets. The computation of the local covariance matrix goes as follows. Feature matrices $(f_i : i = 1, 2, \ldots, D)$ extracted from a $W \times H$ subregion of an image are stacked to form a $W \times H \times D$ dimensional feature tensor $T_f(:, :, :)$, see Figure 5-1. W and H are the width and height of the subregion and D is the number of features extracted from the subregion. Which features this exactly are, is discussed in the second part of this section.

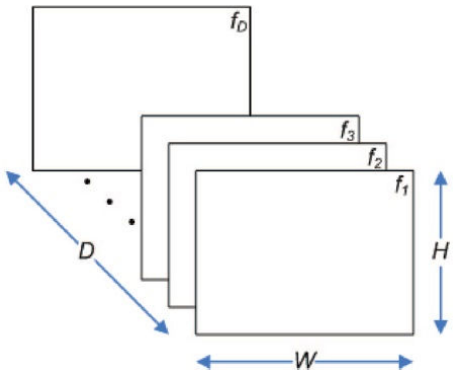


Figure 5-1: The feature tensor T_f formed by placing feature matrices back to back, [8]

In the feature tensor, the elements in each layer with the index (w, h) are sorted to construct the feature vector $\underline{S}_k$, see Eq. (5-1). In total, $K = W \times H$ feature vectors $\underline{S}_k$ are constructed.

$$\underline{S}_k = [f_1(w, h)\ f_2(w, h) \cdots\ f_D(w, h)] \tag{5-1}$$

where $w = 1, 2, \ldots, W$, $h = 1, 2, \ldots, H$, $k = 1, 2, \ldots, K$ and $K = W \times H$.

The local covariance matrix, M_{loc}, can be computed by using the feature vectors as follows:

$$M_{loc}(i, j) = \frac{1}{1 - K} \left(\sum_{k=1}^{K} \underline{S}_k(i) \underline{S}_k(j) - \frac{1}{K} \sum_{k=1}^{K} \underline{S}_k(i) \times \sum_{k=1}^{K} \underline{S}_k(j)) \right) \tag{5-2}$$

where $i, j = 1, 2, \ldots, D$.

From Equation (5-2) it follows that the local covariance matrix does not depend on the size of the subregion, but only on the amount of features that are extracted from the region. Therefore, this approach makes it possible to classify bounding boxes with different sizes, since the extracted features will remain constant. Which features this are, and how many, is discussed hereafter. Notice that it will suffice to use only the upper- or lower triangle of the covariance matrix, due to its redundancy.

Besides using the covariance matrix as feature input, there are also other solutions that allow to compare bounding boxes of different sizes. For example, it would be possible to average features determined per pixel or to resize the boxes such that they become of the same size. Before continuing, it is checked whether the covariance matrix provides better results than averaging the features or resizing the boxes. Therefore, these three solutions are tested using the features as described in the next paragraph, and in combination with three basic classifiers that are described later on in this section. The classifiers used are: Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier (QDC), Nearest Mean Classifier (NMC) and are evaluated using 'leave multiple sets out 10-fold cross validation' as described in the next section and a dataset containing 71740 background- and 914 target samples. The results of the tests are shown in Table 5-1.

Table 5-1: Comparison of different feature input solutions: 1) Average feature values, 2) Features of bounding boxes resized to 6 by 10 pixels, 3) Local Covariance Matrix, by means of the average AUC value and standard deviation, between brackets, of 10-folds.

	Average (9-D)	**Resized Box (540-D)**	**Cov. Matrix (45-D)**
	AUC	**AUC**	**AUC**
LDC	0.814 (0.067)	0.892 (0.044)	0.927 (0.041)
QDC	0.900 (0.042)	0.867 (0.049)	0.894 (0.037)
NMC	0.789 (0.095)	0.790 (0.094)	0.883 (0.028)

Based on the results as reported in Table 5-1, it is concluded that the local covariance matrix is best to use as feature input for the classifier. This solution provides for two of the three classifiers the largest average AUC value and for all classifiers the lowest standard deviation. So, in general, this solution is most consistent and furthermore it provides the largest average AUC value of all tested combinations in combination with the LDC. Therefore, it is chosen to adopt the local covariance matrix as feature input throughout the rest of the research.

Furthermore, Table 5-1 shows that averaging the features performs, on average, worst of the tested solutions. This worst performance is likely due to the fact that only 9 features (9-D) are used, which is a very small number and as it seems is not sufficient. Best classifier in this case is the quadratic classifier with an average AUC value of 0.900, which is significantly larger than the 0.814 and 0.789 of the other two classifiers. This illustrates that a more flexible classifier is way better capable of separating the classes in case of low dimensionality. On the other hand, using too many features does not have a positive effect on the discriminative power of the classifier as well. This is illustrated by the results of the resized boxes and the covariance matrix. In case the boxes are resized, this results in 540 features (540-D) and in case the local covariance matrix is used this results in 45 features (45-D). If the results are compared, it turns out that resizing the boxes provides a lower performance than the

covariance matrix for all three classifiers. This phenomenon is often referred to as 'the curse of dimensionality', which is also the case in this example. Finally, given the fact that the average AUC values of the three basic classifiers are reletively high, it is concluded that the 9 features as described hereafter are quite discriminative and suitable to use at first.

Feature extraction

In [8] the features that are determined per pixel of the subregion and used for computation of the local covariance matrix as previously described, are:

- Intensity (I)
- Horizontal position (x)
- Vertical position (y)
- First derivative in horizontal direction ($\partial_{1,x} = \partial I/\partial x$)
- First derivative in vertical direction ($\partial_{1,y} = \partial I/\partial y$)
- Second derivative in horizontal direction ($\partial_{2,x} = \partial I^2/\partial x^2$)
- Second derivative in vertical direction ($\partial_{2,x} = \partial I^2/\partial y^2$)
- Gradient Magnitude ($GM = \sqrt{\partial_{1,x}^2 + \partial_{1,y}^2}$)
- Gradient Orientation ($GO = \tan^{-1}(\partial_{1,y}/\partial_{1,x})$)

Since reported results in [8], using these features in combination with a Support Vector Machine (SVM) classifier, are high, in the order of 99.0 - 99.8%, it is decided to use this feature set as a starting point for the classification step. In this work the first and second derivates in x and y direction are approximated by filtering the image with a Prewitt filter, [25]. Later on, during the performance analysis, it is investigated whether additional features improve the classification results.

5-1-2 Classifiers

In this work there is considered a set of six standard classifiers, using the local covariance matrix calculated with the features as described in the previous section as feature input. The function of a classifier is to determine, based on a training set, a decision boundary in the feature space that allows to predict the classes - in our case target and background - as good as possible. The classifiers that are considered in this work are: the Linear Discriminant Classifier (LDC), Quadratic Discriminant Classifier (QDC), Nearest Mean Classifier (NMC), K-Nearest Neighbour Classifier (KNNC), Parzen Classifier (PARZENC), and Fisher Classifier

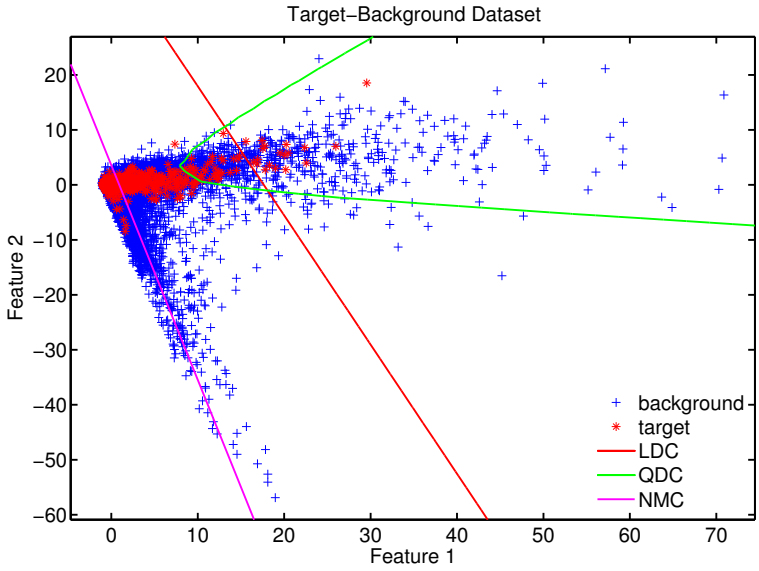


Figure 5-2: Examples of decision boundaries separating two classes in a two dimensional feature space. The red, green and magenta lines depict the decision boundaries resulting from a LDC, QDC and NMC respectively.

(FISHERC). Since the way in which each classifier determines the decision boundary is different, the resulting decision boundaries are different as well, see Figure 5-2. Tests should point out which of the classifiers listed above provides the best decision boundary. In the remainder of this section the main principles of the classifiers considered are briefly discussed, for a full description there is referred to [26].

As stated earlier, a classifier determines a decision boundary based on training samples to predict the class of new samples. But how are these decision boundaries determined? A LDC approaches the determination of the decision boundary by assuming that the conditional probability density functions of the classes are normally distributed with certain means and equal variances. Under these assumptions, Bayes rule is used to determine the posterior probabilities and the decision boundary is formed by a line, surface or hyperplane at which the posterior probabilities are equal. A test sample will now be assigned to the class with the largest posterior probability. If it is not assumed that the classes have equal variances, the resulting classifier is a QDC. A similar, but slightly different classifier is the Fisher classifier. The FISHERC also assumes normally distributed classes but uses the ratio of the variance between the classes and the variance within the classes to determine the decision boundary. More different are the NMC, KNNC and PARZENC. A nearest mean classifier determines the class of a sample by comparing the distance of the sample to the centroids (means) of the classes in the feature space, and assigns the sample to the class of which the centroid is closest. Even so, a k-nearest neighbour assigns a test sample to the class which is most frequent among the k-training samples nearest to the test sample. Finally the Parzen classifier is somewhat similar to a K-Nearest Neighbour classifier, it estimates the probability density functions of the classes by using Gaussian kernels of width 'h' around each of the samples in the training set. If 'h' is chosen small, only samples nearby the test sample will be considered while for larger values also more distant samples are considered. A test sample is classified accordingly

to the class with the highest posterior probability. For the practical implementation of the classifiers in Matlab , the pattern recognition toolbox 'PRTools' is used. PRTools is the basic software package used and described in [27].

5-2 Performance Evaluation of a Classifier

Before training and testing of the various classifiers can take place, the evaluation procedure has to be defined. Two common performance measures will be used to evaluate the performance of the classifiers: the classification error and the Area Under the ROC Curve (AUC) value.

5-2-1 Classification Error and the Area Under the ROC Curve (AUC)

First measure that will be used to compare the performance of the classifiers is the classification error, E. The classification error is computed using the labelled test set by counting the erroneously classified samples per class. The classification error is then calculated as follows:

$$E = \sum_{c=1}^{C} \frac{N_{e,c}}{N_{s,c}} \cdot P_c \tag{5-3}$$

where: E is the classification error, C is the total number of classes, N_e is the number of erroneously classified samples of class c, N_s is the total number of samples of class c, and P is the class prior of class c.

Although the classification error provides an indication of the performance of a classifier and allows to easily compare various classifiers, it does not fully represent the performance of a classifier. The classification error reflects the performance at only one specific operating point of the classifier and does therefore not fully describe the behaviour, which might result in a distorted view of the performance and may lead to suboptimal decisions.

Therefore, a second common performance measure is considered as well, the AUC value. A Receiver Operating Characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a classifier at different discrimination thresholds. The curve is created by plotting the false positive rate, FPR $= FP/(FP + TN)$, against the true positive rate which is equal to recall, TPR $= TP/(TP + FN)$, for various operating points of the classifier. Hence the ROC curve fully describes the obtainable performance. Although the ROC curve could directly be used as performance measure it is chosen to use the AUC value instead, because this is a single measure that reflects the ROC curve and allows easy comparison of multiple classifiers. For determination of the AUC value in Matlab, the data description toolbox 'dd_tools' is used [28].

In order to obtain reliable, unbiased values of the classification error and the AUC value an adapted form of the standard cross-validation method, Leave Multiple Sets Out Cross-Validation (LMSO CV), will be used for the evaluation. This evaluation procedure is explained in the next subsection.

5-2-2 Leave Multiple Sets Out Cross-Validation

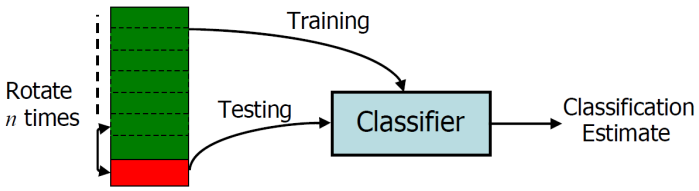


Figure 5-3: Illustration of n-fold cross validation [29].

Cross validation is a technique for assessing how results will generalize to an independent data set and is commonly used in the field of pattern recognition to estimate how a classifier will perform in practice. In n-fold cross-validation the original dataset is (randomly) partitioned into n equally sized subsets. Of the n subsets, $n-1$ subsets are used to train the classifier and 1 subset is retained as validation set to test the trained classifier. The process of taking $n-1$ subsets for training and one for testing is repeated n times (the folds), where each of the n subsets is used exactly once as test set, see Figure 5-3. Finally the results of the n-folds can be averaged to produce a single estimation. The advantage of n-fold cross-validation over, for example, repeated sub-sampling is that all samples are used for both training and testing and each sample is used for validation only once. Because 10-fold cross-validation is most commonly used, it is decided to set n to 10 in this work as well.

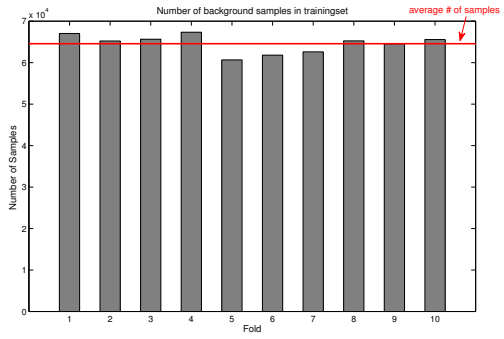
Although the standard cross-validation procedure as described above allows in most case to obtain reliable, unbiased classification estimates, this does not hold for the dataset used in this work. Since from each of the 160 visible light videos there are on average 5 images included in the final dataset, the images originating from the same video show great similarity in terms of objects/targets that are present as well as in terms of meteorological- and environmental circumstances. Due to the affinity between those images, it must be taken care that the samples originating from those images do not end up in both the training and test set because this would introduce a bias. If the samples are plainly or randomly divided in 10 sets, this is likely not the case and hence causing a bias. In order to avoid this bias, the samples are sorted per video and the subsets are constructed using the samples per video accordingly. In case of the 10-fold cross-validation this implies that the samples belonging to 144 videos are included in the training set and the samples of the remaining 16 videos in the test set. This procedure is hence referred to as LMSO CV and is adopted as the evaluation procedure in this work.

5-2-3 Validation of the Performance Evaluation

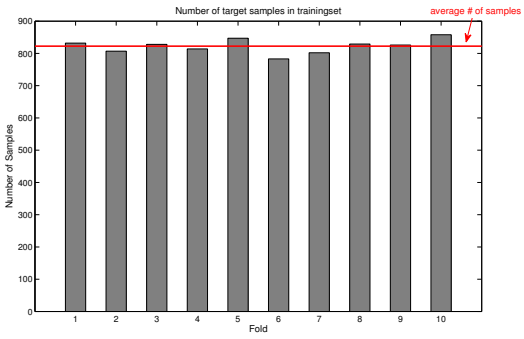
By adopting LMSO CV as evaluation procedure, the certainty of fixed amount of samples in both the training- and test set has been lost. As a consequence, the amount of samples in the training- and/or test set may become imbalanced between the folds and compromise the commonly accepted value of 10-folds. As a consequence it must be verified whether 10-fold LMSO CV is possible and a sensible choice. This choice is verified by examining the amount of samples in the training- and test set of each fold and an AUC learning curve in which the number of training videos is plotted against the average AUC value obtained after 20 runs.

First of all the amount of target- and background samples per fold in the training- and test is determined for a dataset containing a total of 71740 background- and 914 target samples. The results are shown in Figure 5-4. From this figure it follows that the number of samples of both classes is more or less equally distributed in both the training and test set. Since none

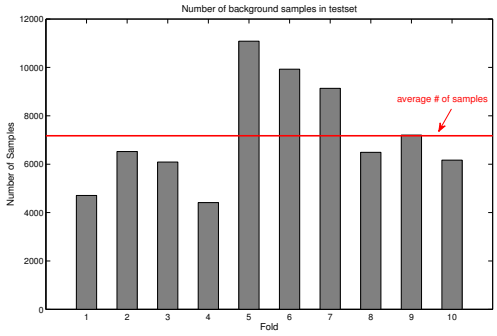
of the folds severely deteriorate from the average number of samples per fold it is concluded that 10-fold LMSO CV does not form an obstruction for the evaluation and should provide an unbiased and reliable performance estimate.



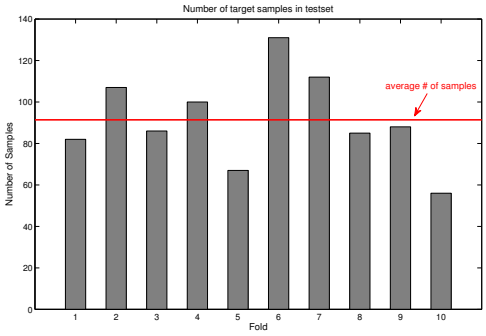
(a) Nr. of background samples in the training set



(b) Nr. of target samples in the training set



(c) Nr. of background samples in the test set



(d) Nr. of target samples in the test set

Figure 5-4: Distribution of the amount of background/target samples in the training and test set per fold

Secondly the influence of the number of videos included in the training set is investigated. Using the same dataset as before in combination with a linear discriminant classifier the amount of videos in the training set is varied from 1 to 159 and are chosen randomly. The trained classifier is tested using the samples of the remainder of the 160 videos and the performance by calculating the AUC value. Since the performance depends on which videos are exactly included in the training and test set, especially when only a few of the videos are included in either the training or test set, the AUC value is determined over 20 runs and averaged. Figure 5-5 shows the average results obtained after 20 runs.

Based on Figure 5-5 it is concluded that 10-fold LMSO CV is acceptable as well in terms of stability and learning ratio. If the amount of videos in the training set is small, the performance of the classifier is low due to a small number of training samples. Increasing the number of videos in the training set, increases the AUC value and hence the performance of the classifier. Until approximately 144 videos the performance converges to it's maximum performance and is more or less constant. If more videos are included, the performance does not significantly improve any more and starts to deteriorate as can be seen at the oscillation at the end of the curve. This deterioration confirms that the performance strongly depends

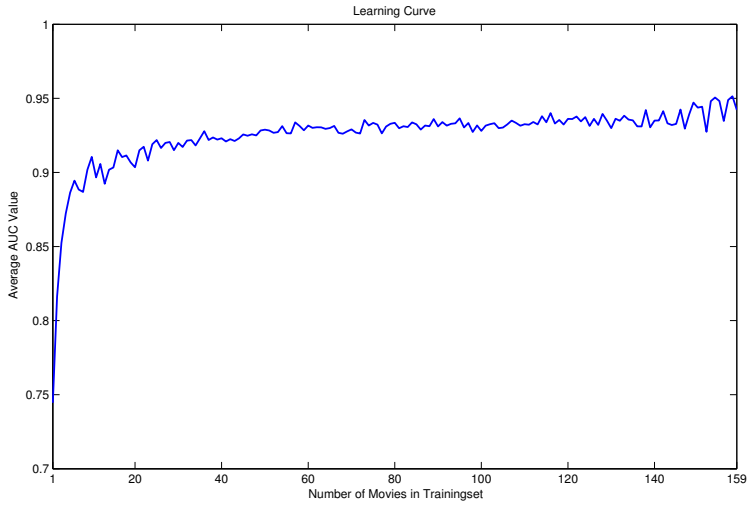


Figure 5-5: Learning curve for LMSO-CV. The curve shows the averaged AUC values obtained after 20 runs, the training- and test videos are chosen randomly, in combination with a LDC.

on which samples are included in the training and test set. Since 144 videos in the training set lies at the boundary of stable behaviour and provides a well trained classifier, it is all in all concluded that 10 fold cross-validation, leaving multiple sets out, is viable and a sensible choice to use.

5-3 Tests and Results

In the final section of this chapter the feature datasets and the tests and results to determine which of the classifiers performs best are described and discussed. At the end, the best performing classifier is combined with the baseline detection algorithm and the results of this initial two-stage detection approach is compared with the performance of the baseline.

5-3-1 Feature Dataset

Now the evaluation and testing procedure is established, a feature dataset for training and testing of the classifiers is required. For the tests, two feature datasets are constructed, using the detection output of the baseline detection algorithm as samples. From each image in the dataset the areas that are detected by the baseline detection algorithm are 'cut-out' of the corresponding image. From the extracted image regions, the local covariance matrix is calculated using the features as described in the first section of this chapter. The resulting samples are next used to construct two different feature datasets:

1. **Dataset with detections as target samples**
   The first dataset uses the detection samples that have no overlap with a Ground Truth (GT) as background samples and those with an Overlap Ratio (OR) bigger than 0.1 as target samples. It is chosen only to include 'pure' target and background samples, and

therefore samples that have an overlap ratio lower than 0.1 or with multiple Ground-Truths are discarded. This dataset contains 71740 background samples and 916 target samples.

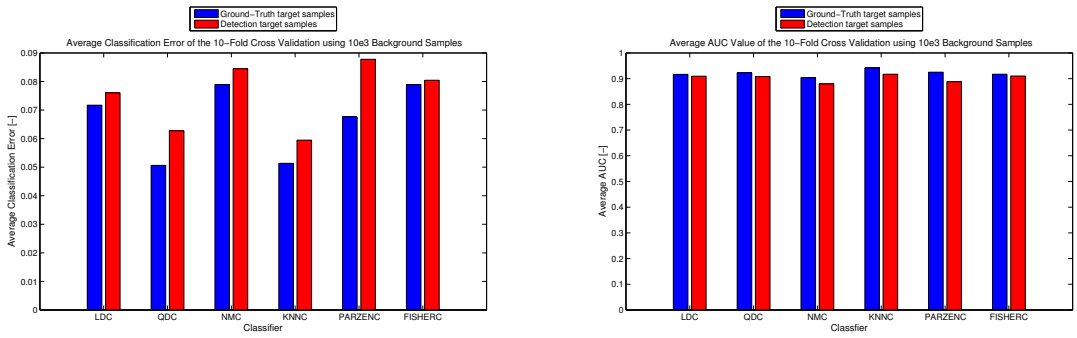
2. **Dataset with GT as target samples**
   Like the first dataset, this dataset is constructed using the detection samples with no overlap as background samples and disregards the samples with an overlap ratio lower than 0.1 or with multiple Ground-Truths. In this dataset, however, the target samples are not formed by detections with a large enough overlap, but by the GT itself. The reason why this dataset is created is to verify whether it makes a difference with respect to the performance of the classifier. This dataset contains 71740 background samples and 1021 target samples.

During the tests it must become clear whether the classifier performance differs if properly enclosed targets are used for training or not.

5-3-2 Initial Classification Results

In the first performed test, the set of six classifiers using the local covariance matrix and the features as described in Section 5-1 as feature input, is trained and tested using the leave multiple sets out cross-validation procedure. Although preferably all the classifiers are tested with the complete feature datasets, this is impossible due to memory constraints. Especially the KNN and Parzen classifiers require a lot of memory and cause an out of memory error due to the large amount of background samples in both datasets. Therefore, a subset of ten thousand background samples is used instead.

The results of these initial tests, are shown in Figure 5-6 and Table 5-2.



(a) Average Classification Errors

(b) Average AUC Values

Figure 5-6: Average classification results if a subset of ten thousand background samples is used.

Based on the results, a couple of conclusions are drawn. First of all, if Figure 5-6(a) is compared with Figure 5-6(b) it is noticed that the results are quite different. The average errors are much more fluctuating between the classifiers than the AUC values are. This confirms the initial expectation that the classification error is not a very strong performance measure and this measure will therefore be dropped in further tests. Secondly, it is noticed that for all

Table 5-2: Average classification error and AUC value for six different classifiers using a subset of ten thousand background samples. The value between brackets is the standard deviation of the Error/AUC between the folds.

	BB target-samples		**GT target-samples**	
	Error	**AUC**	**Error**	**AUC**
LDC	0.076 (0.036)	0.910 (0.059)	0.072 (0.028)	0.917 (0.052)
QDC	0.063 (0.025)	0.908 (0.034)	0.051 (0.011)	0.922 (0.044)
NMC	0.084 (0.021)	0.879 (0.032)	0.079 (0.021)	0.904 (0.046)
KNNC	0.059 (0.013)	0.917 (0.025)	0.051 (0.011)	0.942 (0.025)
PARZENC	0.088 (0.025)	0.888 (0.034)	0.068 (0.019)	0.925 (0.040)
FISHERC	0.080 (0.027)	0.910 (0.059)	0.079 (0.026)	0.917 (0.052)

classifiers the feature dataset with the GT target samples provides slightly better results than the dataset with detection Bounding Box (BB) as targets. Finally, if the average obtained AUC values of the classifiers are compared, it turns out that the KNNC has the largest AUC value and the lowest standard deviation between the folds. It is therefore concluded that of the considered classifiers, this classifier seems to provide the best results and is best to use with the problem.

Although the initial results indicate that the KNN classifier is the best classifier for the problem, the results might change when all available background samples are used. This is likely to be the case because the learning curve, see Figure 5-5, clearly shows that using more samples (videos) will improve the discriminative power of at least a linear discriminant classifier. It is expected that this will also hold for other classifiers.

5-3-3 The Influence of the Number of Samples in the Training Set

Since it is expected that increasing the amount of background samples in the training set will have a positive effect on the classification performance of the classifiers, the LD, QD and NM classifiers are evaluated with the complete feature datasets, containing 71740 background samples, as well. The results of these tests are shown in Figure 5-7 and Table 5-3.

Table 5-3: Average AUC values for three different classifiers using all available background samples. The value between brackets is the standard deviation of the AUC between the folds.

	AUC - BB	**AUC - GT**
LDC	0.927 (0.041)	0.936 (0.037)
QDC	0.894 (0.037)	0.908 (0.048)
NMC	0.883 (0.028)	0.905 (0.046)

If these results are compared with the results obtained when a subset of background samples is used, see Figure 5-6 and Table 5-2, it turns out that using more background samples in the training set has indeed a positive effect on the overall performance of the some of classifiers. The average AUC values of the linear and nearest mean classifiers have slightly risen. Only exception though is the quadratic classifier, of which the AUC is slightly dropped from 0.908 to

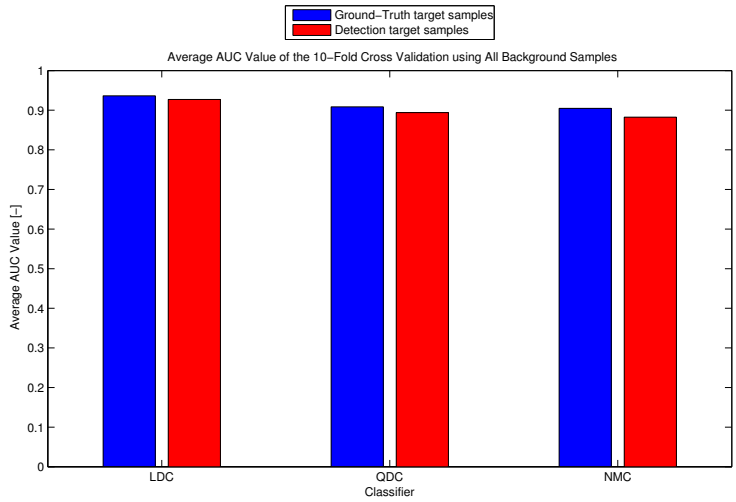


Figure 5-7: Average classification results if all available background samples (70k+) are used.

0,894 and from 0.922 to 0,908 and performs worse. Furthermore, once again the performance is a little bit better for the dataset with the GT samples as target than the dataset with the (detection) BB as target samples.

Despite the fact that the using the ground-truth provides better classification results, it is chosen to continue the research with the dataset using the detections (BBs) as target samples. The reason for this is that the samples included in this dataset reflect reality opposed to the ground-truth dataset. In reality the detections are most often not fully containing the object, but only partly. The better classification performances obtained with the ground-truth samples, however, underlines the importance of properly enclosing detection results. Finally it is concluded that the best performing classifier, although the standard deviation is not the lowest, is the linear discriminant classifier with an average AUC value of 0.927. In the remainder of this project the performance of this classifier will therefore be further analysed.

5-3-4 Performance of the Two-Stage Detection Algorithm

In the previous section it is shown that, on average, the best performing classifier is the linear discriminant classifier trained and tested with a feature dataset with the detection output of the baseline classifier as target and background samples. Before continuing with a detailed analysis of this classifier, the corresponding ten trained classifiers are used to classify the output of the baseline detection algorithm. By doing so, the aimed two-stage detection algorithm is born and its performance is evaluated as described in Section 4-3 using an overlap threshold of 0.1. The results of the detection algorithm after classification of the initially returned bounding boxes is shown in Table 5-4 and a comparison with the performance of the previous detection algorithms is shown in Figure 5-8.

Compared with the baseline, the performance of the detection algorithm after classification is simultaneously improved as well as degraded. The performance is especially improved in terms of the precision, which has made a jump from 0.011 to 0.200, and in terms of the specificity,

Table 5-4: Detection results of the baseline and after classification with the linear discriminant classifier.

	α	δ	**Precision**	**Recall**	**Specificity**	**Accuracy**	**NPV**	**Av. OR**
Baseline	2.00	1	0.011	0.810	0.000	0.011	0.000	0.522
Classified	2.00	1	0.200	0.333	0.045	0.165	0.086	0.501

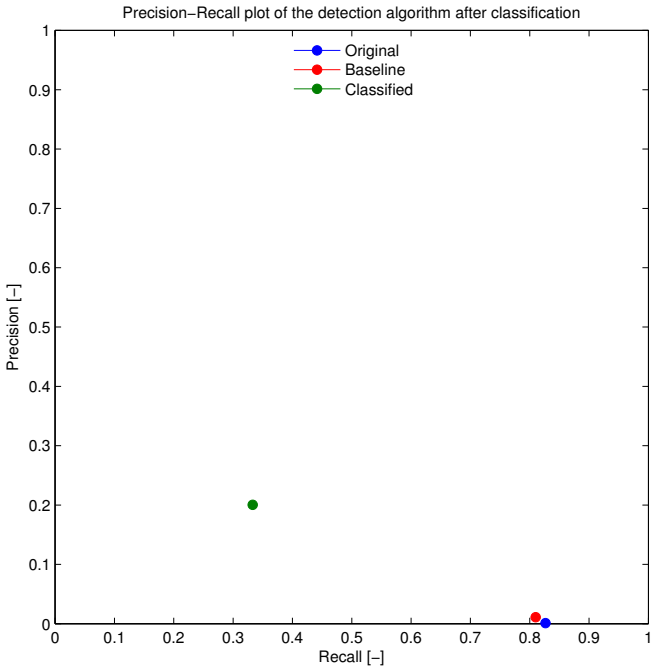
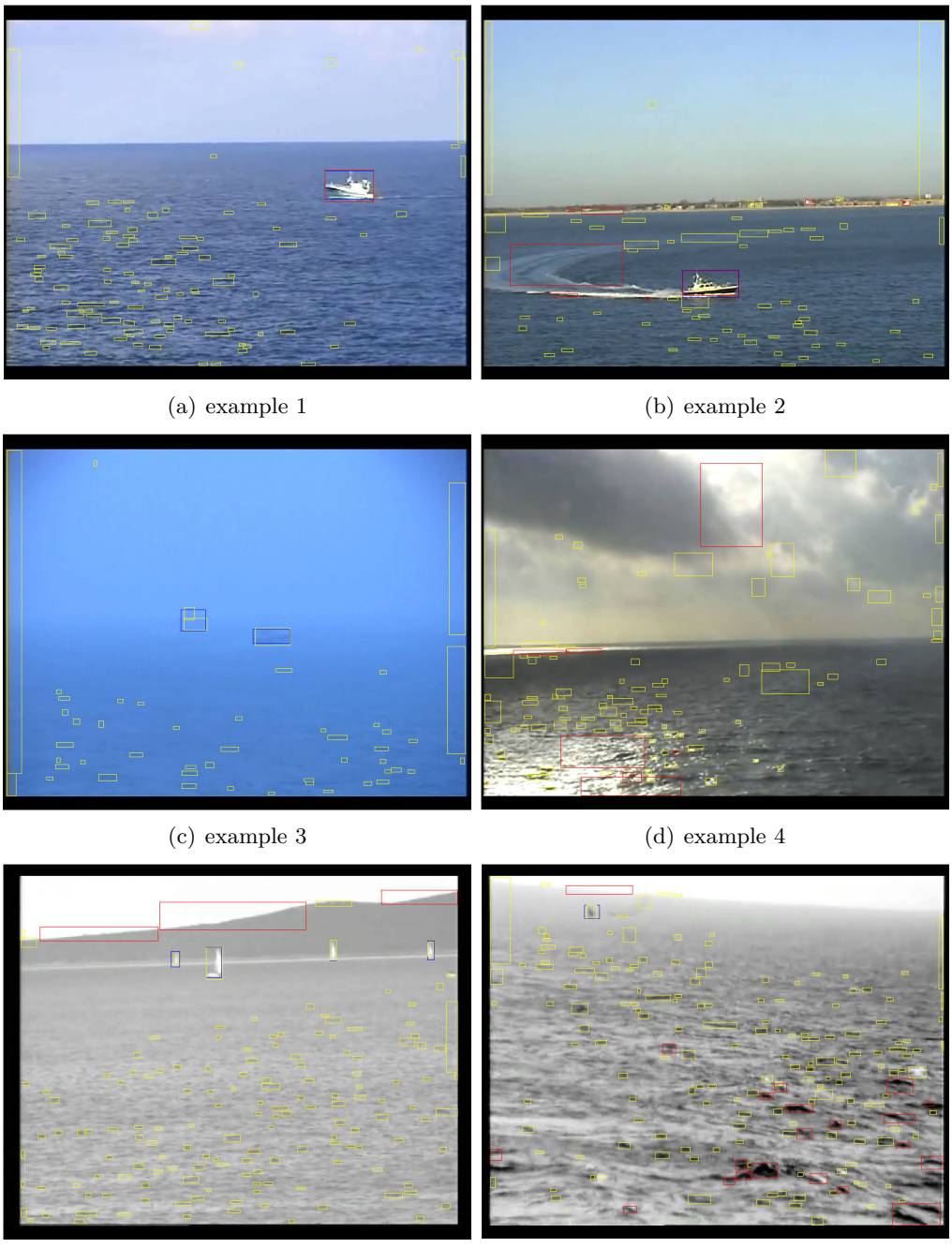


Figure 5-8: Precision-Recall plot for the detection algorithm after classification of the initial detections.

accuracy and negative predictive value. Unfortunately the performance is especially degraded in terms of the recall, which has dropped from 0.810 to 0.333. Also the average overlap ratio has become slightly worse. Although a lot of false positives are eliminated by the classification step, also a lot of true positive detections are classified as false positive. Not exactly the result that was hoped at beforehand and still far from the ideal precision of 1 and recall of 0.810. During the performance analysis it must become clear which factors, if any, have a positive effect on the performance of the classifier, such that both the precision and recall are increased.

Figure 5-9 illustrates the results of the two-stage detection algorithm after classification graphically, and shows some critical factors which are causing false positives to remain and/or true positives to disappear after classification. After visual inspection of the entire dataset, it turned out that especially low contrast objects are not marked as a detection any more after classification, while objects with high contrast most often remain (1-3). Furthermore, it turns out that the majority of false positives after classification are caused by glare/glint and clouds (4), coastal structures and wake (2), sharp transitions between land and see or sky and sea (5,6), and contrast differences caused by waves at the water surface (6). During the perfor-

mance analysis these factors will be taken into account in an attempt to identify possibilities for improvement.



(a) example 1 (b) example 2

(c) example 3 (d) example 4

(e) example 5 (f) example 6

Figure 5-9: Graphical illustration of the results obtained with the two-stage detection algorithm. Blue: ground-truth, Yellow: initial detections, Red: detections after classification.

Chapter 6

Performance Analysis

In this final chapter the performance of the initial two-stage detection algorithm, as described in the previous chapter, will be analysed in more detail. Since the linear discriminant classifier provided the best overall performance, the analysis is continued with this classifier only. During the performance analysis, various parameters of the algorithm are changed in order to determine their influence on the performance. The parameters considered are, in chronological order, the features used as input for the classifier, the evaluation procedure, the size of the area of which the features are calculated and finally fusion of detection bounding boxes before and after classification. In the remainder of this chapter the tests and results corresponding to these parameters are discussed per parameter.

6-1 Features

As described in Section 5-1, during the development of the two-stage detection algorithm an existing set of promising features found in the literature has been used in the classification step. It is generally known that the performance of a classifier strongly depends on the features used. No matter how good a classification algorithm, if the features are bad, so is the performance of the classifier. Therefore, the first parameter considered are the features. Although the initial set of features does not even perform too bad, it is investigated whether additional features can improve the results.

6-1-1 Features of Contrast Enhanced Boxes

During inspection of the output of the initial two-stage detection algorithm it was noticed that especially low-contrast targets were lost. This resulted in the idea to improve the contrast of the detected areas and to determine the features on the contrast enhanced intensities instead of the original intensities. This should ideally result in better separable target and background feature vectors and hence an improvement of the performance of the classifier.

Two contrast enhancement techniques are tested: basic normalization and histogram equalization. If an image is normalized, the lowest occurring intensity is set to 0, the highest occurring intensity to 1 and the intensities in between are linearly transformed between 0 and 1. Histogram Equalization (HE) on the other hand is a technique that spreads the histogram of an input image, such that the intensity levels of the output image span a wider range of the intensity scale and that the intensity levels become more uniformly distributed [25]. The performance of the classifier if the features are calculated on the contrast enhanced boxes instead of the original box are shown in Table 6-1.

Table 6-1: Average AUC values of the LDC, obtained in case the contrast of the detected area is enhanced by means of normalization or by histogram equalization.

	Covariance Matrix	**Concatenated**
Features	**AUC**	**AUC**
Original BB	0.927 (0.041)	—
Normalized BB	0.911 (0.030)	—
Histogram Equalized BB	0.925 (0.025)	—
Original BB + Normalized BB	0.943 (0.033)	0.930 (0.042)
Original BB + Histogram Equalized BB	0.952 (0.025)	0.936 (0.040)

If the results are compared in with the previously obtained results using the features calculated on the original intensities (Original BB), it turns out that the performance of the classifier has not been improved by using normalisation or histogram equalization on the Bounding Box (BB), given the lower average AUC values than 0.927. In an attempt to still improve the classification performance, the features of the contrast enhanced box are added to the features determined on the original box, except for the x- and y-position features because they are the same. For adding the features of the contrast enhanced box to those of the original box, there are two options. First option is to add the features before the local covariance matrix is calculated, which results in a covariance matrix of $9 + 7 = 16 \times 16$, and hence 136 features in total. Second option is to calculate the covariance matrices separately and to concatenate the resulting features, which results in $45 + 28 = 73$ features in total. Both options are tested and the results obtained are shown in Table 6-1 as well. From the results it follows, that adding the features of the contrast enhanced box to those of the original box using either of the two adding options, does improve the classification results. Concatenating the features however does not improve the results as much as adding the features before the covariance matrix is calculated and results in larger variation between the folds. Best improvement is obtained in case the contrast is enhanced using histogram equalization and if the features are added before the covariance matrix is calculated. The average AUC in this case has risen from 0.927 to 0.952 and the variation between the folds is lowest with a standard deviation of 0.025.

If this set of features is applied in the two-stage detection algorithm instead of the original set of features, the performance of the system increases as well. The results are shown in Table 6-2.

The results confirm that contrast enhancement indeed helps to separate the detections which contain target from the detections that only contain background. Compared with the initial set of features the precision has risen with 0.046 to 0.246 and the recall with 0.167 to 0.500. The jump in the recall indicates that a substantial amount of true positives are recovered

Table 6-2: Detection results after classification with the linear discriminant classifier using extra features.

Features	Precision	Recall	Specificity	Accuracy	NPV	Av.OR
Initial	0.200	0.333	0.045	0.165	0.086	0.501
Initial+Hist.Eq.	0.246	0.500	0.036	0.215	0.102	0.526

compared with the initial two-stage algorithm and also more false positives are eliminated. Contrast enhancement by means of histogram equalisation especially recovers some of the objects with lower contrast and is therefore useful to deploy. Although the amount of false positives has been reduced, there are still too many left. Visual inspection of the results shows this is largely due to the sea surface, wake, clouds glare, coastal structures and the coastline/horizon.

6-1-2 Additional Features

Given that there are still too many false positives left, it is investigated whether general features of the bounding box or of the image can contribute in the elimination of these false positives. Two cases are tested. In the first case, a couple of general features of the bounding box are concatenated to the initial features and the features of the histogram equalized bounding box. These additional features are determined on the original bounding box area as well, and are: the mean intensity, standard deviation of the intensity, minimum- and maximum intensity, entropy, contrast, correlation, energy, homogeneity, area of the bounding box, and the x- and y coordinate of the central point of the BB. In total this results in $136 + 12 = 148$ features. Hopefully these general features are different for the two classes and helpfull to separate them better. In the second case, the covariance matrix of the entire image, using the initial set of features, is calculated and concatenated to the features of the bounding box and histogram equalized bounding box. In total this results in $136 + 45 = 171$ features. The idea is that these features of the entire image provide some information about the texture of the image en might help to put the samples in perspective. Since, for example, an image containing a rough sea surface and a cloudy/sunny sky is very different from an image with a smooth sea and fog, it is reasoned that additional information about the image as a whole might be helpful to separate the classes. The test results are shown in Table 6-3.

Table 6-3: Average AUC values of the LDC, obtained in case additional features are added to the features of the original- and histogram equalized box. The value between brackets is the standard deviation between the folds.

Features	AUC
Initial+Hist.Eq.	0.952 (0.025)
Initial+Hist.Eq.+Additional	0.948 (0.031)
Initial+Hist.Eq.+Image	0.963 (0.017)

From the results it follows that adding general features of the bounding box area does not improve the classification performance, given the slightly lower average AUC value. Adding

features of the entire image on the other hand does improve the classification performance with an average AUC value of 0.963 compared to the 0.952 without these features. Also adding these features makes the classification results between the folds more consistent, given the standard deviation of 0.017 which is lower than 0.025. This better performance is probably due the fact the features of the entire image provide information that helps to put the sample in perspective. For example, the features from a sample of a cloudy sky might, without further knowledge, be relatively close the features of a target sample. Additional information about the entire image, including the sky, might help to put these feature vectors further away from each other.

The performance of the two-stage detection algorithm, using the features of the initial and histogram equalized BB and the features of the entire image, are shown in Table 6-4 and Figure 6-1.

Table 6-4: Detection results after classification with the linear discriminant classifier using additional features of the image.

Features	Precision	Recall	Specificity	Accuracy	NPV	Av.OR
Initial+Hist.Eq.	0.246	0.500	0.036	0.215	0.102	0.526
Initial+HE+Im.	0.267	0.527	0.035	0.231	0.099	0.528

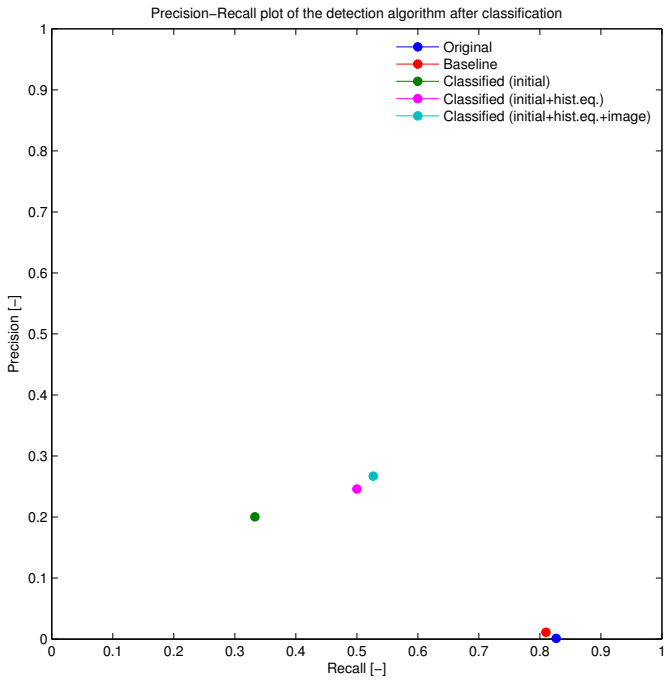


Figure 6-1: Precision-Recall plot for the detection algorithm after classification with new/different features.

As can be seen, the performance has slightly increased compared to the situation without the features of the entire image and provides the best performance so far with a precision of

0.267 and recall of 0.527. Although the precision is still far from 1, the recall is already much closer to the maximum achievable value of 0.810.

Based on the results as reported in this section, a couple of conclusions are drawn. First of all, it is concluded that the initial features are good descriptors to distinguish the two classes, since they are capable of providing high classification performances. Without changing the features themselves an average AUC value of 0.963 is achievable. Secondly, the results also indicate that the covariance matrix is a powerful tool, given that the performance is better when features are added to the covariance matrix than if extra features are concatenated. Finally it is concluded that adding features calculated on the contrast enhanced bounding boxes and of the entire image have a positive impact on the performance of the classifier and detection algorithm. Including some additional features of the detected area did not improve the results, but further research might show that there are features that do improve the results.

6-2 Evaluation Settings

During visual inspection of the results obtained in the previous section, it was noticed that after classification there often are multiple detections that have an intersection with the Ground Truth (GT) of large objects and most often these detections lie completely inside the GT. As stated in Section 4-3, ideally an object is enclosed/marked by a single bounding box. In practice it turns out that this is often the case for small objects, but not for large objects. Consequently, as described in Section 4-3, only one of the boxes with an intersection and a sufficiently Overlap Ratio (OR) will count as True Positive (TP) while the others will be counted as False Positive (FP). Although this seemed a legitimate definition beforehand, it is worthwhile to reconsider, because in practice the extra detections on large objects are not problematic. Consequently, counting these detections as FP is unfairly exacerbating the performance measures. Therefore, it is chosen to count the intersecting box with the largest overlap ratio as a TP and the remaining intersecting boxes as Near True Positive (NTP). Remark that NTP's are not considered during the calculation of the precision, recall and any of the other performance measures. This new definition will definitely have a positive influence on the performance of the detection algorithm, and especially the precision, since the amount of FP will drop to some extent. The recall on the other hand remains the same. Table 6-5 and Figure 6-2 show the results if this new definition is used during the performance evaluation.

As can be seen in Table 6-5 and Figure 6-2, the influence is negligible for the baseline detection algorithm, which is due to the very large amount of false positives. Different becomes the situation for the two-stage detection algorithms after classification. The amount of false positives are significantly dropped compared to the baseline and the effect of introducing NTP's becomes visible. In case of the best found combination of features, the precision increases with 0.057 to 0.324. Furthermore, the amount of NTP's shows that there are many detections present in the images that have an overlap with a GT. Hence quite some objects are not covered by a single box, but by multiple boxes. A post processing fusion step might be used to fuse the detections into more properly enclosing bounding boxes afterwards, but probably better would be a more appropriately clustering procedure for pixels labelled as detection before classification to prevent larger objects to be covered by multiple boxes. All

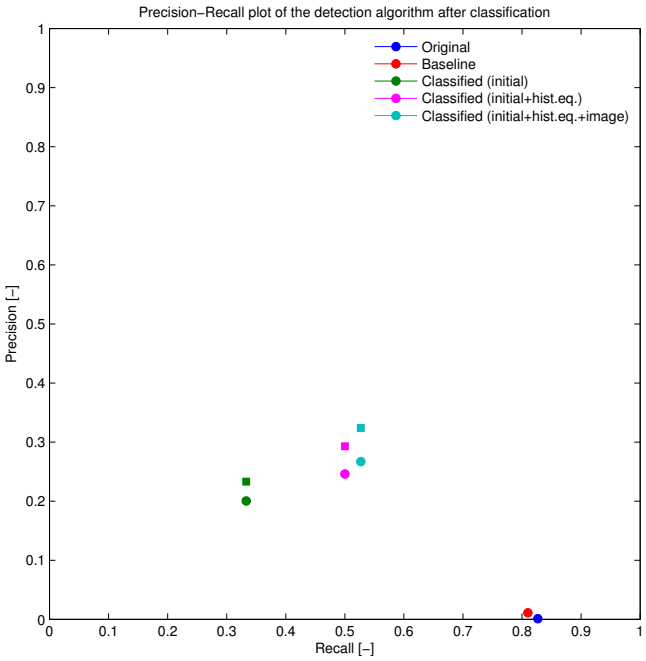


Figure 6-2: Precision-Recall plot for the detection algorithm after changing the evaluation procedure by introducing NTP's. The circles represent the initial results and the squares the results after the new evaluation procedure.

Table 6-5: Original detection results (upper part), and the detection results if detections that intersect with a GT are not counted as FP but as NTP (lower part)

Features	Precision	Recall	Specificity	Accuracy	NPV	NTP
Baseline	0.011	0.810	0.000	0.011	0.000	—
Initial	0.200	0.333	0.045	0.165	0.086	—
Initial+Hist.Eq.	0.246	0.500	0.036	0.215	0.102	—
Intial+Hist.Eq.+Im.	0.267	0.527	0.035	0.231	0.099	—
Baseline	0.011	0.810	0.000	0.011	0.000	2351
Initial	0.233	0.333	0.054	0.183	0.086	239
Initial+Hist.Eq.	0.293	0.500	0.045	0.246	0.102	331
Initial+Hist.Eq.+Im.	0.324	0.527	0.045	0.269	0.099	360

in all the results show that the clustering or fusion procedure (before or after classification) is an important factor with respect to the performance of the algorithm as well. During the remainder of the experiments, the detection algorithm will be evaluated using this new evaluation procedure, since it is believed to better reflect the actual performance of the algorithm.

6-3 Enlargement of the Bounding Boxes

In [8] it is stated that the features are best determined on an area slightly bigger than the object of interest, because this increases the classification performance. As an explanation, it is mentioned that the edges of the object are responsible for this improvenent, and hence they must preferably be fully captured. Therefore, the operator is instructed to draw a box around the object of interest that is slightly bigger than the object itself. Consequently, the object is always enclosed entirely. In this work however, this is often not the case and may the objects be enclosed only partly. Initially, the features are determined strictly on the detected area by the algorithm. It is therefore interesting to determine whether determination of the features on a slightly larger area will improve the classification results in our scenario as well. In order to check this, a couple of tests are performed in which the original boundaries of the detected area are increased with δ pixels in which δ is varied from 1 to 5. The results from these experiments with respect to the classifier and the detection algorithm are shown in Table 6-6 and Table 6-7 respectively.

Table 6-6: Average AUC values of the LDC, in case the boundaries of the detected area are shifted by 'δ' pixels.

δ	**AUC**
0	0.963 (0.017)
1	0.965 (0.011)
2	0.971 (0.012)
3	0.968 (0.017)
4	0.970 (0.015)
5	0.966 (0.021)

Table 6-7: Detection results if the area of which the features are calculated is enlarged by shifting the boundaries with 2 pixels.

Features	**Precision**	**Recall**	**SP**	**AC**	**NPV**	**Av. OR**
Init.+HE+Im.	0.324	0.527	0.045	0.269	0.099	0.528
Init.+HE+Im. ($\delta = 2$)	0.342	0.521	0.053	0.280	0.104	0.525

Compared to using strictly the detected area to classify the boxes, using a larger area does improve the performance of the classifier in our scenario as well. The best performance is obtained when the boundaries are shifted 2 pixels, resulting in an average AUC value of 0.971 compared to an average AUC value of 0.963 if the original area is used. Even so the standard deviation between the folds has lowered from 0.017 to 0.012, which means the results are more consistent. If the results of the detection algorithm are compared, it turns out that the precision has increased from 0.324 to 0.342, but that the recall has decreased from 0.527 to 0.521. Although a few true positives are lost, more false positives are eliminated. Therefore, it is concluded that using a slightly bigger area is indeed beneficial. Furthermore, the results indicate that larger values for δ than 2, do not cause a further improvement of the classifier. This is most likely due to the fact that a larger proportion of the area in the box is background instead of target for target sampes, which reduces the performance of the classifier.

6-4 Fusion of Bounding Boxes

As stated earlier, larger objects are at the moment often covered by multiple boxes. As a solution it is suggested to fuse bounding boxes after classification, or to use a clustering procedure for detected pixels that performs better than the straightforward one currently used. To get a grasp of the improvement this might cause, it is decided to test fusion of the detections after classification. Due to time constraints it is impossible to implement and test other clustering procedures. This fusion of the bounding boxes will certainly improve the results to some extent, but will also result for the algorithm to be less discriminative between targets that are close to each other. Instead of two separate boxes, they might become fused into one. In this work it is chosen to fuse boxes for which the boundaries are within 5 pixels of each other. This value of 5 is chosen more or less arbitrarily for illustration purposes only. The results of this fusion procedure for the last two settings of the two-stage detection algorithm are shown in Table 6-8 and Figure 6-3.

Table 6-8: Original detection results (upper part) and detection results if detections within 5 pixels away from each other are fused (lower part).

Features	Precision	Recall	SP	AC	NPV	Av. OR
Init.+HE+Im.	0.324	0.527	0.045	0.269	0.099	0.528
Init.+HE+Im. ($\delta = 2$)	0.342	0.521	0.053	0.280	0.104	0.525
Init.+HE+Im.	0.388	0.516	0.060	0.304	0.097	0.541
Init.+HE+Im. ($\delta = 2$)	0.376	0.514	0.061	0.298	0.103	0.542

From these results it follows that fusion has largely a positive effect on the performance of the detection algorithm, given the significant improvement in the precision and minor decrease in recall. A logical result, since false positive detections close to each other are fused to a single false positive. For this improvement a small price is payed with respect to the recall. Remarkable is that the best performance occurs in the case that the features are determined on the original bounding box area with a precision of 0.388 and recall of 0.516 and not in case the features are determined on a larger area. Apparently, the fusion procedure cancels the benefits of determining the features on a larger area. Most likely this is because more false positives can be fused. For example, if the features are determined on strictly the detected area would result in 4 false positives, while if the features are determined on a larger area in 3 of the 4 false positives, the performance of the latter case is better. But, if the detections are fused it might very well be that the 4 boxes can be fused into one FP, while the 3 boxes only into two FP's because the linking box is missing. Then after fusion the former case performs better. However, in both cases the results show that fusion can be used to optimize the detection output and that it provides boxes that are better covering/enclosing the objects, given the higher average overlap ratio's. Although fusion of bounding boxes is not really a parameter of the detection algorithm itself, but in fact a post processing step, it underlines the importance of the initial clustering of the detected pixels and confirms that this parameter deserves more attention.

Finally, with a best obtained precision and recall of 0.388 and 0.516 respectively, it must be concluded that a classifier based system itself does not provide a complete solution in the quest of elimination of false detections while maintaining true detections. If this final result is

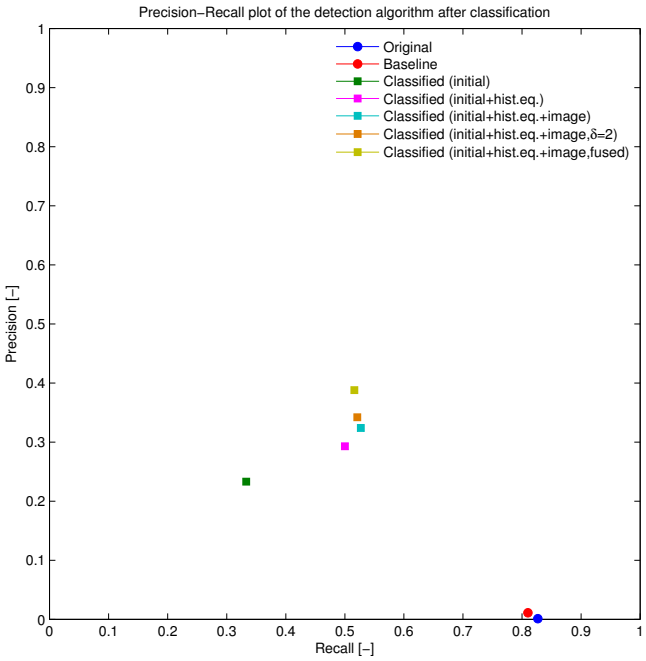


Figure 6-3: Precision-Recall plot for the detection algorithm after fusion of the bounding boxes.

compared with the precision and recall of 0.011 and 0.810 of the baseline detection algorithm, it turns out that a step in the right direction has been made. A lot of false detections are eliminated, but unfortunately 30% of the true detections are eliminated as well. Regrettably there are still too many false positives left at the moment for the detection algorithm to be of practical use for unsupervised surveillance. Usage as support for the crew however is starting to become within reach and a more realistic option.

Chapter 7

Conclusions and Recommendations

Robust automatic object detection in Electro-Optical (EO) camera images of a maritime environment is becoming an increasingly important issue for the Royal Netherlands Navy (RNLN). Automated object detection in the maritime environment is a complex problem however, due to various complicating factors. These factors include the highly dynamic background, camera movement, the variety in possible objects and their appearance, and the diverse meteorological/environmental circumstances. As a starting point a detection algorithm based on polynomial background estimation has been developed. Although this approach is capable of detecting possible objects of interest, it also produces a huge amount of false detections. The main research goal of this thesis project was to find out whether a system that learns from examples can be used to eliminate these false detections, while maintaining the true detections.

First of all a dataset consisting of eight hundred images has been constructed and the original detection algorithm has been optimised in order to detect as many objects as possible. Secondly, a set of six different standard classifiers has been tested using samples of false- and true detections and a set of features found in the literature. Finally, the performance of the newly obtained two-stage detection algorithm has been further analysed by means of various tests.

Of the tested classifiers, it turned out that the linear discriminant classifier provides the best performance and is most appropriate to use. Although the initial classifier eliminates a substantial amount of the false detections it also eliminates a significant part of the true detections. Most true detections that are lost are due to low contrast while false detections that remain are largely due to clouds, glare, coast lines and/or coastal structures, wake and rough sea surfaces with high contrast differences. In additional tests it is found that the features used and the clustering procedure are important factors with respect to the performance of the algorithm. Although the initial features found in the literature have proven to be good descriptors, it is beneficial to add more features. Especially adding the same features of the entire image and of the contrast enhanced versions of the detected areas by means of histogram equalization improved the results. In this case less true detections are lost and even more false detections are eliminated. Future research should point out whether

adding other features can improve the results even further. Furthermore, it is expected that a better clustering procedure will improve the results as well. With a best obtained precision of 0.388 and recall of 0.516 it is concluded that our current classifier based system is at the moment not nearly able of fully eliminating the false detections while remaining the true detections and that further research is required. Although usage for unsupervised surveillance is still far away, usage of the two-stage detection algorithm as support for the crew is becoming a realistic option.

Besides additional features and a more appropriate clustering procedure there are a couple of other factors marked for future research. First of all, in this work only one classifier was trained for the entire image. It is expected that a separate classifier for the sea- and air part would allow to improve the results significantly. In this case the horizon must be known however. To this end it would be best to use information from the camera and sensors available aboard the ship, such as compass and stability sensors, to calculate the position of the horizon in the image. Also a horizon detection algorithm might be used to determine the position of horizon, but this is believed to be less reliable. Another advantage of knowing the position of the horizon is that the results can already be improved by discarding the detections that occur in for example the air part, if air objects are not of interest, since clouds are often causing false detections. Also it is noticed that the results could probably be improved by using a more flexible classifier than the linear discriminant classifier, such as a k-nearest neighbour classifier. Due to a lack of memory we were not able to test this classifier on the large dataset, but given the better performance on the smaller dataset it is expected that if it would be possible to use this classifier, the performance would be better than obtained with the linear discriminant classifier. Whether adding more samples, especially target samples, would improve the classification results is a point for future research as well. Since using more background samples caused an improvement, it might very well be that this is also the case if the amount of target samples is increased.

Second point for further research is to involve the time dimension. In this research the detections are determined per frame and it is expected that if multiple frames are used before the detections are returned, this will highly contribute to the elimination of false positives. A simple possibility would be to count the presence of a detection in multiple frames and to return a detection only in case it occurs in the majority of the frames considered. This way false detections due to waves, wake and glare will probably be eliminated due to their irregular behaviour. More complex would be to track the detections and determine their class based on the tracking record.

Third point for future research is to optimise the operating point of the classifier by hand or through a cost function. During this project, the operating points of the classifiers were set automatically by the algorithms of the classifiers. By changing the operating point of the classifier, it would be possible to maintain more true detections at the expense of a loss in the elimination of false positives. Maybe an optimum between the loss in elimination of false detections and the gain in maintaining true detections can be found either by hand or through a cost function that penalizes the loss in true detections.

Finally preprocessing, which is not regarded in this work, is marked as a point for future research as well. The effect of, for example, smoothing the image with an averaging filter of some kind may reduce the initial amount of false detections, without causing a significant loss in true detections. Since smoothing affects peaks in the intensity, it may be helpful to

prevent some of the false detections caused by the sea surface, glare, and clouds due to the sharp intensity transitions that are characteristic for these areas. On the other hand, the intensities at the edges of objects of interest are affected as well, which may exacerbate the detection capabilities. Especially very small targets of only a few pixels are less likely to be detected after smoothing. However, given the huge amount of initial false detections, it would be highly beneficial if this amount can be reduced beforehand. Further research should show whether preprocessing allows a reduction in false detections at an acceptable loss of true detections.

As a final remark we would like to point out that the automatic object detection algorithm as presented in this work is highly generic and could be applied to Visible Light (VL) as well as Infrared (IR) images. In this work it is only tested on colour- and gray-scale VL images, but it would be highly interesting to see whether the results obtained in this work would change drastically if the algorithm is applied to IR images. Furthermore, it would be valuable for the RNLN to determine which type of camera (VL or IR) provides better detection probabilities in combination with the detection algorithm as proposed in this work. This would require a dataset with images from both types of cameras, captured at the same time and with identical focus, such that identical images are obtained except for the fact that they originate from different sensors. Such a dataset is not available at the moment unfortunately. Even so would it be interesting to see whether the resolution of the images has an impact on the performance as well, by testing it on higher-resolution images than the images used in this work (720x576 pixels). An example would be to test it on data originating from the Gatekeeper EO sensor system aboard the OPV's of the Holland class, which should soon become available. Expected is that the algorithm should perform comparably on these high-resolution images, but that the clustering parameter for detected pixels, δ, must be optimised again.

Appendix A

Practical Implemention of the Detection Algorithm

The detection principles as described in Section 4-2 are implemented in a six step detection algorithm, as visualized in Figure A-1. The first and second stage are preprocessing steps, which make sure that an image is border free and in grey-scale format. Steps 3 till 5 subsequently are the actual detection steps and the sixth and last step is a post processing step in which the detected pixels are clustered into bounding boxes enclosing the detected areas. Per stage a more detailed description will be given in the remainder of this section. At the Centre for Automation of Mission-Critical Systems (CAMS) - Force Vision an algorithm based on polynomial background estimation was already available. Some steps are copied from this Matlab script, while others are added or rewritten in order to run more time efficiently.

1. Black Border Removal
   As can be seen in Figure 2-3 and Figure 4-1, the images in the dataset contain a black border. Before the image is converted to grey-scale and the polynomials are fit, it is important that the border is removed first or will be ignored. The border is not part of the actual image and it will have a negative influence on the conversion to a grey-scale image, as well as that it will cause huge artefacts in the fitted polynomial, if it would not be removed. Unfortunately, the size of the border is not constant over all images in the dataset. Therefore, the images cannot be cropped using fixed parameters. Instead it is chosen to determine the parameters of the border automatically using a Canny edge detector. The parameters found are used to ignore the border during the rest of the steps and are stored in an ASCII file so they can be quickly loaded if the image is used again. This step is added to the original algorithm.

2. Grey-Scale Conversion
   After the border is removed, the image is converted to a grey-scale image. Conversion from colour to grey-scale is performed in two steps. First, each channel of the colour image is normalized separately, using 'mat2gray', in which the minimum occurring value is set to zero and the maximum occurring value to 1. Subsequently the normalized colour

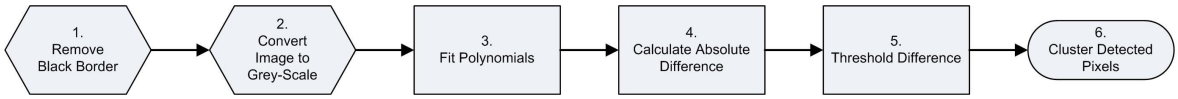
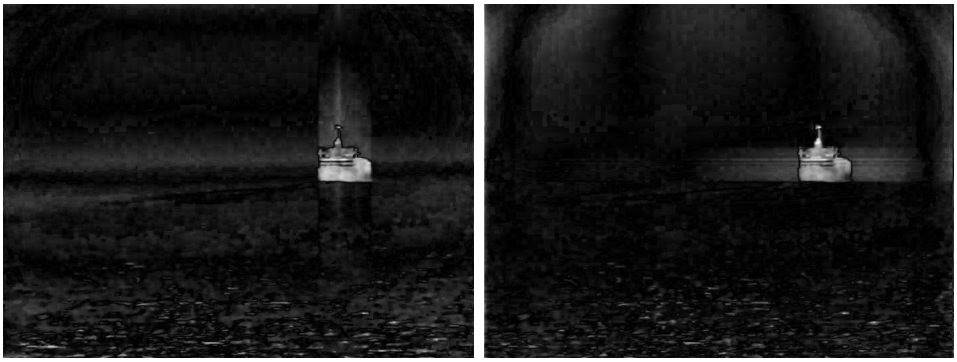


Figure A-1: Work flow of the detection algorithm based on polynomial background estimation.

image is converted to grey-scale, using 'rgb2gray', by eliminating the hue and saturation information while retaining the luminance. In case a grey-scale image is used, it will only be normalized using 'mat2gray'. This conversion scheme is copied from the original algorithm.

3. Polynomial Fit
   After border removal and conversion to a grey-scale image, polynomials are fitted to the intensity values of each column in accordance with (4-1). It is chosen to use polynomials of the third order, since in [5] it is shown that this order for the polynomials provide the best performance. The code for this step and the two steps hereafter originates from the original script, but is rewritten in order to run more time efficiently. The direction in which the polynomials are fitted may be vertical (per column) as originally described, but may also be horizontal (per row) and yield different results. Which direction is best to use is determined in Section 4-4.

4. Absolute Difference
   The polynomials that are fitted in the previous step form an estimated background image and the difference between this and the original image is calculated in accordance with Equation (4-2). Examples of two such difference images are shown in Figure A-2. This figure also clearly illustrates that the direction in which the polynomials are fitted yields different results.



(a) Vertical Polynomial Fit (b) Horizontal Polynomial Fit

Figure A-2: Two examples of absolute difference images. Left: resulting difference image if the polynomials are fitted per column. Right: resulting difference image if the polynomials are fitted per row. Both difference images were normalized in order to be viewable.

5. Thresholding
   Once the difference image is obtained, it is thresholded using Equations (4-3) and (4-4). The result is a logical matrix, which contains ones at the positions of pixels labelled as a detection and zeros at the positions of pixels labelled as background. Like the

fit direction of the polynomials, also the direction in which the detection thresholds are determined will yield different results. Which direction is best to use must be determined during optimisation of the algorithm. Furthermore also the optimal value of the detection threshold parameter, α, will have to be determined (see Section 4-4).

6. Clustering
   Finally the pixels labelled as a detection are clustered into bounding boxes, which enclose the areas in which detected pixels occur. The procedure used for clustering is as follows. Of the detected pixels, the pixel with the lowest x and y value is chosen as starting point and enclosed by a Bounding Box (BB). Detected pixels that are within $\pm\delta$ pixels from the edges of the detected area, in this case a single pixel, are considered to belong to the same cluster and are included. The parameters of the BB are updated accordingly to enclose the new cluster of detected pixels and again the algorithm searches for pixels that are within $\pm\delta$ pixels from the edges of the updated detected area. This procedure continues until there are no pixels left in the vicinity of the current detected area. If this is the case then, of the remaining detected pixels, the detected pixel with lowest x and y value is used as new starting point and the procedure as described above is repeated and continues until all detected pixels are clustered. In the end, the output of the object detection algorithm is the set of bounding boxes found after clustering of the detected pixels. During optimisation of the algorithm it must be determined which value for the clusterings parameter, δ, yields the best results (see Section 4-4).

Bibliography

[1] M. Hartemink, "Automatic object detection in a maritime environment." Literature Survey, February 2012.

[2] F. Bolderheij, *Mission-Driven Sensor Management: Analysis, Design, Implementation and Simulation*. TUDelft, 2007.

[3] http://www.naval-technology.com/projects/dezeven/dezeven7.html. [Last accessed on August 22nd, 2012].

[4] W. Qiying, C. Huali, D. Xiaofeng, W. Mingfen, and J. Taisong, "Real-time moving maritime objects segmentation and tracking for video communication," in *Communication Technology, 2006. ICCT '06. International Conference on*, pp. 1 –4, November 2006.

[5] T. Y. C. van Valkenburg-van Haarst, F. Bolderheij, and F. C. A. Groen, "Automatic detection in a maritime environment: gradient filter versus intensity background estimation," vol. 6967, no. 1, p. 69670V, 2008.

[6] G. Santhalia, N. Sharma, S. Singh, M. Das, and J. MulChandani, "A method to extract future warships in complex sea-sky background which may be virtually invisible," in *Modelling Simulation, 2009. AMS '09. Third Asia International Conference on*, pp. 533 –536, May 2009.

[7] H. Wei, H. Nguyen, P. Ramu, C. Raju, X. Liu, and J. Yadegar, "Automated intelligent video surveillance system for ships," vol. 7306, no. 1, p. 73061N, 2009.

[8] S. Çakır, T. Aytaç, A. Yıldırım, and O. N. Gerek, "Classifier-based offline feature selection and evaluation for visual tracking of sea-surface and aerial targets," *Optical Engineering*, vol. 50, no. 10, p. 107205, 2011.

[9] M. D. R. Sullivan and M. Shah, "Visual surveillance in maritime port facilities," in *Visual Information Processing '08*, pp. –1–1, 2008.

[10] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art," *Recent Patents on Computer Science*, vol. 1, pp. 32–54, 2008.

[11] 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.

[12] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1219 –1225, October 2009.

[13] D. Zhou, L. Wang, X. Cai, and Y. Liu, "Detection of moving targets with a moving camera," in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pp. 677 –681, December 2009.

[14] B. Qi, T. Wu, H. He, and T. Hu, "Real-time detection of small surface objects using weather effects," in *Proceedings of the 10th Asian conference on Computer vision - Volume Part III*, ACCV'10, pp. 27–38, Springer-Verlag, 2011.

[15] T. Y. C. van Valkenburg-van Haarst and K. A. Scholte, "Polynomial background estimation using visible light video streams for robust automatic detection in a maritime environment," vol. 7482, no. 1, p. 748209, 2009.

[16] T. Y. C. van Valkenburg-van Haarst, A. V. van Leijen, and F. C. A. Groen, "Colour as an attribute for automated detection in maritime environments," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pp. 1679 –1686, July 2009.

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

[18] J. Nascimento and J. Marques, "Performance evaluation of object detection algorithms for video surveillance," *Multimedia, IEEE Transactions on*, vol. 8, pp. 761 –774, August 2006.

[19] F. Bashir and F. Porikli, "Performance evaluation of object detection and tracking systems," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2006)*, June 2006.

[20] L. M. Brown, A. W. Senior, Y. li Tian, J. Connell, A. Hampapur, C. fe Shu, H. Merkl, and M. Lu, "Performance evaluation of surveillance systems under varying conditions," in *In: Proceedings of IEEE PETS Workshop*, pp. 1–8, 2005.

[21] K. Smith, D. Gatica-Perez, J. Odobez, and S. Ba, "Evaluating multi-object tracking," in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, p. 36, June 2005.

[22] X. Desurmont, C. Carincotte, and F. Brémond, "Intelligent video systems: A review of performance evaluation metrics that use mapping procedures," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 127 –134, September 2010.

[23] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 319 –336, February 2009.

[24] N. Acito, A. Rossi, M. Diani, and G. Corsini, "Optimal criterion to select the background estimation algorithm for detection of dim point targets in infrared surveillance systems," *Optical Engineering*, vol. 50, no. 10, p. 107204, 2011.

[25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Prentince Hall, third ed., 2008.

[26] A. R. Webb, *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, October 2002.

[27] F. van der Heijden, R. Duin, D. de Ridder, and D. M. J. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*. Wiley, 1 ed., November 2004.

[28] D. Tax, "Ddtools, the data description toolbox for matlab," May 2012. version 1.9.1.

[29] "Lecture slides week 8, pattern recognition course," 2010. Delft Univerity Technology.

[30] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2010.

[31] R. Collins, A. Lipton, and T. Kanade, "A system for video surveillance and monitoring," in *American Nuclear Society 8th Internal Topical Meeting on Robotics and Remote Systems*, 1999.

[32] X. Wang and T. Zhang, "Clutter-adaptive infrared small target detection in infrared maritime scenarios," *Optical Engineering*, vol. 50, no. 6, p. 067001, 2011.

[33] Y. Gao, S. Hu, Z. Miao, and S. Xu, "Research on seaskyline detection in complex sea background," in *Innovative Computing, Information and Control, 2007. ICICIC '07. Second International Conference on*, p. 452, September 2007.

[34] Z. Ji, Y. Su, J. Wang, and R. Hua, "Robust sea-sky-line detection based on horizontal projection and hough transformation," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pp. 1 –4, October 2009.

[35] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, pp. 62 –66, January 1979.

[36] J.-W. Lu, Y.-Z. Dong, X.-H. Yuan, and F.-L. Lu, "An algorithm for locating sky-sea line," in *Automation Science and Engineering, 2006. CASE '06. IEEE International Conference on*, pp. 615 –619, October 2006.

[37] J. Wu, S. Mao, X. Wang, and T. Zhang, "Ship target detection and tracking in cluttered infrared imagery," *Optical Engineering*, vol. 50, no. 5, p. 057207, 2011.

[38] C. Zhang, S.-C. Chen, M.-L. Shyu, and S. Peeta, "Adaptive background learning for vehicle detection and spatio-temporal tracking," in *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, vol. 2, pp. 797 – 801 vol.2, December 2003.

[39] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *Image Processing, IEEE Transactions on*, vol. 13, pp. 1459 –1472, November 2004.

[40] A. Borghgraef, O. Barnich, F. Lapierre, M. Van Droogenbroeck, W. Philips, and M. Acheroy, "An evaluation of pixel-based methods for the detection of floating objects on the sea surface," *EURASIP J. Adv. Signal Process*, vol. 2010, pp. 5:1–5:7, January 2010.

[41] J.-W. Lu, J.-C. Ren, Y. Lu, X.-H. Yuan, and C.-G. Wang, "A modified canny algorithm for detecting sky-sea line in infrared images," in *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on*, vol. 2, pp. 289 –294, October 2006.

[42] J.-H. Park, K.-G. Nam, and J.-H. Joo, "A partially occluded sea-sky line detection algorithm," *Image Processing, Computer Vision, and Pattern Recognition (IPCV'11), The 2011 International Conference on*, vol. II.

[43] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981.

[44] S. Fefilatyev, D. Goldgof, and C. Lembke, "Tracking ships from fast moving camera through image registration," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 3500 –3503, August 2010.

[45] T. Can, A. O. Karali, and T. Aytaç, "Detection and tracking of sea-surface targets in infrared and visual band videos using the bag-of-features technique with scale-invariant feature transform," *Appl. Opt.*, vol. 50, pp. 6302–6312, November 2011.

[46] M. Hartemink, "Automatic object detection in a maritime environment; the influence of preprocessing on object detection in electro-optical camera images." BSc Thesis, March 2010.

[47] H. Bouma, D.-J. J. de Lange, S. P. van den Broek, R. A. W. Kemp, and P. B. W. Schwering, "Automatic detection of small surface targets with electro-optical sensors in a harbor environment," vol. 7114, p. 711402, SPIE, 2008.

[48] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *In Proc. 9th European Conf. on Computer Vision*, pp. 589–600, 2006.

[49] A. Toet, *Detection of dim point targets in cluttered maritime backgrounds through multisensor image fusion*, vol. 4718, pp. 118–129. The International Society for Optical Engineering, 2002.

Glossary

List of Acronyms

AC	Accuracy
ACE	Average Correlation Energy
ACH	Average Correlation Height
ASM	Average Similarity Measure
AUC	Area Under the ROC Curve
BB	Bounding Box
BBs	Bounding Boxes
C2	Command and Control
CAMS	Centre for Automation of Mission-Critical Systems
COP	Common Operational Picture
DoD	Department of Defence
EO	Electro-Optical
FISHERC	Fisher Classifier
FN	False Negative
FP	False Positive
GT	Ground Truth
HE	Histogram Equalization
HFHT	Horizontal Fit, Horizontal Threshold
HFVT	Horizontal Fit, Vertical Threshold

HNLMS	Her Netherlands Majesty
IFF	Identification Friend or Foe
IR	Infrared
KNNC	K-Nearest Neighbour Classifier
LCF	Air Defence and Command Frigate
LDC	Linear Discriminant Classifier
LMSO CV	Leave Multiple Sets Out Cross-Validation
MACH	Maximum Average Correlation Height
MoG	Mixture of Gaussians
NMC	Nearest Mean Classifier
NPV	Negative Predictive Value
NTP	Near True Positive
ONV	Output Noise Variance
OPV	Oceangoing Patrol Vessel
OR	Overlap Ratio
OT	Overlap Threshold
PARZENC	Parzen Classifier
PDF	Probability Density Function
PR	Precision
QDC	Quadratic Discriminant Classifier
RCS	Radar Cross-Section
RHIB	Rigid Hull Inflatable Boat
RNLN	Royal Netherlands Navy
ROC	Receiver Operating Characteristic
ROI	Region(s) Of Interest
SP	Specificity
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

VFVT	Vertical Fit, Vertical Threshold
VFHT	Vertical Fit, Horizontal Threshold
VL	Visible Light

List of Symbols

α	Predefined threshold parameter
δ	Predefined clustering parameter
Ω	Subset of pixels
$\tilde{I}(x, y)$	Estimated background image
$\underline{S}_k$	Feature vector
E	Classification error
$I(x, y)$	Gray-scale input image
$I_{diff}(x, y)$	Absolute difference image
M_{loc}	Local covariance matrix
N_e	Number of erroneously classified samples
N_s	Number of samples
P	Class prior
r_i	Residue at pixel (x_i, y_i)
T_f	Feature tensor
T_{high}	Upper threshold
T_{low}	Lower threshold
D(x,y)	Logical detection matrix