DTU

Detection and classification of objects at sea using computer vision

Juan Mollá Ramírez (s161539)
Master of Science in Electrical Engineering
2018

DTU Electrical Engineering
Department of Electrical Engineering

Detection and classification of objects at sea using computer vision

Report written by:
Juan Mollá Ramírez (s161539)

Advisor(s):
Mogens Blanke, Professor at the Electrical Engineering Department of DTU
Søren Hansen, Associate Professor at the Electrical Engineering Department of DTU
Jonathan Dyssel Stets, Postdoc at the Department of Applied Mathematics and Computer Science of DTU

DTU Electrical Engineering
Technical University of Denmark
2800 Kgs. Lyngby
Denmark

s161539@student.dtu.dk

Project period:	22. January- 22. June
ECTS:	30
Education:	M.Sc.
Field:	Electrical Engineering
Class:	Public
Edition:	1. edition
Remarks:	This report is submitted as partial fulfillment of the requirements for graduation in the above education at the Technical University of Denmark.
Copyrights:	©Juan Mollá Ramírez, 2018

Abstract

The project of this paper presents four different approaches for object detection and classification at sea using computer vision. Recent years have seen a large increase in the use of optical detection and tracking methods in autonomous cars. However, this is challenging in maritime environments since objects at sea can be hard to distinguish from the waterline or hidden waves. Furthermore, cameras have a limited range and objects far from the camera might be impossible to detect. In this paper detection of marine vessels and aids for navigation (buoys, beacons...) are the main priority. This project is only focused on the detection performance and the evaluation of the algorithms present.

Preface

I would like to thank my supervisors Mogens Blanke, Søren Hansen and Jonathan Dyssel for all the time, patience and for always being willing to have a discussion no matter if my questions were good or not. As a result of that, I was able to learn a great deal of new skills and fully appreciate five months of working on my master thesis project. It was a tremendous journey.

Many thanks to all the fantastic friends that I had a chance to meet during my master studies in Denmark. You made this period great for me, I feel lucky to have encountered you in my life. Moreover, thanks to all my friends from Spain. You proved that true friendships last no matter of the distance or few chances to see each other.

Last but not least, I would like to thank all my family for enormous support of all kind that I got during my whole life from you, especially during master studies. There are no words to describe how grateful I am for all the love, wisdom and great memories that you gave me. Gràcies per tot!

Kongens, Lyngby, June 28, 2018

Juan Mollá Ramírez (s161539)

Contents

List of Figures

List of Tables

Acronyms

ACF Aggregate Channel Features.

AIS Automatic Identification System.

CNN Convolutional Neural Network.

DCT Discrete Cosine Transform.

ECDIS Electronic chart display and information system.

ENC Electronic Navigational Chart.

EO Electro-optical.

FMCW Frequency Modulated Continous Wave.

GMM Gaussian Mixture Model.

GST Danish Geodata Agency.

HSV Hue Saturation Value.

IHO International Hydrographic Organization.

IHO International Maritime Organization.

IMU Inertial Measurement Unit.

NIR Near Infra-Red.

RCNN Region-based Convolutional Neural Network.

RGB Red Green Blue.

RoI Region of Interest.

RPN Region Proposal Network.

SIFT Scale Invariant Feature Transform.

SURF Speeded Up Robut Features.

SVM Support Vector Machine.

UAV Unmanned Aerial Vehicles.

USV Unmanned Surface Vehicles.

VMS Vessel Monitoring System.

CHAPTER 1

Introduction

1.1 Motivation

Nowadays, improvements in technology let computers assist in most of the duties made by humans. The integration of sensor systems and advanced algorithms to manage data let the technology go in a way where tedious and dangerous tasks carried out by humans can be replaced by robots. Automated technology systems are aimed at taking the role of the eyes and brain of an operator for controlling an automated vehicle. Automated cars, Unmanned Aerial Vehicles (UAV), Unmanned Surface Vehicles (USV) and automated industrial systems are examples where such technologies are applicable.

In the case of ship navigation, the level of autonomy within marine operations is increasing, and this trend is expected to continue in the future [20, 23, 32]. Many experts foresee that unmanned and autonomous ships will gradually replace manned ships and become a key technology of safe, cost-effective and environmentally friendly marine transportation. Autonomy in ship navigation would lead to reduction in crew numbers as a result of re-skilling and relocation of crew to the shore, potentially resulting less vigilant look-out [25]. The main benefits could be improved safety and reduction of operating costs. Also better energy efficiency and protection of environment support the idea of using unmanned ships for transportation of goods and raw materials over longer distances [1]. Especially in Denmark, a country consisting on more than 70 populated islands, food and goods transport and means of travel for commuters are important.

As the applications of automated and unmanned systems have grown in recent years, the need for precise and robust guidance and navigations has increased as well. For instance, technologies that have been traditionally deployed for military purposes, eg. radars and sonars, now are found to be of immense utility in providing support for navigation as well[25]. Therefore, surveillance systems play an important role in management and monitoring of littoral and maritime areas in providing tools for situation awareness, threat assessment, and decision making [10].

Several sources of surveillance, monitoring, maritime safety information are available. These include Automatic Identification System (AIS), Vessel Monitoring System (VMS), ship- and land-based radars, air- and space-borne optical sensors and harbour-based visual surveillance. Therefore, it is important to use this situational awareness sensors for better collision avoidance and navigation. Integration of several information sources has been developed during the last decade. This integration from different sources, also known as sensor fusion, aims to obtain a lower detection error probability and a higher reliability by using data from multiple distributed sources [6]. Example of this include systems that integrate AIS/VMS with SAR-imagery [18, 26], radar- and visual-based surveillance for ports [29, 27], land-radars with visual information from air-borne platforms, and ship-based systems that integrate visual and other sensor's information [19, 36].

Autonomous technology on-board ships will require that computer interpreted sensors information is made available to navigation algorithms. These navigation algorithms should deal with a proper navigation which is mainly dependent on the right answers to the following questions:

- Where am I?
- What are the local conditions at this position and in the vicinity?
- Which path should I follow to reach my destination?

Furthermore, collision avoidance is crucial for safe navigation. An autonomous vessel, which should perform autonomous operations, will require local situation awareness by sensing the immediate environment to avoid collision with other ships ot with other traffic areas of the sea bed that are too shallow to allow safe passage on the sea surface. The International Regulations for Preventing Collisions at Sea 1972 (COLREGs) requires all ships to be equipped with radars for proper lookout to provide early warning of potential collision [25]. Therefore, sensor fusion will result in a better situation awareness so the 2^{nd} question above will help decision support on-board.

Beside sensors, electronic sea charts are used to enhance local situation awareness. In the past, knowledge and information about the sea was storage in paper sea charts. These charts helped seafarers navigate with the information available in that chart. Nowadays all that information has been collected in what is called Electronic Navigational Chart (ENC). They contain all geographic, hydrographic, and geophysical information for the area, the sea traffic arrangements, and the administrative regulations that are also shown on paper charts as well as described and illustrated in relevant printed nautical publications. Therefore it is important to distinguish between paper and electronic charts. Paper nautical charts contain a fixed amount of information, while electronic chart data can contain a far greater amount of data [33].

Unlike land-based navigation technology, marine equipment such the ENC undergoes strict supervision by national and international bodies. The shipping nations of the world cooperate based on regulations they have agreed under the umbrella of the International Maritime Organization (IHO) . This means that the resources to obtain the information have to be official.

1.2 Scope of the project

This thesis deals with object detection and classification at sea. Having detected an object, its nature is subsequently determined. As a third step, algorithms determine to which category the object belongs.

One category is objects that inform about risks or regulations in the area. These are traffic marks of the seas, implemented as buoys anchored to the seabed. Another category is floating objects that own vessel must avoid. Depending on relative speed, location and direction of travel, objects in this category should influence own vessel's navigation.

The thesis employs computer vision techniques to observe objects at sea.

A sea chart informs on positions and nature of all traffic marks at the sea and correlating information in an electronic sea chart with observations that we consider to be buoys, could confirm the correctness of navigation. Such correlation could also confirm hypotheses that objects in category 2 are indeed moving objects.

The original aim of the project was to supplement object detection and classification with fusion with sea chart information to obtain a second confirmation of observations and classifications. Electronic sea charts should be computer readable, i.e. the thesis project should need unencrypted information. Such S57 charts are only available from the Danish Geodata Agency (GST), and agreements on terms between GST and DTU were finalised so late that it was not possible to include electronic sea chart information within the timeframe of the thesis project.

Instead, the project was re-formulated such that the scope of the thesis includes a chapter on how the information fusion with sea chart data can be done when they become available. The focus of the thesis is therefore to detect and classify objects at sea as observed by on-board cameras. The scope is as follows

- Literature review of relevant computer vision methods.
- Assess existing methods for use in a maritime environment.
- Implement promising methods and evaluate their features and performance on images from sea.
- Analyse the performance of algorithms in terms of correct and false detections
- Draft the theory and methodology to include electronic sea chart information when this becomes available

The thesis selected four algorithm-based methods for object detection and employed convolutional neural network methods for classification. The thesis provides a thorough testing of the selected methods, all of which were validated on the entire set of photos.

Processing of radar data and correlation with camera observations is not within the scope of this thesis but was formulated by DTU as a companion thesis.

Eduardo Vasquez Salvador, who focused on other computer vision methods, writes this parallel thesis. Small parts of the work was done jointly to save time and efforts, and the thesis shows clearly where joint work was done or findings by Eduardo are referred to.

1.3 Outline of the thesis

This thesis is organized in 7 chapters, as well as a appendix covering some implementation aspects. The chapters are numbered 1-7. Here is a short description of the contents of each chapter:

- Chapter 2 describes the global system and the main resources used in the project. Further, the equipment that includes the system is briefly presented.
- Chapter 3 presents the camera as well as some fundamental image processing techniques. These include blurs, edge detectors and morphological operations.
- Chapter 4 is the main part of the project where object detection using computer vision is presented. Firstly, 4 methods used are described each one individually. To elaborate, they are divided according to the technique employed, namely classifier (Method 1) or detector (Method 2). Method 1 section includes a detailed description of how an object is detected including different techniques for image processing. At the end of the chapter an evaluation of 4 approaches employed is presented as well as a discussion for each approach.
- Chapter 5 introduces the main aspects of an ENC and the methodology for sensor fusion and correlation with an ENC.
- Chapter 6 describes the experiments carried out for evaluating the performance of the object detection using computer vision. Results are presented with a discussion for each method.
- Chapter 7 concludes the report. Here, the main findings are presented, as well as possible future work related to what has been presented.

CHAPTER 2
System Description

This chapter presents the ”big picture” where this MSc thesis participates. Project work takes a small part within a larger project carried out by Danmarks Tekniske Universitet (DTU). The scope of this project is part of a larger research undertaking that aims on a fully autonomous ship. As a university, DTU Electrical Engineering (professors, researchers, students...) has been researching on marine control and sensors along with other universities.

- section 2.1 gives an overall description of the framework of the project. This section aims to provide the reader with an overview of how the system is structured and of the field this MSc within the context.
- section 2.2 presents the different kind of sensors employed during the implementation.
- section 2.3 describes the database found which will be used in chapter 4.

2.1 Platform description

This semester, three to four students are involved in the Electronic Outlook part of the project mentioned before. Figure below shows a overall structure of a navigation system with ability to avoid obstacles at sea.

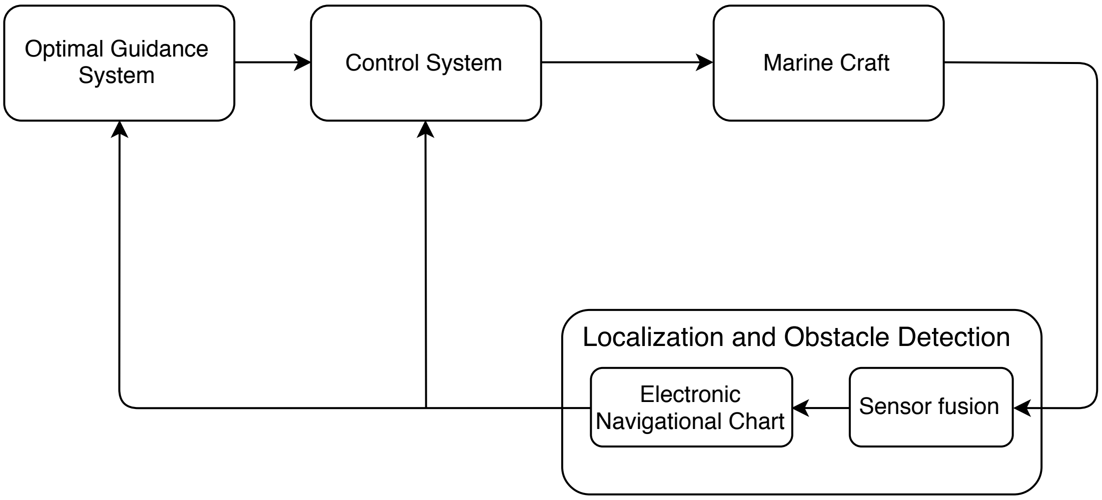

Optimal Guidance System
Control System
Marine Craft
Localization and Obstacle Detection
Electronic Navigational Chart
Sensor fusion

Figure 2.1: Overall structure of a navigation system with ability to avoid obstacles at sea

Firstly, the ship is provided by a path generator that defines a mission through a set of way-points. That mission is followed by the marine craft with the control system which includes a Model Predictive Controller (MPC). This advanced controller outputs an optimal control output that allows the vessel follow the reference in an optimal way.

Situational awareness takes part in the Localization and Obstacle Detection box. This results in information (position and attitude) in the surroundings of the ship. To do so, first data from sensors is processed, fused and finally correlated with an Electronic Navigational Chart (ENC).

2.2 Equipment

Sensors used in this project are presented in this section. Since this thesis takes part in an Electronic Outlook state of the project some of the sensors described below has been subjected to changes. It is important to say, that due to several problems in the acquisition of an Electronic Navigational Chart, this sensor or aid for perception is not included in this section.

2.2.1 IMU and GPS

Information regarding the position and attitude of the ship is performed using an Inertial Measurement Unit (IMU) sensor and a GPS. These sensors are integrated in a single device called MTi-G. The MTi-G is a measurement unit for navigation and control of vehicles and other objects. The internal low-power signal processor runs a real-time Xsens Kalman Filter (XKF) providing inertial enhanced 3D position and velocity estimates.

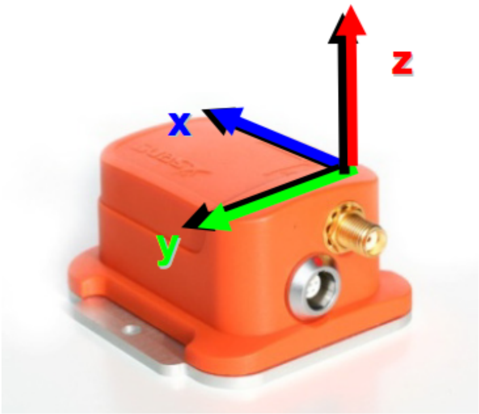


Figure 2.2: MTi-G: integrated GPS and Inertial Measurement Unit (IMU) with a Navigation and Attitude and Heading Reference System (AHRS) processor

2.2.2 Radar

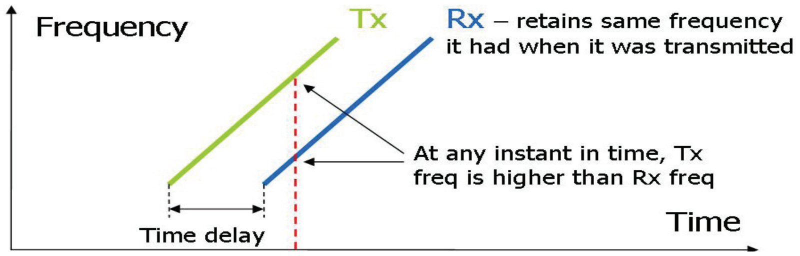
A Radar is used to detect objects in the surrounding of the vessel. The Radar used in this project is a Broadband 3GTM Radar, which uses Frequency Modulated Continous Wave (FMCW). The way that this radar works is briefly described below.

First the radar transmits a 'rising tone' (Tx wave) with linearly increasing frequency. The wave propagates out of the transmitter retaining the frequency it had when it was transmitted. When reflected from an object, the echo has the same frequency it had when it was originally transmitted. The Doppler shift due to relative velocity can be disregarded for sea targets.

The difference between both the currently transmitted and received frequencies, coupled with the known rate of frequency increase, allows a time of flight to be calculated. The radar uses this to calculate distance to target.

2.2.3 Electro-optical sensors

Electro-optical types of sensors used in this project imply cameras operating in the visible and infrared portions of the electromagnetic spectrum. These set of cameras includes two color cameras and one monochrome camera.


Frequency
Tx
Rx – retains same frequency it had when it was transmitted
At any instant in time, Tx freq is higher than Rx freq
Time delay
Time

Source:https://www.westmarine.com/buy/lowrance–broadband-3g-18-radar–12677993

Figure 2.3: How does the radar work?

2.3 Datasets

Works in maritime image processing typically use military owned or propietary datasets which are not made available for research purposes [25]. In order to have a good performance in object detection, several databases were used. These datasets were used for different purposes due to the objects that contain the images. In a maritime scenario, objects that can appear are ships, leisure boats, aids for navigation (buoys), land, buildings on land or harbours. The following datasets were used for training purposes.

- Singapore maritime dataset contains primarily images containing ships.
- Imagenet database include ships and buoys that are used for training detectors and classifiers as it is explained in chapter 4.
- Data collection dataset consists of images taken in some expeditions in Danish waters.

Below, some of these database are presented in detail.

Singapore maritime dataset

Prasad et al. [25] created Singapore Maritime Dataset, using Canon 70D cameras around Singapore waters. All videos were recorded in high definition (1080 × 1920 pixels). Dataset is divided into parts, on-shore videos (visible and near-infra red) and on-board videos, which are acquired by camera placed on-shore on fixed platform. Some details of the dataset are given in table 2.1.

The sample images in figure 2.4 depict the various characteristic in real ocean conditions.

	On-board videos	On-shore videos
Number of videos	4	32
Total number of frames	1196	16254
Number of frames in a video	229	∈[206, 995]
Size of frames (pixels)	1920 ×1080	1920 × 1080

Object detection related	
Number of objects per frame	∈[2, 20]
Number of stationary objects per frame	∈[0, 14]
Number of moving objects per frame	∈[0, 10]
Total number of object annotations in a video	192980
Total number of stationary objects in a video	137485
Total number of moving objects in a video	55495

Table 2.1: Details of the Singapore maritime dataset

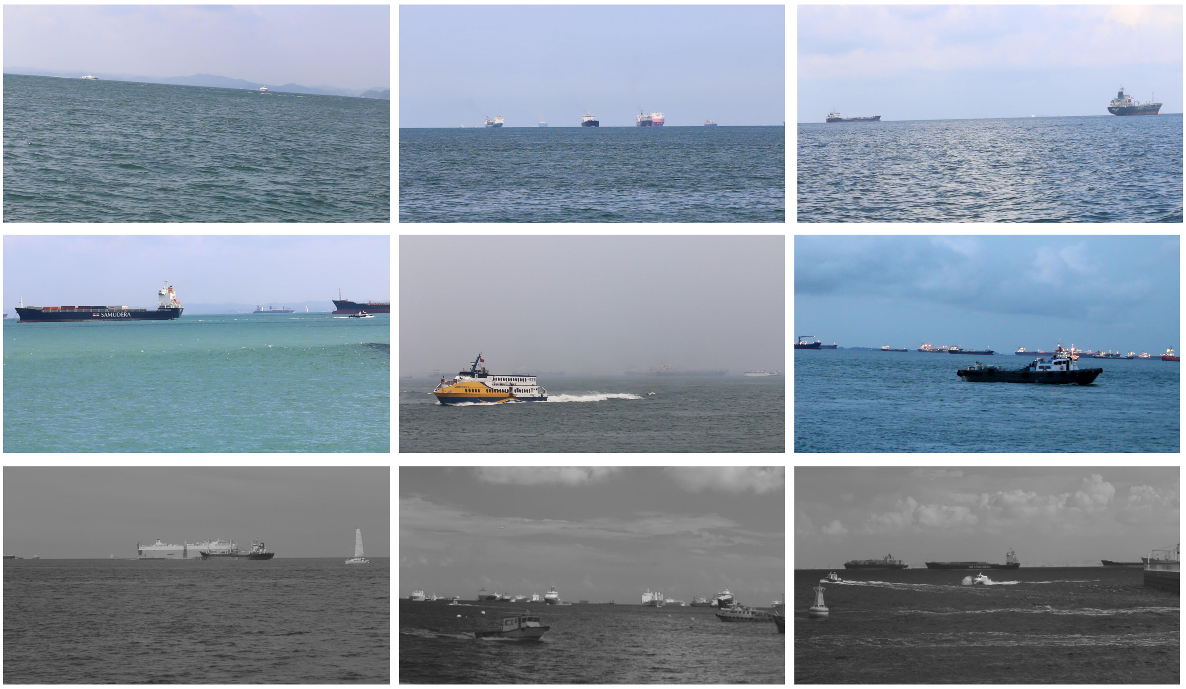


Figure 2.4: Sample from test images (Singapore Dataset)

Imagenet

Imagenet is a large scale hierarchical image database that aims to encompass the largest amount of annotated images for a vast of objects. The hierarchy of imagenet is a tree-structure, where each brand is a sub-category of the parent. For instance, the parent *vessel* is shown to contain children images sets such as *boat*, *ship* and *yacht*. The imagenet sets used in this thesis are shown in table 2.2. Figure 2.5 shows some of the images obtained from Imagenet.

Category	**images**	**Category**	**images**
Buoys	94	Fishing boat	413
Cabin Motorboat	751	Large sailboat	941
Cargo Ship	844	Open motorbooat	220
Cruise Ship	704	Personal watercraft	177
Ferry	659	Small boat	593
Small sailboat	700		
Total number of images	6096		

Table 2.2: Details of dataset from Imagenet

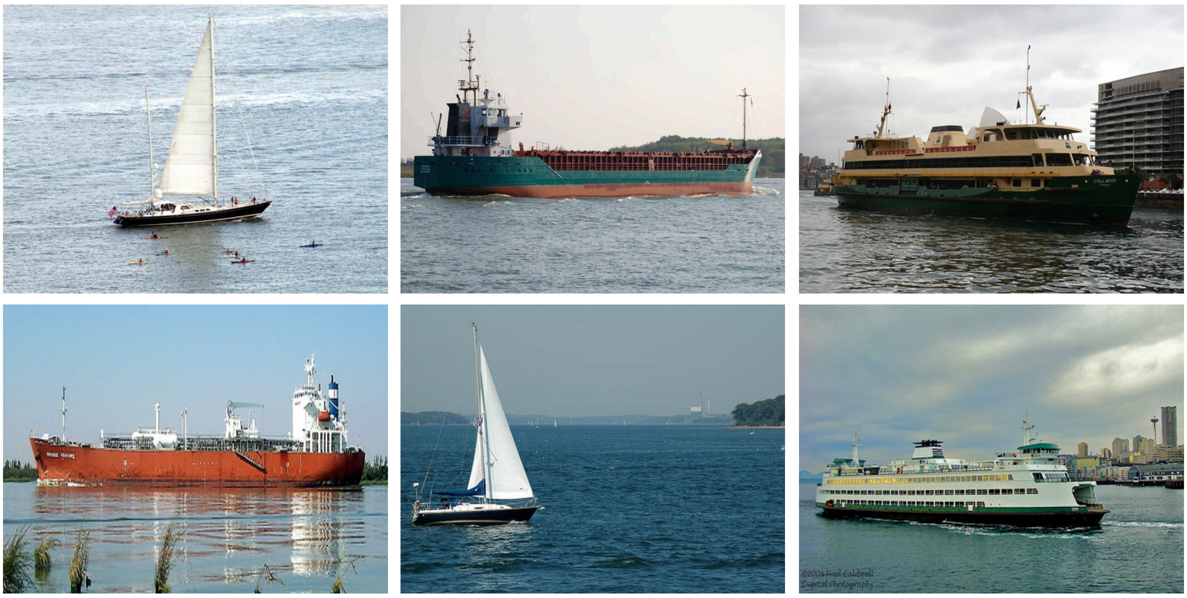


Figure 2.5: Sample from test images (Imagenet)

Data collection

During the thesis some expeditions were conducted to gather data of sea objects such as ships or buoys. The expedition consisted of sailing a ferry on the route Hundested-Rørvig to collected data in different scenarios that a seafarer may encounter. Unfortunately, during the expeditions there were not many merchant ships. But, there were other objects such as buoys and small fishing boats (figure 2.6) that showed up in the journey.

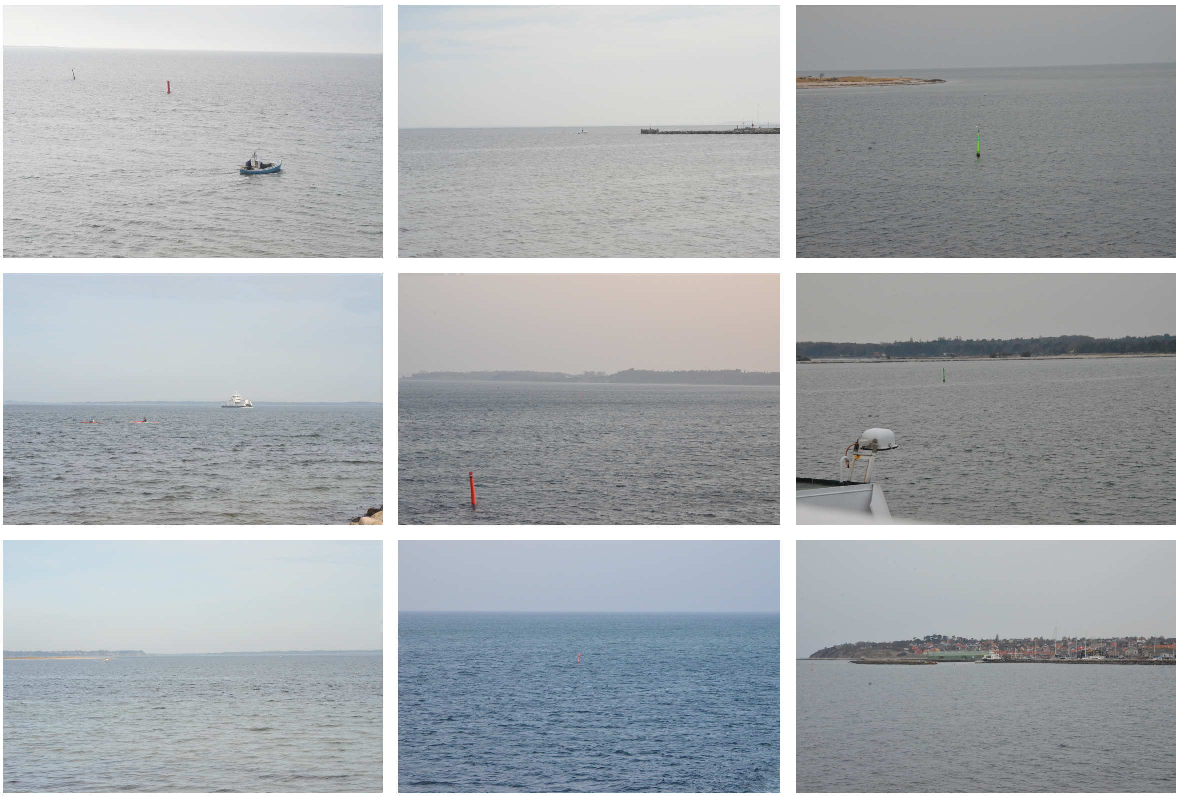


Figure 2.6: Sample from test images(Data collection)

2.4 Summary

To provide situation awareness for sailing an structure of the navigation system as well as the sensors implemented are described. First, the ship estimates the position and attitude through the IMU and GPS. Second, objects around are detected through fusion of radar and electro-optical sensors. In order to have a good performance, electro-optical sensor used algorithms to process images and detect objects. These algorithms are trained using large datasets containing different objects that can appear in a maritime scenario. Finally, some of this objects (static object such as buoys) are correlated with the information provided by the Electronic Navigational Chart.

CHAPTER 3
Image Processing

The analysis of images and their processing are two major fields that are known as computer vision and image processing. A brief introduction to the camera and how it acquires color images is presented in this chapter. It is also described some important image processing techniques that make changes in image properties. This chapter is targeted towards readers with little experience in image processing. The general information in this chapter is obtained in Siegwart, Nourbakhsh, and Scaramuzza's book *Introduction to autonomous mobile robots* [28].

- Section 3.1 introduces some key aspect of camera operation
- Section 3.2 presents image processing techniques such as Gaussian smoothing, edge detectors and morphological operations.

3.1 Image acquisition

A camera in the broadest term could be defined as a photon collection machine. The digital cameras, after starting from one or more light sources, reflecting off of one or more surfaces in the world (subject), and passing through the camera's optics (lenses), light finally reaches the imaging sensor (figure 3.1). The mosaic filter depicted in the figure is a square array of color filters consisting of red, blue and green. When the camera is exposed to light, the photosites of the sensor are excited, converting photons into current. Since the sensor only measure the brightness of the light, no its colour, colour information is gathered by red, green and blue filters. These filters are arranged into 2×2 sets of four in a mosaic known as Bayer pattern. Normally, two pixels of 2 × 2 block measure green while the remaining two pixels measure red and blue light intensity. The reason why there are twice as many green filters as red and blue its that the luminance signal is mostly determined by green values.

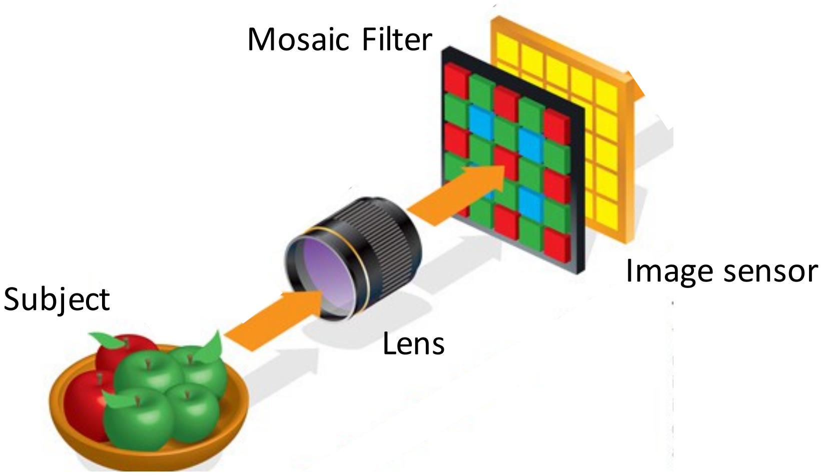


Image courtesy: www.digitalcameraworld.com

Figure 3.1: Digital camera diagram

The next step for acquiring and image is demosaicing. Demosaicing is the process of translating this Bayer array of primary colors into a final image which contains full color information at each pixel. Instead of thinking of the 2×2 array of red, green and blue, we can consider a single full cavity. (figure 3.2). This would work fine, however most cameras take additional steps to extract even more image information from this color array.

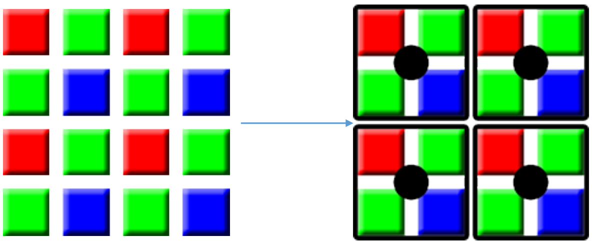


Figure 3.2: Bayer demosaicing

3.2 Image processing

Image processing can be treated as signal processing where the input is an image (such as a photo or a video) and the output is either an image or a set of parameters associated with the image. Images are widely used in image processing techniques as a two dimensional signal $I(x, y)$ where x and y are the pixel coordinates (spatial image) of the input image and I is the amplitude of the image for any given x and y coordinates.

3.2.1 Smoothing filters

Image filtering is one of the most used tools in image processing. The reason why are they called filters is due to the fact that in frequency domain processing, the world ”filtering” refers to the process of accepting or rejecting certain frequency components. In this section smoothing filters are presented but first an introduction to spatial filtering is described.

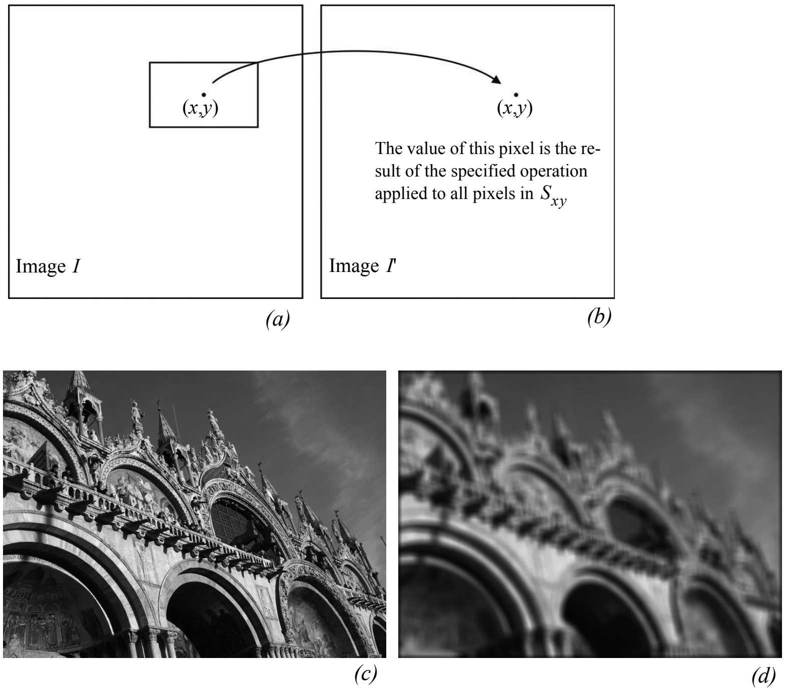
Image filters can be implemented both in the frequency domain and in the spacial domain. In the latter case, the filter is called mask or kernel. A kernel consists of (1) a region around the pixel examination (typically a small rectangle), and (2) a predefined operation T that is performed on the image pixels encompassed by the neighborhood of the pixel. Let defined $S_{x,y}$ the set of coordinated of the neighborhood pixels around an arbitrary point (x,y) in an image I. Image filtering will generate a new output image I' where the value of each pixel is determined by the specified operation. This operation can be expressed as follows:

$$I'(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) \cdot I(x - s, y - t), \tag{3.1}$$

where w is the filter of size $m \times n$ (with $m = 2a + 1$ and $n = 2b + 1$) which are usually assumed odd integers. The expression presented is call a convolution an it can be written in a more compact way as:

$$I'(x, y) = w(x, y) * I(x, y), \tag{3.2}$$

where $*$ denotes the convolution operator.


(x,y)
(x,y)
The value of this pixel is the result of the specified operation applied to all pixels in S_{xy}
Image I
Image I'
(a)
(b)
(c)
(d)

Source: *Introduction to autonomous mobile robots* [28]

Figure 3.3: Illustration of the concept of spatial filtering. (c) Input image. (d) Output image after application of average filter

Smoothing filters are special filters used for blurring and noise reduction. One of the filters used for smoothing is the 2-D Gaussian filter. The continuous 2-D Gaussian is given by:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\dfrac{x^2 + y^2}{2\sigma^2}}. \tag{3.3}$$

This 2-D Gaussian can be discretize by sampling about its center. For example, for generating a 5 × 5 mask with $\sigma = 1$ we obtain:

$$G_d = \frac{1}{289} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 28 & 16 & 4 \\ 7 & 28 & 49 & 28 & 7 \\ 4 & 16 & 28 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{3.4}$$

Main parameters that can be adjusted in a Gaussian filter are the size of the kernel or mask and σ value. The kernel size is usually set to 5×5 kernel for fast implementation however the value for σ is directly related to the blurring of the image. In this project, this technique is used for removing noise produced by wakes or foam in the images.

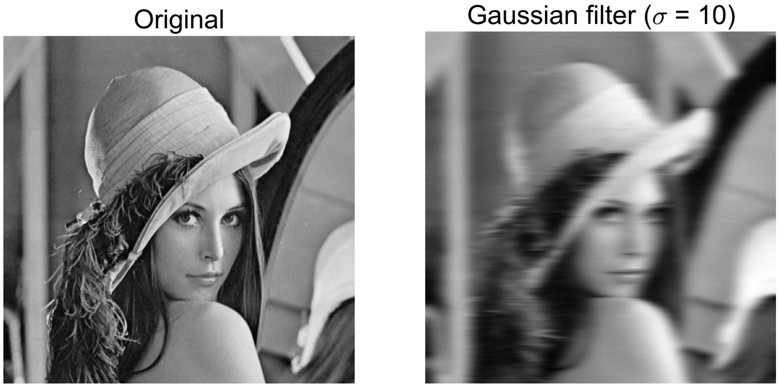


Figure 3.4: Illustration of Gaussian filtering

3.2.2 Edge Detection

Edges define regions in the image plane where a significant change in the image brightness takes place. As shown in figure 3.5, salience of the edges reduces the amount of information from the image and it may result in an useful feature during image interpretation. Therefore a practical edge detector should simply differentiate, since an edge by definition is located where there are large transitions in intensity.

One of the most competent edge detector is the Canny edge detector [28]. This edges extractor smooths the image I via Gaussian convolution and then looks for maxima in the derivative. In practice, the smoothing and differentiation are combined into one operation because of this property of convolutions:

$$(G * I)' = G' * I. \tag{3.5}$$

Since edges might appear in any directions, it is required to apply two perpendicular filters:

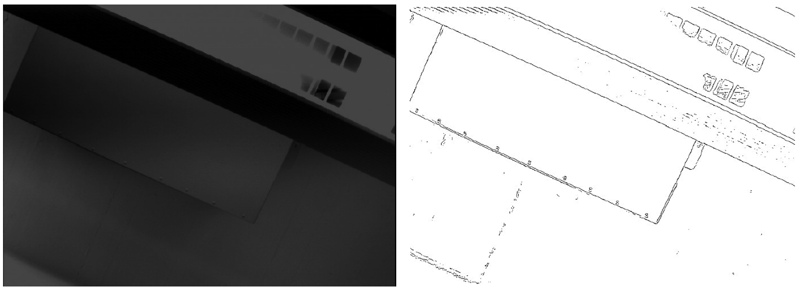
$$f_V(x,y) = G'_\sigma(x)G'_\sigma(x)f_h(x,y) = G'_\sigma(y)G'_\sigma(x) \tag{3.6}$$

where G'_σ is the first derivative of G. This results in a basic algorithms for detecting edges at arbitrary orientations:

- Convolve the image $I(x,y)$ with $f_V(x,y)$ and $f_H(x,y)$ to obtain the gradient components $R_V(x,y)$ ans $R_H(x,y)$, respectively.

- Define the square of the gradient magnitude $R(x,y) = R_V(x,y)^2 + R_H(x,y)^2$

- Mark those peaks in $R(x,y)$ that are above some predefined T This threshold is selected manually so weakest edges are eliminated.



(a) (b)

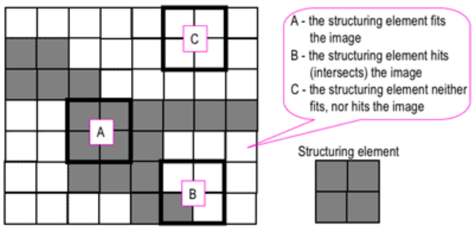
Source: *Introduction to autonomous mobile robots* [28]

Figure 3.5: Edge detection

3.2.3 Morphological operations

In the following chapter, binary images are mentioned. In this case, the binary images are generated by a threshold, which leads in numerous imperfections due to noise o texture. The aim of morphological image processing is removing these imperfections trough some operations that are described below. Morphological image processing is a collection of non-linear operations related to the shaper or morphology of features in an image. Since morphological operations rely on the relative ordering of pixels, not on the numerical values, they are especially suited to the processing of binary images.

Morphological techniques use a small shape or template called a structuring element. Structuring elements play in morphological image processing the same role as convolution kernels in linear image filtering. This structuring element is positioned in all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Operations consist of testing whether the element fits within the neighbourhood or if it hits or intersects the neighbourhood (3.6). The impact of the structuring element shape is related with the element we want to modify in the image.

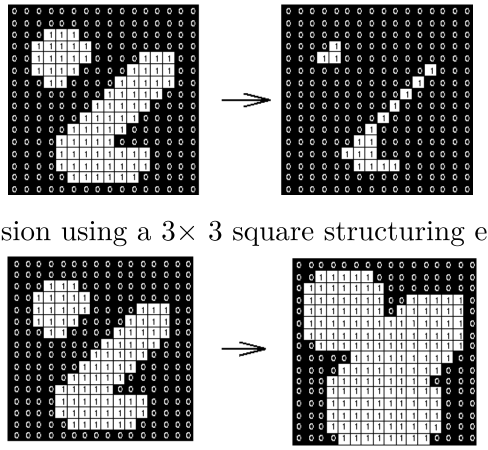

C
A
B
A - the structuring element fits the image
B - the structuring element hits (intersects) the image
C - the structuring element neither fits, nor hits the image
Structuring element

Source: https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm

Figure 3.6: Testing on an image with a structuring element

Erosion and dilation

Some of the basic operations used in morphological image processing are erosion and dilation. Both operations produce a new binary image that is generated by a certain structuring element. Theses operations are commonly used consecutively in order to remove small objects and to joint objects from the image that are close. To illustrate figure 3.7 shows an example of the two morphological operations. First the image is eroded which means that the structuring element shrinks the image by stripping away a layer of pixels from both the inner and outer boundaries of regions. The holes and gaps between different regions become larger, and small details are eliminated. Then holes enclosed by a single region and gaps between different regions become smaller, and small intrusions into boundaries of a region are filled in.



(a) Erosion using a 3× 3 square structuring element

(b) Dilation using a 3× 3 square structuring element

Source: www.cs.princeton.edu/ pshilane/class/mosaic/

Figure 3.7: Morphological operations: erosion and dilation

CHAPTER 4 Object Detection using Computer Vision

As it was mentioned above, an autonomous vessel will require situation awareness by sensing the immediate environment to avoid collision with other ships or with other traffic areas. One way to enhance situation awareness is by using Electro-optical (EO) sensors. In the latest years camera-systems have been employed to aid humans and robots alike in detecting and classifying objects in the real world. EO sensors are prepared to complement ranging devices and they are of interest for two main reasons. Firstly, the image generated by them are directly interpretable and intuitive for human operators. Secondly, the image stream from them can be used to image processing and computer vision such that advanced intelligence can be generated without significant human intervention.

In this chapter, several approaches are introduced to detect objects using computer vision. Further, and evaluation of each of these methods is presented.

- Section 4.1 contains a brief summary of the approaches implemented in this project. The section aims to provide the reader with insight into the two main methods used for object detection.
- Section 4.2 describes the two approaches using Method 1.
- Section 4.3 describes the two approaches using Method 2.
- Section 4.4 presents the results for the different approaches.

4.1 Methodology

Four different methods for object detection have been applied in this project both for color and infrared cameras. This section briefly presents them whereas theoretical principles and techniques used on them will be treated in the following sections. The aim is to compare the performance and choose the most suitable detector.

Two of them use conventional approaches whereas the two others use more advanced techniques that could be considered as the state-of-the-art in image analysis (involving training of Convolutional Neural Networks (CNN)). List of acronyms is presented in table 4.1.

Detection approaches are arranged according to the method and technique employed. Methods are divided into Region of Interest (RoI) with classifier and pure detector. Different techniques adopted in the project discern between whether the method employs either neural networks or classical techniques. Table 4.1 shows the arrangement used.

		Technique	
		Conventional	**CNN**
Method	**Method 1**	ROI algorithm + SVM	ROI algorithm + CNN classifier
	Method 2	ACF detector	Faster RCNN detector

Table 4.1: Methods for object detection

RoI	region of interest	Areas of an image whose color and/or texture differs from that of their background.
SVM	support vector machine	Supervised learning model used for classification of data.
ACF	aggregate channel features	Supervised learning model that extracts features from different channels of images for detection purposes.
CNN	convolutional neural networks	Specific type of neural networks (trained through supervised learning) used for object classification.
RCNN	region-based convolutional network	CNN applied to object detection.

Table 4.2: List of acronyms from table 4.1

Method 1 consist of a multi-scale process to detect a particular RoI from an image and a classifier to allocate the object detected. These RoIs contain potential to be categorized. First step is to find these RoIs that are defined as bounding boxes. How these RoIs are obtained is explained in the next section. Second step consists of classifying each of the bounding boxes by using two different types of classifiers: Support Vector Machine

(SVM) and Convolutional Neural Network (CNN) classifiers, that are covered below. The part from classifiers has been implemented commonly by a fellow student, Eduardo Vázquez, whose MSc thesis is tightly connected to the one presented in this report.

Method 2 does not use a multi-scale process to detect the objects. The main difference between both methods is that the two steps in Method 1 (detect + classify), in the case of the Method 2 it take place internally. Two different techniques have been employed: Aggregate Channel Features (ACF) detector which uses aggregate channel features and the state-of-the-art Faster Region-based Convolutional Neural Network (RCNN) detector. This part was performed by Eduardo Vázquez. Main principles will be cover below.

4.2 Object detection using Method 1

This section presents Method 1 for object detection. Figure 4.1 shows a description of the implementation to detect and classify an object. First, RoIs are detected in the image based. Since any found RoI could differ from others a classifier is needed. Therefore, a classifier is trained previously using several images containing objects to be classify. Second step is classification where based on the trained classifier RoIs are arranged into different categories.

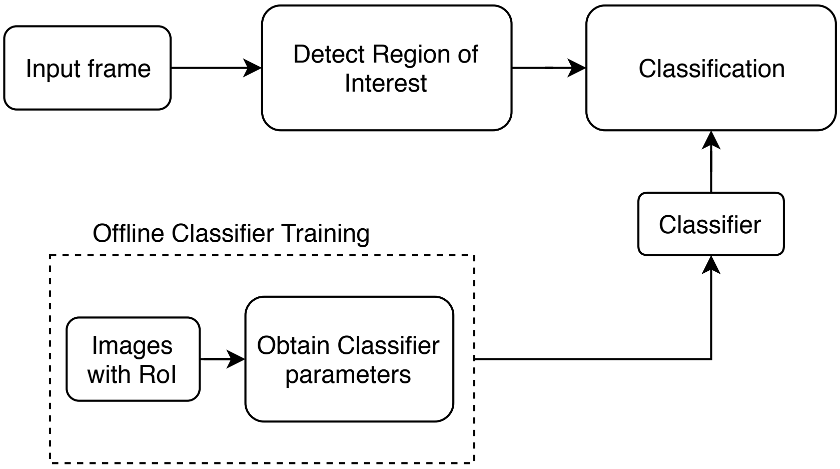

Input frame
Detect Region of Interest
Classification
Classifier
Offline Classifier Training
Images with RoI
Obtain Classifier parameters

Figure 4.1: General Framework of object detection using Method 1

Firstly, an overall description of how the RoI is found is described. Secondly, the two different classifiers used are presented.

4.2.1 Detect Region of Interest

The algorithm presented is based on the general framework for object detection used in conventional forms. Prasad et al. [25] work contains some of the historical milestones for object detection in the maritime environment. This paper provides a comprehensive overview of various approaches of video processing for object detection.

For object detection in maritime EO data processing, each frame of the EO video stream is considered independently without taking temporal information into account. The general pipeline is presented in figure 4.2. It consists of three main steps, namely, horizon detection, background subtraction and foreground segmentation.

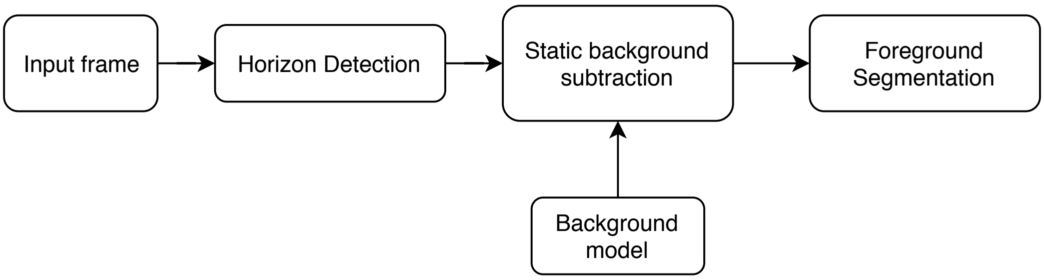


Figure 4.2: General pipeline of maritime EO data processing for object detection

4.2.2 Horizon detection

In the first step of the method the location and orientation of the horizon line are detected. Horizon detection is useful in maritime electro optical data for various purposes being registration for mobile sensors such as buoys, maritime vessels, unmanned aerial vehicles [24]. Further, the horizon line can be used as a reference to limit the regions of interest and reduce the execution time of detection.

In literature, basically three main approaches for horizon detection can be distinguished: projection based, region based and hybrid approach. In this project, projection based has been selected due to its simplicity and computational speed. Projection methods computes the edge map of the image by using edge detectors. Then, it is projected to another space where prominent line features in the edge map can be identified easily [15].

In maritime scenario, horizon often appear as a straight line and therefore is expected to be simple. Consequently, some of the assumptions of the horizon extraction algorithm include that the sky is visible and there is a horizon in the images. The general notation and derivation is inspired by Bao, Xiong, and Zhou work “Vision-based horizon extraction for micro air vehicle flight control” [2].

The given algorithm is based on the orientation projection method which includes, image preprocessing, projection of scope determination, pixel projection calculation, projection direction determination with maximum projection value and horizon estimation. The algorithm is extracted from Bao, Xiong, and Zhou work and the following is an adaptation with some enhances of the algorithm.

The main changes made in the algorithm are:

- the use of the Canny edge detector,
- incorporation of an optimal threshold threshold selector,
- resize of the image to increase computational speed,
- increase the orientation accuracy by selecting larger projection directions,
- algorithm to estimate the goodness of the horizon line estimation.

Image preprocessing

First step is image preprocessing, which involves noise removing, resize of the image, image binaryzation and edge detection. By adjusting the size of the image, computational time is reduced in the case the image's size is considerable. Further, Gaussian filter is used to remove high noise produced by the different shapes of the sea such as waves, foam, etc.

Since it is taken the existence of a horizon, there is also a local intensity difference between the sea surface below the horizon and the air (or coast) above the horizon (or coast line). This can be used to select a threshold to binarize based on the statistical properties of the image. Brief description of the threshold selection is explained below.

Input frame from Figure 4.2 is assumed to be sampled to form a $L\ x\ M$ (320 x 240) discrete gray scale image with integer density values from the range [1,m]. Suppose that the total number of the pixels with gray level is n_i. Then the total number of the pixels in the image is given by

$$N = \sum_{i=1}^{m} n_i \tag{4.1}$$

The probability of each gray level is

$$P_i = \frac{n_i}{N} \tag{4.2}$$

Separate the gray levels into two groups C_0 = {1,2,..,k} and C_1 = {k+1,k+2,...,m} by the integer k. Then the probabilities and means of C_0 and C_1 are given by

$$\begin{cases} \omega_0 = \sum_{i=1}^{K} P_i \\ \mu_0 = \sum_{i=1}^{k} iP_i/\omega_0(k) = \mu(k)/\omega_0(k) \end{cases} \tag{4.3}$$

$$\begin{cases} \omega_1 = \sum_{i=k+1}^{m} P_i = 1 - \omega_0(k) \\ \mu_1 = \sum_{i=k+1}^{m} iP_i/\omega_1(k) = [\mu - \mu(k)]/[1 - \omega_0(k)] \end{cases} \tag{4.4}$$

where $\omega = \sum_{i=1}^{m} iP_i$ is the statistical mean of the whole image gray levels. According to 4.3 and 4.5,

$$\mu = \omega_0(k)\mu_0(k) + \omega_1(k)\mu_1(k). \tag{4.5}$$

The variance between the two groups is

$$\sigma^2 = \omega_0(k)[\mu_0(k) - \mu]^2 + \omega_1(k)[\mu_1(k) - \mu]^2. \tag{4.6}$$

The optimal threshold k is the gray level by which $\sigma^2(k)$ is maximum. Once the optimal threshold is found binaryzation is implemented by

$$G(i,j) = \begin{cases} 255, & G(i,j) > k \\ 0, & G(i,j) \leq k \end{cases}, i = 1, ..., L, j = 1, ..., M \tag{4.7}$$

Where $G(i,j)$ denotes an observed density value of the pixel (i,j). After applying optimal threshold edge detection is implemented.

Several methods can be used in the edge detector such as LoG (Laplacian of Gaussian) operator, Prewitt, Robert, Sobel, Canny etc. The canny operator is adopted in this project since it is less likely than the other methods to be fooled by noise. After being processed by the Canny detector a binary image is obtained, in which the line information is prominent.

Figures 4.3 and 4.4 shows an example of the process described above. First, the RGB image is converted to a greyscale (figure 4.3). Then it is resized and high noises removed (figure 4.4a). Figure 4.4a is shown in a greyscale. Then an optimal threshold that separate the grey levels in two groups is applied (figure 4.4b) and the image is binarized. Lastly, edges in the binary image are detected.

Orientation Projection Algorithm

The orientation projection algorithm takes place after image preprocessing. First, 162 projection directions are defined, which are denoted by the orientation number $m (m = 0, 1..., 161)$. This value was selected in order to cope with all the possible orientations and have a good performance. In order to balance the computational precision with the speed, projection directions are chosen from $0\,^\circ$ to $45\,^\circ$ and from $135\,^\circ$ to $180\,^\circ$ ($0\,^\circ$ and $180\,^\circ$ are considered as the same direction). One may notice, that directions larger than $45\,^\circ$ and smaller than $135\,^\circ$ are not taken into account since horizon line rarely appear in those directions in a maritime scenario. Therefore the orientation projection can be made for an image in the selected direction.

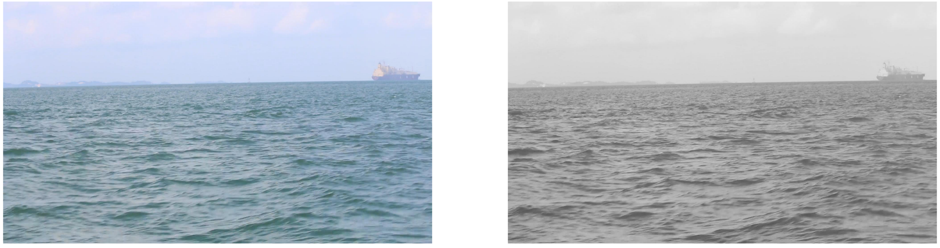
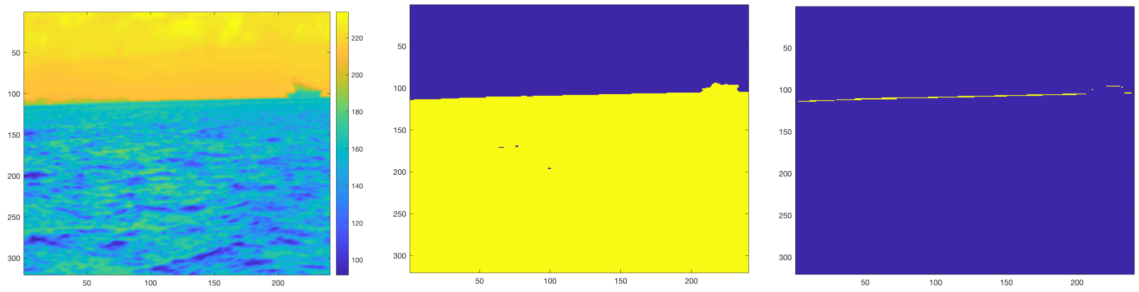


Figure 4.3: Conversion from color image (left) to greyscale image (right).



(a) Noise removing and resize

(b) Global threshold selection

(c) Egde detection

Figure 4.4: Preprocessing procedure in an image for horizon detection

Since the image is binary, white pixels (defining edges) are used to obtain the projection value, which is the number of white pixels along the selected projection direction in the image. The maximum projection value in each direction and the position corresponding the maximum value are recorded. The horizon line is then drawn based on the projection direction and the position where the global maximum projection value is obtained.

Figure 4.5 shows the sketch map orientation projection. Two coordinate systems are shown, the image coordinate system defined by OX and OY and the projection axis coordinate defined by OP. Where O is the common origin of the two coordinate systems, Φ is the orientation angle dependent on the orientation number m, representing the projection direction. OP is the projection axis, which is perpendicular to the selected orientation angle Φ. A and B are the projection limits. L_1L_2 is a line parallel to the selected projection direction, which is vertical to OP. (x, y) is the coordinates of a pixel on L_1L_2. n is the position at which the line L_1L_2 is projected onto OP.

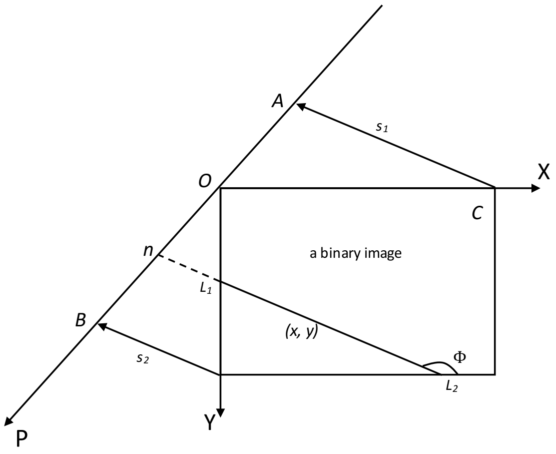


Figure 4.5: Sketch map of orientation projection for an image

Once main parameters are defined, all the pixels of the images are projected to the projection axis along the selected projection direction within the limited scope. The relationship between the projection coordinate n and the coordinates (x, y) of a projected pixel located on L_1L_2 satisfies the following equation

$$n = -x\sin(\Phi) - y\cos(\Phi), \quad 0 \leq \Phi \leq \tfrac{\pi}{4} \quad and \quad \tfrac{3\pi}{4} \leq \Phi \leq \pi \tag{4.8}$$

$P(x, y)$ is defined as the pixel projection values of the pixel (x, y),

$$P(x, y) = \begin{cases} 1, & G(x, y) = 255 \\ 0, & G(x, y) = 0 \leq k \end{cases} \tag{4.9}$$

and $P_m(n)$ represents the projection value of the image pixels along the line L_1L_2

$$P_m(n) = \sum_{(x,y)\in L_1L_2} P(x, y). \tag{4.10}$$

Figure 4.6 shows a pixel projection curve of an image in the selected direction m. The

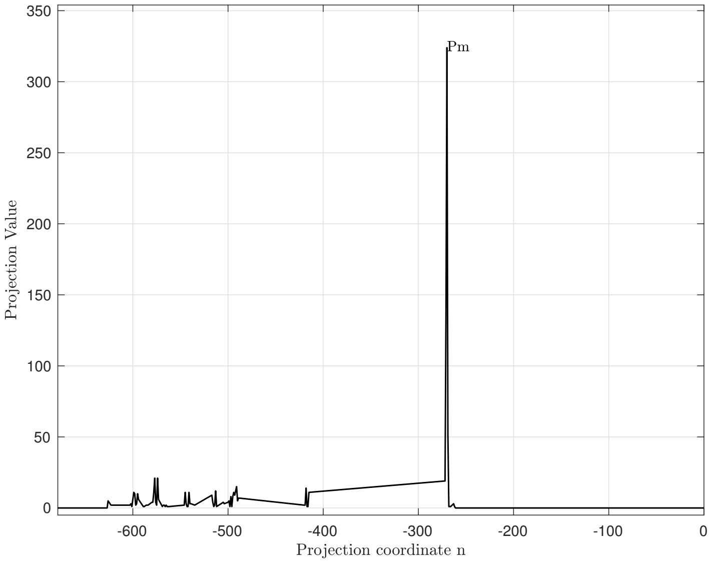


Figure 4.6: Pixel projection curve of an image in the projection direction m

peak of the projection value P_m occurs at the position n_0. This peak means that the longest edge detected in the selected direction Φ passed through n_0. Those values, m, P_m, n_0 are recorded so they can be used for further calculations. One example of this can be seen in Figure 4.7, where an original image is preprocessed and the pixel projection computed in one direction. The left image is an original image captured from a on-board camera. The right one is the preprocessed image with the sketch map of the pixel projection curve in one direction. The line OD represent the projection direction. The curve above the projection axis is the orientation projection curve.

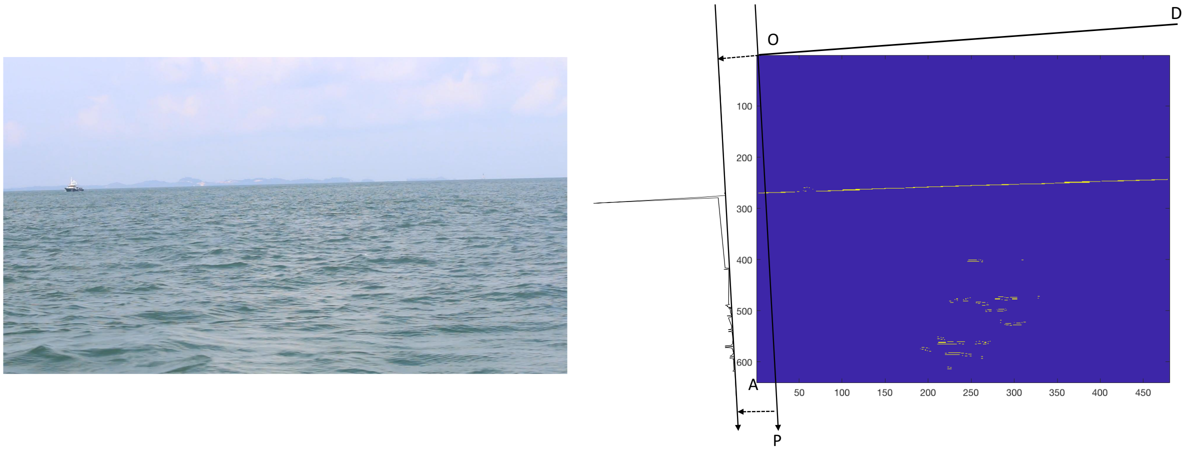


Figure 4.7: Original image and the pixel projection result in one direction

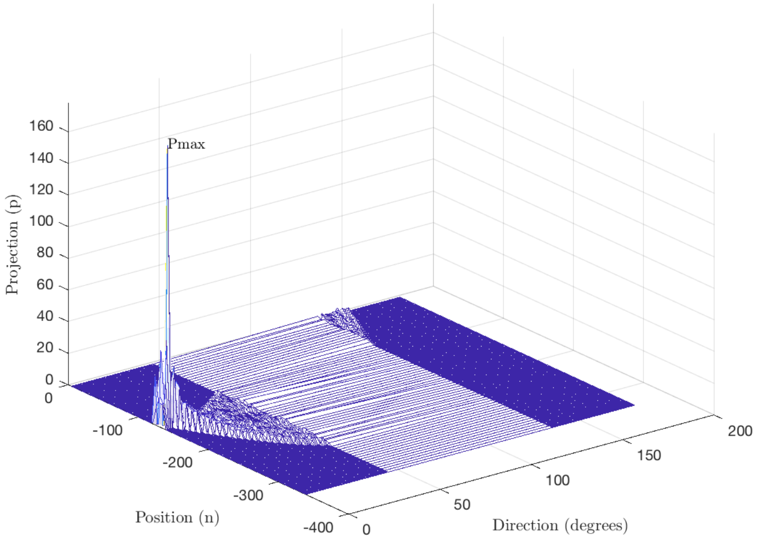
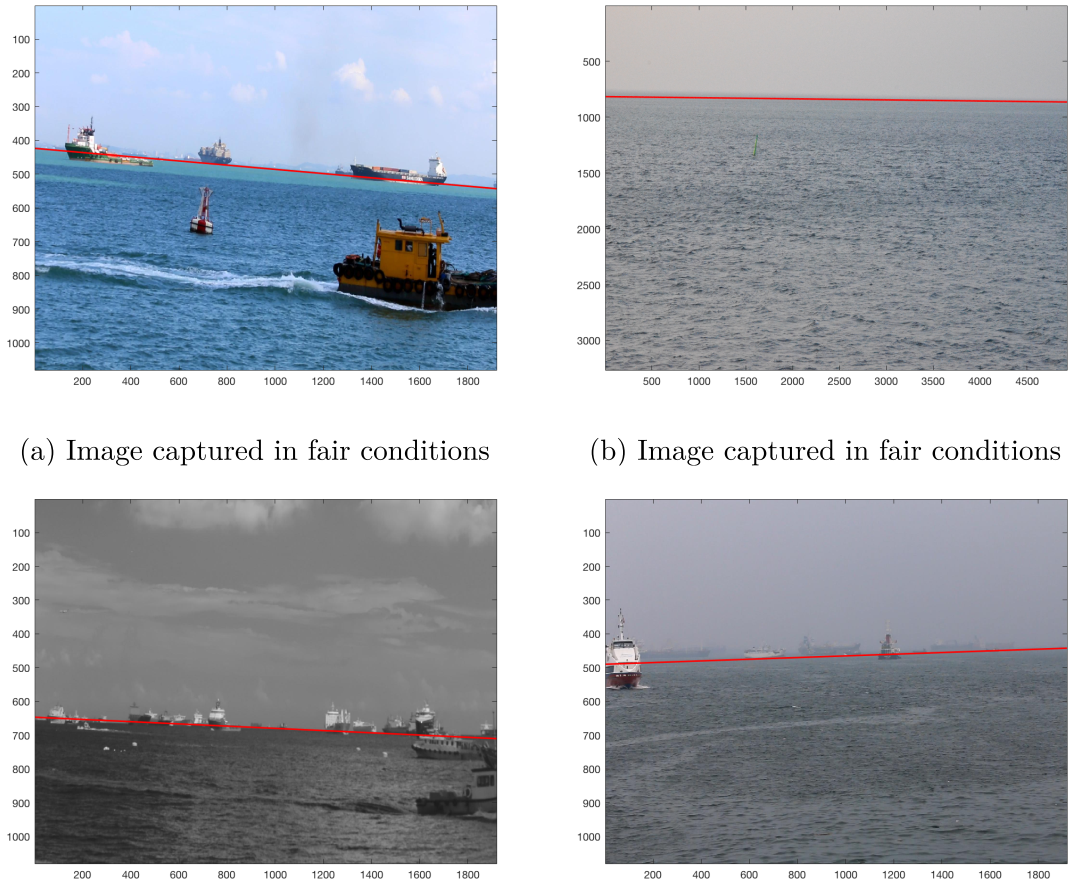


Figure 4.8: Three-dimensional graph of orientation projection

In order to analyze all possible directions, the image has to be projected in all directions then a three dimensional plot is obtained as seen in figure 4.8. Values recorded previously (orientation number, m, maximum projection value P_m and projection position n_0) are used to find where the global maximum projection value is. Once the P_{max} is obtained is straightforward to obtain the horizon line using the projection direction m and the position n corresponding to the peak P_{max} as it is shown in the following equation

$$\begin{aligned} &\Phi = \begin{cases} m \times \frac{\pi}{4\times 80} & m \in (0,80) \\ m \times \frac{\pi}{161} & m \in (81,161) \end{cases} \\ &y = -x\tan(\Phi) - \frac{n}{cos(\Phi)} \qquad \Phi \in [0,\frac{\pi}{4}] \cup [\frac{3\pi}{4},\pi] \end{aligned} \tag{4.11}$$

Figure 4.9 gives some processing results for different weather conditions. It can be seen that the horizon can be extracted not only from fine images captured in fair conditions but also from blurred images captured in cloudy or even foggy days. Further, the algorithm is effective for both color images (Fig. 4.9a and Fig. 4.9b) and gray images taken from a Near Infra-Red (NIR) camera (Fig. 4.9c).



(a) Image captured in fair conditions

(b) Image captured in fair conditions

(c) Image captured from a NIR camera

(d) Image captured in a foggy day

Figure 4.9: Horizon extracted (red line) from video images

Evaluation horizon line

In order to check the goodness of the horizon estimation it has been developed an evaluator that measures the whether the horizon has been detected correctly. The evaluation method is needed in order to continue the process of object detection correctly. For example, if the horizon is not well detected, the posterior process, namely, background subtraction and foreground segmentation will lead in wrong results.

The evaluation method takes into account the 'strength' of the lines that belongs to the estimated horizon line. In 4.7 (left) it can be seen a prominent line that goes from one side (left) to the other one (right) in the image that corresponds to the horizon line. This line is produce by a bunch of small lines. The evaluation method observes some aspects of the lines that conform the final line. These values give us an idea of the goodness of the estimation.

Therefore, when the horizon is detected, the number of lines and the total length are considered and through some thresholds that were fixed for some test images it will give us whether the horizon is matched or not.

4.2.3 Background Subtraction

Once the horizon is detected next step is background subtraction. There is a large collection of works related to background subtraction. Background subtraction can be considered under two scenarios in a maritime environment: open seas and close to the port/harbour. Both scenarios pose a challenge due to the dynamic of the water background such as waves, wakes, foams and debris in the case of open sea and buildings or stationary vessels in the case of close to port/harbour. The current literature in maritime background subtraction almost exclusive deals in the first case.

Background subtraction technique is a tool for object detection, where a pixel-wise statistical background model is used to classify the input video stream into foreground and background regions. Detection performance relies on the modelled background. For this purpose, 3 different methods for background modeling are presented in this section.

Method A is the most straightforward of the three methods. It consists of a background reference of the sea in the Hue Saturation Value (HSV) color space that is selected manually.

Method B uses Gaussian Mixture Model to represent background based on the luminance of the pixel. The histogram of both visible and infrared images are invariably multimodal in some cases because of the presence of foam, debris or wakes. Therefore Gaussian Mixture Model (GMM), which are suitable for representing multimodal backgrounds [3, 35, 11, 13, 16] can be used for background modeling using pixel luminance values.

Method C is similar to Method B but it presents a novel method that uses Discrete Cosine Transform (DCT) based feature selection of background texture. It can be seen in a large range of maritime videos that the texture feature of sea surface is generally uniform and consistent. Considering this texture consistency of the sea-surface background this method employs GMM for modeling different textures from background.

As it will see later these 3 methods provides different benefits depending on the desired object to be detected. Method A is suitable for detecting small object such as buoys whereas Method B and C have a better performance detecting larger objects such as ships.

Table 4.3 shows an overview of the methods employed in this project.

	Learning Method	Feature	Application	Sensor
Method A	No learning	HSV histogram	Buoys	Visible range
Method B	GMM	Grayscale	Ships	Visible/Infrared range
Method C	GMM	Texture	Ships	Visible/Infrared range

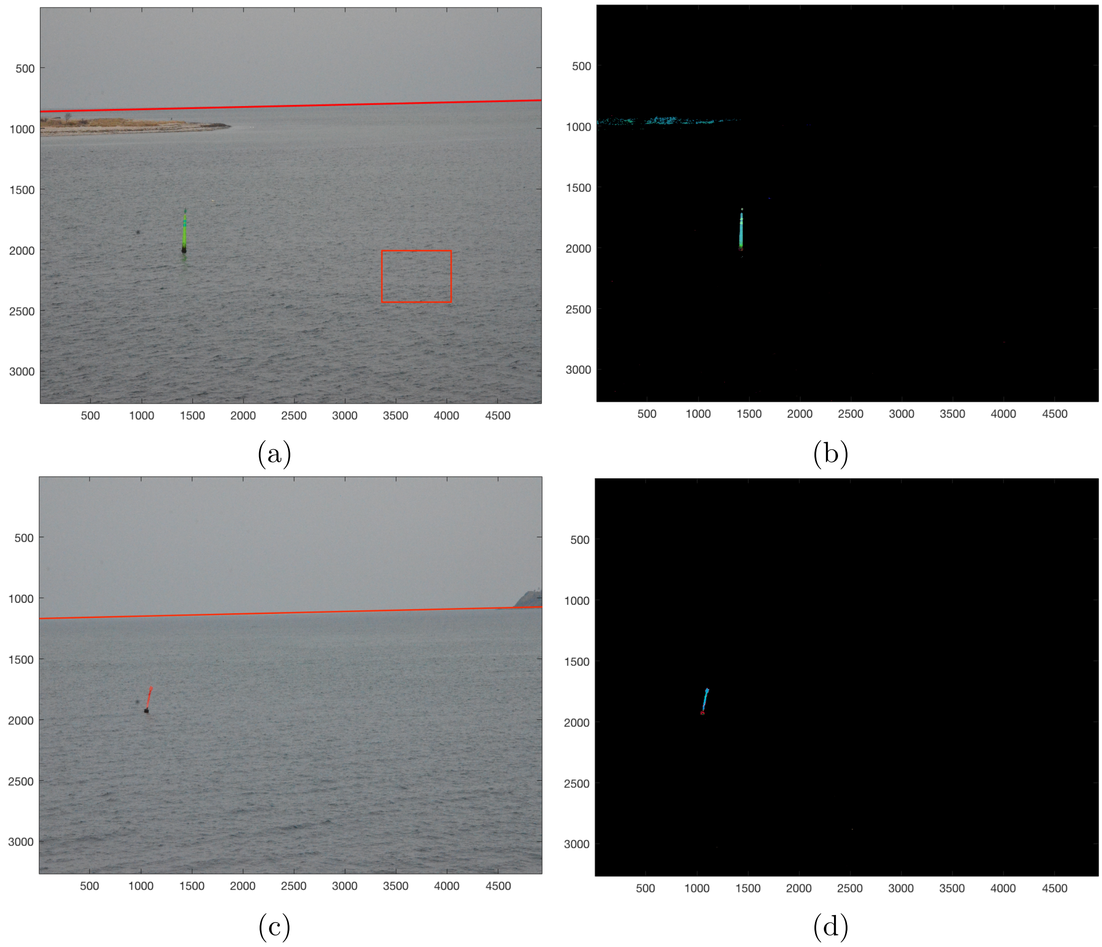
Table 4.3: Background modelling techniques description

This methods are explained in the following separately.

4.2.3.1 Method A

Method A uses a reference mask defined previously. This method relies on light conditions where colours are distinguishable. The reason why this method is simple is because no learning is involved. First step consists of selecting a region of sea background from the input image. Next this region is converted from Red Green Blue (RGB) to Hue Color Value (HSV) color space which has a better performance. This was tested with several images and it was found that HSV describes more accurately a region rather than RGB color space. Now a sea mask is created and it is ready to be used to remove sea from the image. This method only considers objects below horizon line like traffic maritime signals (buoys). Therefore once the horizon line is detected, region above the horizon is cropped in order to reduce processing time and reduce searching boundaries.

Figure 4.10 shows an implementation of this method. From 4.10a horizon is detected and reference selected manually. This mask is then applied to remove pixels within the defined mask. To illustrate this point, figures 4.10b and 4.10d show a background removal using this method where sky has been cropped. Some advantages of this method are that it is able to detect small objects and the background model is accurate enough since it is selected manually by the user. However the aim of an autonomous ship, relies on the fact that the user has the minimum interaction with the system. Further, this method only detects objects below the horizon line. It is was initially though to detect only buoys and due to the lack of time and and the better performance of other methods Method A is only used for that purpose. Thus, as it was mentioned before, region above



(a)

(b)

(c)

(d)

Figure 4.10: Image subtraction using Method A

horizon is not considered since buoys will only appear in the sea. It is true that some of them can be showed up closed to the horizon and may be cropped but those buoys anyway far away and are very difficult to detect due to a poor resolution. Finally, it is important to mention that Method A is only suitable for cameras operating in the visible spectrum in daylight.

4.2.3.2 Method B

In order to give a better insight of this method, GMM is presented. Given the intensity I_t of a pixel, the probability $P(I_t)$ of that pixel belongs to the background is given as Wang et al. [34]:

$$P(I_t) = \sum_{i=1}^{K} \omega_{i,t}\eta(I_t, \mu_{i,t}, \sigma_i^2), \qquad i = 1, 2, ..., K \tag{4.12}$$

where K is the total number of Gaussian distributions, $\omega_{i,t}$ is the normalized weight of the i^{th} Gaussian distribution at time t, $\mu_{i,t}$ and $\sigma^2_{i,t}$ are the mean and variance of the i^{ith} Gaussian in the mixture at time t, respectively. The Gaussian probability density function η is defined as

$$\eta(I_t, \mu_{i,t}, \sigma_i^2) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \sigma_i^2 \mid^{\frac{1}{2}}} e^{\frac{1}{2}(I_t-\mu_i)^T(\sigma_i^2)^{-1}(I_t-\mu_i)} \tag{4.13}$$

The procedure of pixel intensity-based background modeling is to simulate a mixture of Gaussian distributions using pixel intensity values. Using GMM aims to estimate unknown parameters of ω_i, μ_i, Σ_i in the Gaussian probability density functions.

Learning of Gaussian mixture model

The learning mechanism of the proposed background modeling in [38] is described as follows:

(1) Initialize the following GMM parameters as:

$$\omega_{i,0} = 0, \qquad \mu_{i,0} = M_0, \qquad \sigma^2_{i,0} = V_0^2 \tag{4.14}$$

where M_0 is set to 0, and V_0 is set to a larger number.

(2) Calculate the Mahalanobis distance d_i from an observed input sample I_t to a Gaussian distribution:

$$d_i = \Big((I_t - \mu_{i,t-1})^T(\sigma^2_{i,t-1})^{-1}(I_t - \mu_{i,t-1})\Big)^{\frac{1}{2}}, \tag{4.15}$$

where I_t corresponds to the intensity value of a pixel in the input image I and d_i measures the matching degree between I_t and the i^{th} Gaussian. Let T_d be the threshold of maximum Mahalanobis distance from a input sample I_t to the cluster of center. If d_i is smaller than threshold T_d, then I_t is categorized into the i^{th} Gaussian, and the corresponding parameters in the i^{th} Gaussian distribution are updated as:

$$\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} + \alpha, \\ \mu_{i,t} &= (1-\beta)\mu_{i,t-1} + \beta I_t, \\ \sigma^2_{i,t} &= (1-\beta)\sigma^2_{i,t-1} + \beta(I_t - \mu_{i,t-1})^2 \end{aligned} \tag{4.16}$$

where α denotes the learning rate of ω_i, β denotes the learning rate of μ_i and σ_i^2 . Rest of the parameters of the other $K-1$ Gaussian distributions remains the same. If d_i is larger than threshold T_d, i.e. the sample I_t cannot match anyone of the K Gaussian distributions, then all distributions are rearrange in a descending order according to the fitness function:

$$Fit_i = \frac{\omega_i}{(\sigma_i)} \tag{4.17}$$

The m^{th} Gaussian with the lowest fitness value by the following equation:

$$\begin{array}{ll} m = argmin_i(Fit_i) & \omega'_{m,t} = \alpha \\ \mu_{m,t} = I_t & (\sigma_{m,t})^2 = V_0^2 \end{array} \tag{4.18}$$

Moreover, all the weights are normalized as

$$\omega_{i,t} = \frac{\omega'_{i,t}}{\sum_{j=1}^{K} \omega'_{j,t}} \qquad i = 1, 2, ..., K \tag{4.19}$$

where $\omega'_{i,t}$ and $\omega_{i,t}$ are the original and normalized weights, respectively.

(3) Learning step (2) is performed until all samples are implemented. Then, all distributions are rearranged according to the fitness function 4.17 and the first B distributions are selected as the iutput background model,

$$B = argmin_b(\sum_{i=1}^{b} Fit_i > T_b), \tag{4.20}$$

where Tb is a threshold to select the number of Gaussian distributions for background model.

It should be noted that the number of K Gaussian distributions is related to the type of maritime scenario, i.e whether the images are captured in open sea where background regions typically do not contain many elements (and K can be sufficient smaller for modelling) or whether images are close to the coast/harbour where usually there are more elements that may increase the number of elements. In experiments where K = 3-5 the methods has proven good performance.

Ship detection using background subtraction

The way this method is implemented is explained now. Since ships may appear within both sea-surface and sky regions, background subtraction is implemented independently.

First step, needs of horizon estimation so both regions can be treated in two separately ways. Second step is to obtain a Gaussian distribution using learning method explained above for both sea-surface and sky regions. Now that two background models (each one for sea-surface and sky) are formed image subtraction can be implemented.

Figure 4.11 shows two different frames from the same video. In figure 4.11a background model is obtained and then implemented in figure 4.11b.

Learning of GMM for background subtraction using figure 4.11a is shown in fig. 4.12. As it was mentioned before, both regions (sea-surface and sky) are treated independently. In this case K is selected equal to 3 due to the fact that there are a few elements in both regions. In the bottom of the figure are shown the curves of the 3 different distributions for each region.

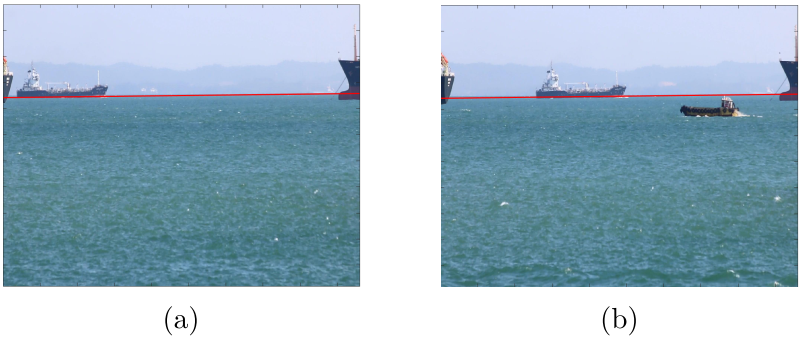
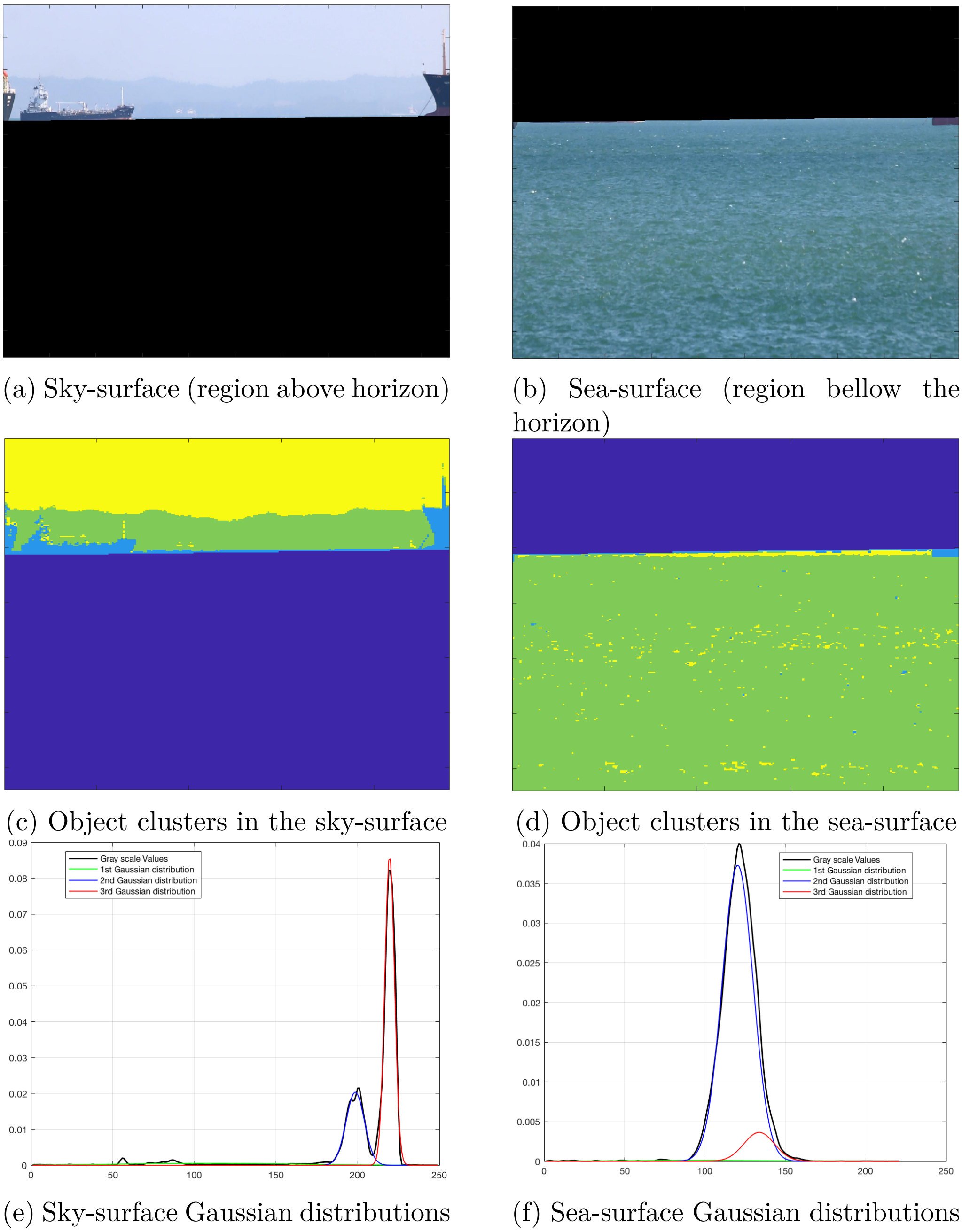


Figure 4.11: Illustration of background subtraction using Method B. (a) Image used to model background. (b) Image used for background removing from model obtained in Image (a).



(a) Sky-surface (region above horizon)

(b) Sea-surface (region bellow the horizon)

(c) Object clusters in the sky-surface

(d) Object clusters in the sea-surface

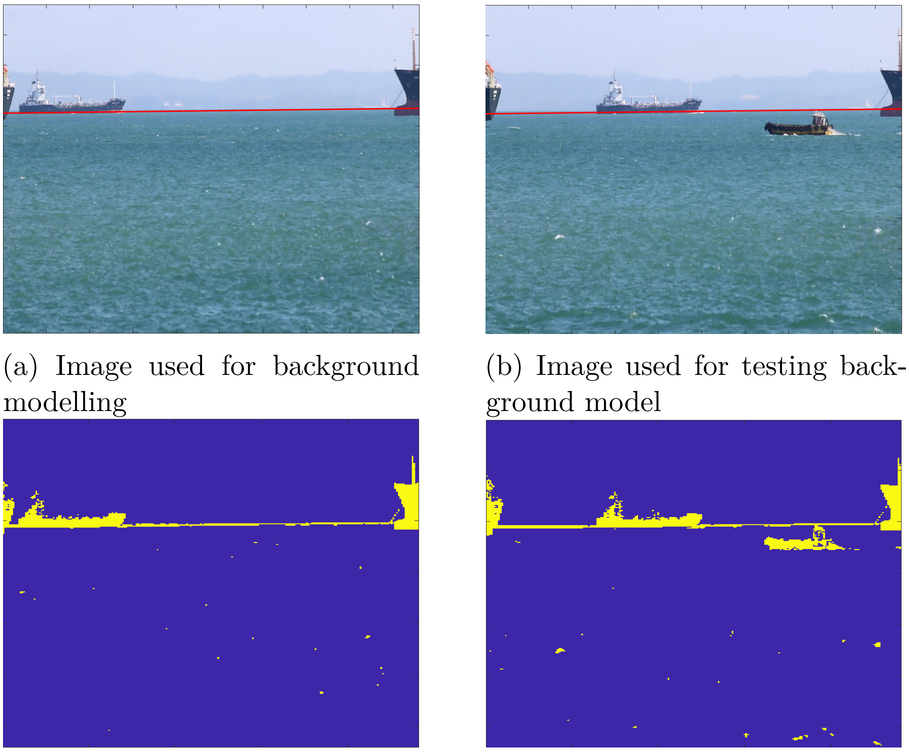
(e) Sky-surface Gaussian distributions

(f) Sea-surface Gaussian distributions

Figure 4.12: Learning og Gaussian Mixture Model Using method B

Figure 4.12 depicts different distributions using K equal to 3 in both cases. In the sea-surface region it can be seen that these 3 distributions are properly clustered according to the distribution. It is clear that the green region correspond (figure 4.12d) to the sea surface whereas yellow represents the foam produces by the waves. In the same figure, in the upper right corner a small piece (blue) of the boat it is also clustered in other distribution. In the case of the sky modelling it is more conspicuous the different clusters formed after the GMM learning. Sky appears in the yellow part of the image, mountains in the background an some small pieces of ships are represented with green and finally ships are depicted with color blue. This can also be represented with the curves in the bottom of the figure where for K=3 the 3 different Gaussian distributions are plotted. Curves with the largest peak and narrowest variance represents the sea-surface/sky in the picture. In the case of the sky, the second peak (2nd Gaussian distribution, blue color) describes the mountain whereas in the case of the sea the red curve describes the foam existing in the sea. The other remain distributions (green colors in both cases) represents the foreground regions.

Once the the two background models are obtained they can be implemented in futures frames. Procedure is applied in both regions. An image pixel is classified as foreground/region if the Mahalanobis distance d_i defined in equation 4.15 is larger or smaller than threshold T_d. This is repeated until all pixels are inspected.



(a) Image used for background modelling

(b) Image used for testing background model

Figure 4.13: Image subtraction using Method B

After background subtraction, the objects are enclosed in the bounding boxes by morphological operations. This part is presented later.

4.2.3.3 Method C

Method C is inspired by the work of Zhang, Li, and Zang [38], where novel sea-surface background modeling algorithm using DCT-based GMM was presented. Essentially, energies coefficients in each DCT block are calculated to feed the learning process of GMM. Once all the ocean regions are modeled, the objects in these regions can be detected by classifying each image block into background or foreground. The method consists of three main steps, namely, (1) decompose the luminance component of an input image in block and apply DCT to these blocks, (2) calculate coefficient in each block and (3) GMM learning. In order to make easier the understanding of the method figure 4.14 illustrates the steps. A color input is converted into grey scale values.

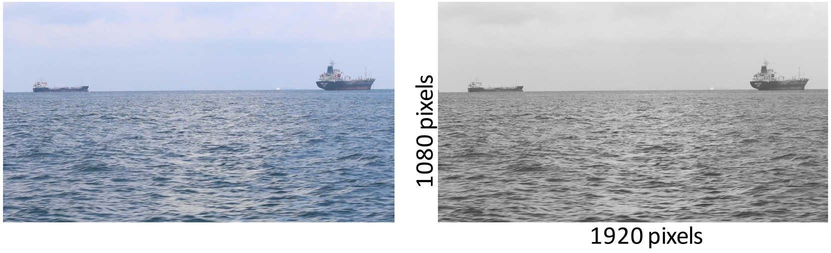


Figure 4.14: Example of image used for illustrate method C. Color image (left) and greyscale image

Image decomposition

First the image luminance component (greyscale image from figure 4.14) is decomposed into 8×8 non-overlapped blocks and the DCT is applied to these blocks:

$$A_{i,j} = \alpha_i \alpha_j \sum_{m=0}^{7} \sum_{n=0}^{7} I_{mn} cos \frac{\pi(2m+1)i}{16} cos \frac{\pi(2(n+1)j)}{16} \qquad i,j = 0,1,...7 \tag{4.21}$$

where I_{mn} is the pixel value at location (m, n) in the 8×8 image block, and A_{ij} is the DCT block. The normalized weight $\alpha_i = \frac{1}{2\sqrt{2}}$, if $i = 0$;otherwise, $\alpha_i = \frac{1}{2}$. Each 8×8 block includes 1 DC coefficient and 63 AC coefficients.

Texture-Based feature vector X generation

The textured-based features in sea-surface background can be considered as the spatial distributions of intensity variations. Zhang, Li, and Zang defines the texture feature as a three dimensional vector in DCT domain, as shown in Figure 4.16. The DCT

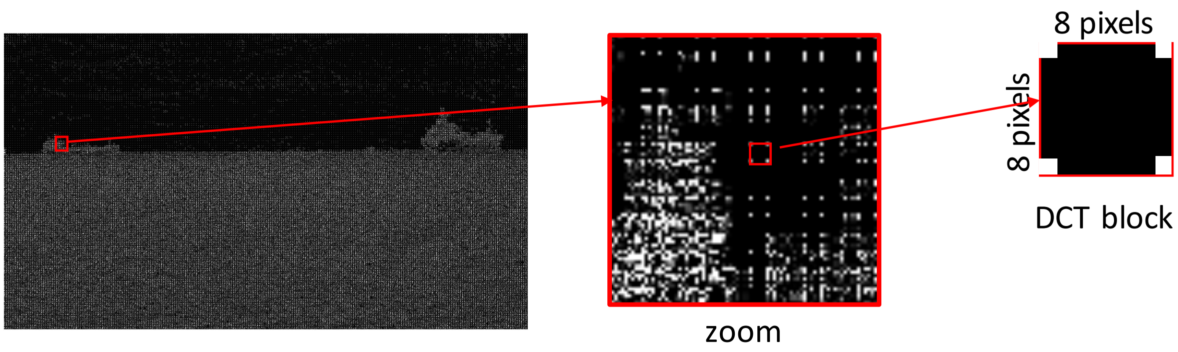


Figure 4.15: Decomposition of the input image in blocks of 8×8 and DCT application

coefficients depicted by different colors account the spectrum component in the corresponding direction. To elaborate, white region R_0 in the upper left corner denotes the direct-current component; R_1 (green), R_2 (yellow), R_3 (gray) regions represent the vertical, diagonal and horizontal frequency variation, i.e., horizontal, diagonal and vertical texture information.

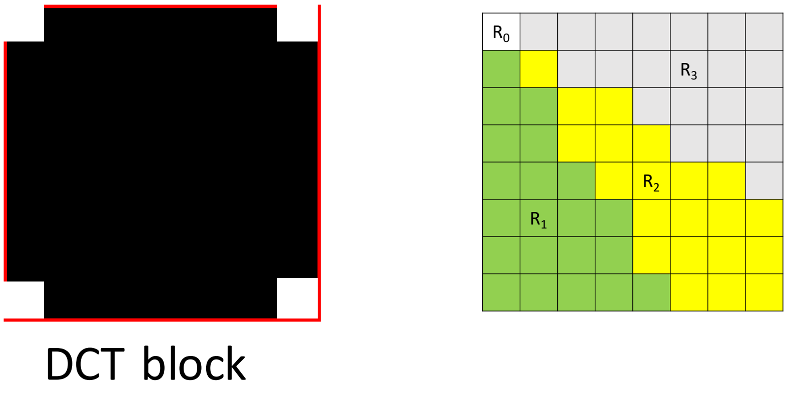


Figure 4.16: Frequency portioning for texture features in a 8×8 DCT block. .(Left) 8×8 block obtained from figure 4.15. (Right) Texture feature description where each color represents a frequency variation in space.

Next energies E_1, E_2 and E_3 of region R_1, R_2 , R_3 respectively are calculated to generate the texture-based feature X:

$$X = (E_1, E_2, E_3)^T \tag{4.22}$$

where the region $E_k(k = 1, 2, 3)$ is defined as follows

$$E_k = \sum_{i,j,R_k} (A_{ij} - \bar{A}_k)^2 \tag{4.23}$$

where $A_{i,j}(i.j \in R_k)$ are DCT coefficients in the region $R_k(k = 1, 2, 3)$, and $\bar{A_k}$ is the average of DCT coefficients of Regions R_k:

$$\bar{A_k} = \frac{1}{\mid R_k \mid} \sum_{R_k} A_{ij} \qquad k = 1, 2, 3 \tag{4.24}$$

This yields a corresponding texture feature vector denoted by $X_i(i = 1, 2, ..., N)$ for each block where N is the total number of blocks or feature vectors within the sea-surface background region. Now, it is possible to model the texture background of sea surface using these texture-based feature vectors.

Learning of Gaussian Mixture Model

Now the background surface is categorized into K clusters using GMM. Procedure is similar to the one explained previously for Method B. Let $D = X_1, X_2, ..., X_n$ account for a sample set of 3-dimensional feature vectors X defined in Eq. 4.22. Each sample corresponds to a DCT block from K clusters with a certain probability. To quantify, the probability of X_t based on Gaussian distribution is written as :

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_i), \qquad i = 1, 2, ..., K \tag{4.25}$$

which is similar to Eq. 4.12 with two main differences: (1) $\mu_{i,t}$ and Σ_i, t is the covariance matrix of the i^{ith} Gaussian in the mixture at time t, respectively and (2) that the Gaussian probability distribution η is defined as :

$$\eta(I_t, \mu_{i,t}, \Sigma_i) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \Sigma_i \mid^{\frac{1}{2}}} e^{\frac{1}{2}(I_t - \mu_i)^T (\Sigma_i)^{-1} (I_t - \mu_i)} \tag{4.26}$$

Since the DCT is an orthogonal transform, coefficients in a DCT block are independent of each other [34]. Therefore, the three component E_1, E_2, and E_3 (coefficient energies) of X are mutually independent. Further, the covariance of independent variables is zeros and the covariance matrix $\Sigma_{i,t}$ is a diagonal matrix:

$$\Sigma_{i,t} = \begin{bmatrix} \sigma^2_{1,i,t} & 0 & 0 \\ 0 & \sigma^2_{2,i,t} & 0 \\ 0 & 0 & \sigma^2_{3,i,t} \end{bmatrix} \tag{4.27}$$

where σ_1, σ_2 and σ_3 are the standard deviations of E_1, E_2, and E_3, respectively.

Two images are used to show the implementation of this method. Figure 4.17 shows the sample of images used for modeling the background. It includes one image (left) captured with a color camera and one image (right) captured with a monochrome camera. The results for the color image are shown in figure 4.18 where the number of Gaussian distributions K is changed. For K=2 this method achieves to distinguish between texture produced by sky an sea and the rest of the elements. For K = 3 the wakes produce by waves are also classify and the ship and two buoys are distinguish. For K = 4 and K=5 the results are almost similar. In both it is clearly identify the ship and buoys and the rest of the background (wakes, sea, sky...).

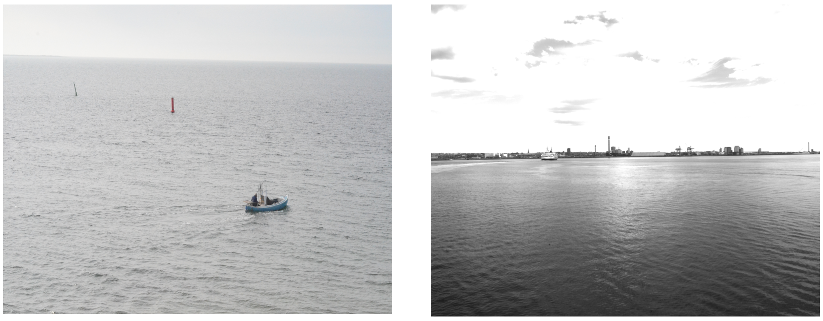


Figure 4.17: Example of two images for bacground modelling using texture features in 8×8 blocks. Color image containing one fishing boat an two buoys (left). Image captured from a monochrome camera containing a ferry and some building on land.

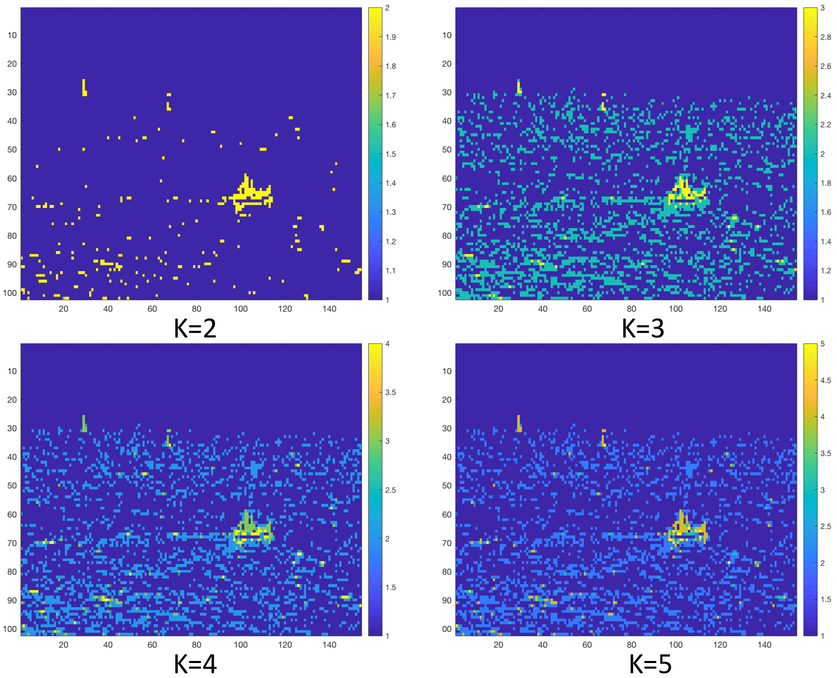


Figure 4.18: Results for background modeling using method C for color image. K represents the number of Gaussian distributions used for modeling

Figure 4.19 shows the result for the image captured with a monochrome camera. It includes a ferry and some buildings on land in the background. For K =2 it only classify in two categories and it sky, some parts of the building are one category. For K =3 sky, sea and building appear to be classified in three different categories. Also some reflections in the water allocated in the category of the buildings. For K=4 and K=5 it seems to perform similar and some elements of the land are categorized as well. However this is one of the worst scenarios for background modeling since it is very difficult distinguish where the different elements of the image are.

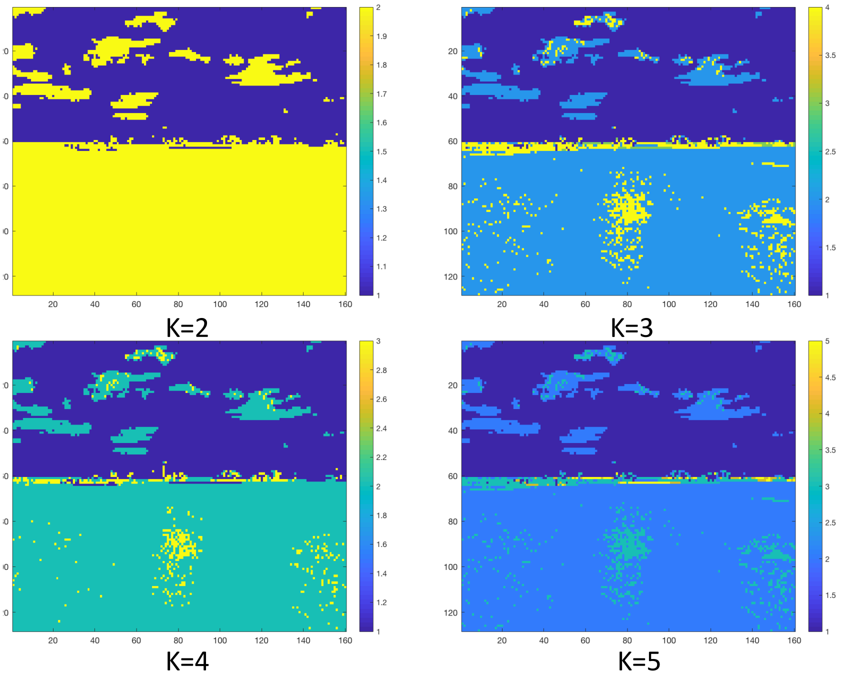


Figure 4.19: Results for background modeling using method C for image captured with a monochrome camera. K represents the number of Gaussian distributions used for modeling

This demonstrates that the method performs correctly and it shows that the number of K Gaussian distributions is related to the type of maritime scenario.

4.2.4 Foreground Segmentation

After background subtraction, objects are enclosed in bounding boxes by morphological operations as it is described in Prasad et al. [25]. Morphological operations used in a given image P consist of morphological opening and closing, which are the combinations of dilation and erosion using the same structuring element Q for both operation:

$$Opening : P \circ Q = (P \ominus Q) \oplus Q \tag{4.28}$$

$$Closing : P \bullet Q = (P \oplus Q) \ominus Q \tag{4.29}$$

where $\oplus$ and $\ominus$ denote the dilation and erosion, respectively. In this project, the improved morphological close-minus-open (CMO) technique presented in Westall et al. [37]

is applied to enhance the foreground segmentation:

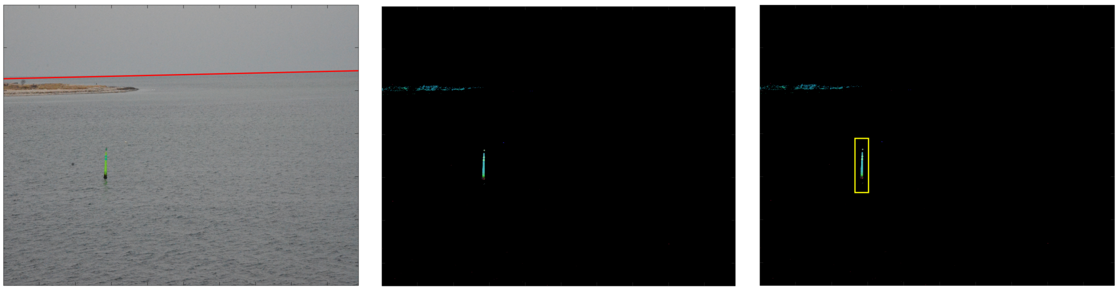
$$I' = (I - (I \bullet SE)) + (I - (I \circ SE)) = 2I - ((I \bullet SE) + (I \oplus SE)) \tag{4.30}$$

where SE denotes the structuring element, I denotes the input image containing foreground, and I' denotes the enhanced foreground results.

Implementation of this method can be seen in fig 4.20. First the horizon line is estimated so we can subtract both sea-surface and sky background (figures 4.20a and 4.20b). Second step is removing background using in this case Method B (figure 4.20c). Then morphological operations are applied; first morphological closing and second morphological opening (figures 4.20d and 4.20e). Structuring element used with this method was a rectangle of 16 × 1 dimensions had a good performance in some images for removing line created in the horizon 4.20d. This structuring element is parameter to consider due to the fact, that a large value will remove objects that are far from the camera whereas a small value will result in small object that will no be erased. Last step is to create a bounding box using a label for each of the blocks showed in figure 4.20e.

4.2.5 Classification

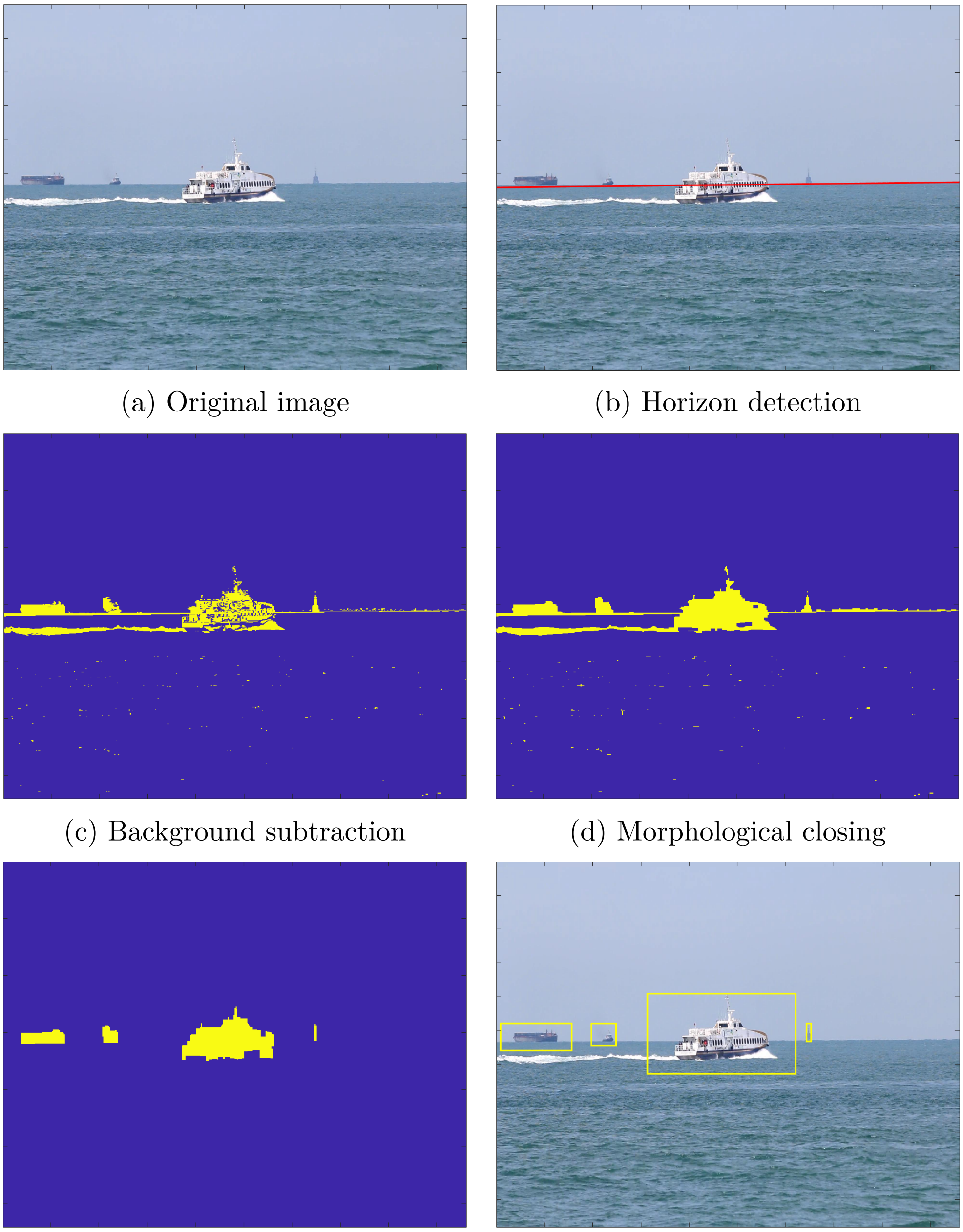
Once the RoI is found as it is shown in Figures 4.20 and 4.21, classifiers appear. The aim of a classifier is to categorize or distinguish an element based on some patterns defined by the category or class that represents. For instance, a buoy has some features that differs from features of a ship or a kayak. Therefore, classification needs samples to train as well as a definition of each category. In this project two main classifiers have been used, i.e Support Vector Machine (SVM) classifier and Convolutional Neural Networks (CNN) classifier. In the following sub-sections this methods are presented as well as some examples that illustrate their performance.



(a) Horizon detection (b) Background subtraction (c) Foreground segmentation

Figure 4.21: Object detection using Method A

Categories used for training classifiers and detectors (this last is presented in the following section) and the number of images used for each category are shown in table 4.4.



(a) Original image

(b) Horizon detection

(c) Background subtraction

(d) Morphological closing

(e) Morphological opening

(f) Egde detection

Figure 4.20: Bounding boxes after morphological operations

Some basic definitions are given below in order to aid the comprehension of the sections that are now presented.

- **Feature vector**: is just a vector that contains information describing an object's important characteristics. It can take many forms; a basic feature representation of an image can be the raw intensity of a pixel or a more complex feature can be circularity, gradient magnitude or gradient direction.

- **Supervised Learning**: machine learning task where a funtion is shaped by inference data in the form of input-output pairs.
- **Unsupervised learning**: machine learning task where a function is inferred from just input data.

Category	**images**	**Category**	**images**
Buoys	94	Fishing boat	413
Cabin Motorboat	751	Large sailboat	941
Cargo Ship	844	Open motorbooat	220
Cruise Ship	704	Personal watercraft	177
Ferry	659	Small boat	593
Small sailboat	700		
Total number of images	6096		

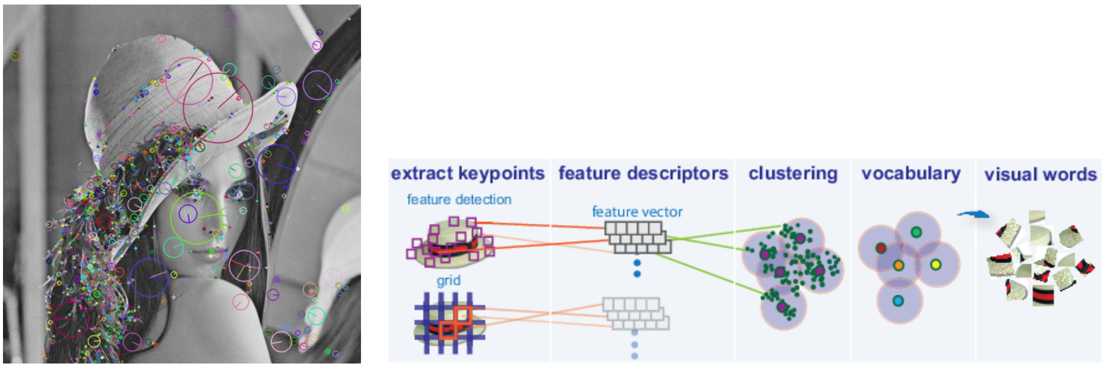
Table 4.4: Categories used for training classifiers and detectors. Repeated for convenience from table 4.4

4.2.5.1 Classification using SVM

This section containing SVM classifier was extracted from Eduardo Vazquez thesis and the following is a summary of his results.

A Support Vector Machine (SVM) classifier is a multi-class image classifier, that assigns a given image to a certain category within a finite range of categories. It is considered a supervised learning method, since instances for every class are provided through annotated images. The process for building and applying such a type of classifier consist of three main steps: building a vocabulary, training the classifier and applying the classifier to images.

A visual vocabulary is composed by a collection of visual words, i.e, mutually exclusive groups, each of them classified according to features extracted from images. Features are recognizable structures of elements in the environment, and can be processed using several techniques. Features are important to consider since they can help us to describe numerically some regions of an image. Most popular techniques are the Scale Invariant Feature Transform (SIFT) or Speeded Up Robut Features (SURF). SIFT method is widely used and robust to rotation and small changes of illumination, scale, and viewpoint whereas SURF method is inspired by SIFT but several times faster [28]. Visual words are assembled by firstly, extracting features from a number of images and secondly, grouping those features by means of clustering. This clustering is an iterative process that is processed until it results in compact and distinctive groups of similar features, that correspond to visual words. The clustering process used is a simple partitioning method: k-means clustering, that iteratively assigns features to their closest cluster centers and recalculates those centers.

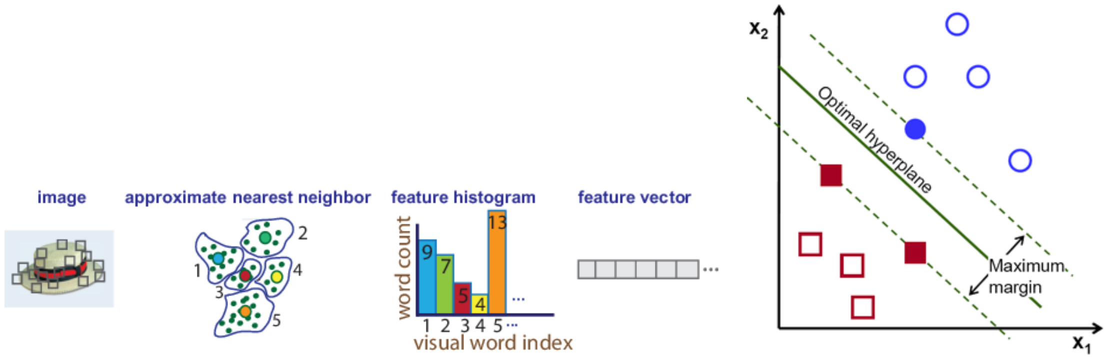


(a) SURF features

(b) Visual vocabulary pipeline

Figure 4.22: Bag of visual words

Once visual words are obtained for each image through feature extraction, SVM classifier training is implemented. SVM classifier takes the visual vocabulary as an input and produces a multi-class classifier. Figure 4.22 shows the process to train the SVM classifier. First, every visual word obtained on images used for training is encoded into a visual word histogram. This procedure is employed for all images in the training set. Further, each histogram, also called feature vector or support vector, is stored and finally the application of the SVM algorithm allows to distinguish between classes by means of optimal hyperplanes.



(a) Procedure for obtaining feature vector from visual word histogram

(b) Illustration of SVM performance

Figure 4.23: Illustration of feature vector and SVM

Once the SVM classifier is trained by using a large and representative set of images from different classes it can be applied to new images so it assigns them to one of those classes or categories. Procedure is straightforward, for each image, features are extracted and the feature histogram is formed. Then, the image is assigned to the category to which its histogram belongs according to the division drawn by planes in the histogram spaces.

4.2.5.2 Classification using CNN

This section containing CNN classifier was extracted from Eduardo Vazquez thesis and the following is a summary of his results.

Now the use of Convolutional Neural Networks (CNN) classifier is described. CNN training is made by first, obtaining an optimal classifier and then testing it in new images to perform the classification task on them. Procedure implemented such as CNN training and relevant learning technique (transfer learning) is covered in this sub-section.

An appropriate training of a CNN consists on determining the optimal values of the model parameters such as weights and biases. Similar to SVM, every categories are defined previously through annotated images. During the training, models are adjusted so the probability of the model performing a correct classification increases. Training is completed when this process is completed for every image of each category. This probability is reflected on an error function minimization, which is backpropagated so that the model parameters are updated to the optimal ones.

In order to have a much faster training process transfer learning has been implemented. Transfer learning is a particular method used very often in machine learning for training purposes. For reducing the time of the process an already trained neural network is used as starting point for the development of the new model. This neural network is built with different classes to those of the original training. The training consist of retraining only the last layers, those specifically trained for the original set of classes, so the are adapted to the new set of classes. General scheme can be seen in figure 4.24 Once a CNN has been trained, it can be used to classify new images. For every input image, the CNN outputs an array of scores relative to the classes which represents the probability that the image belongs to them. Then the image is assigned to the class with the highest score.

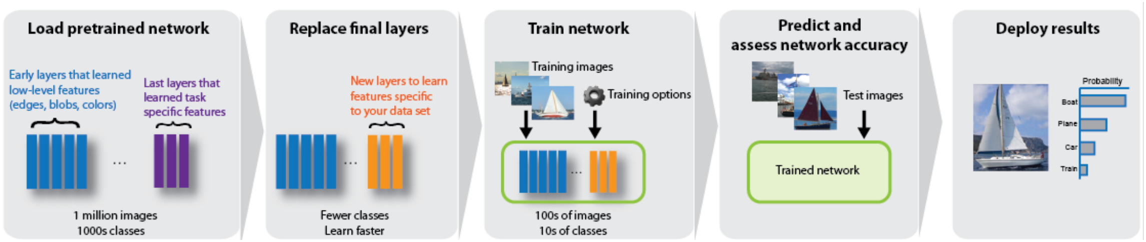
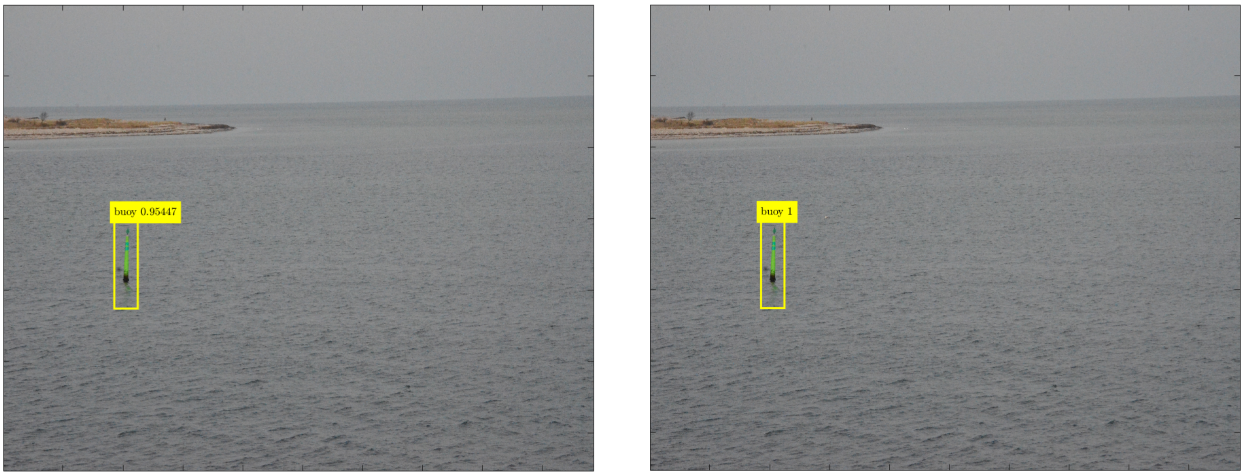


Figure 4.24: General scheme used for training CNN classifier

An implementation of this two methods for classifying bounding box form figure 4.21c is shown in figure 4.25. The score of the classification appear in the top of the bounding box. This score is a percentage output in the classifier function.



(a) Buoy detection using SVM classifier

(b) Buoy detection using CNN classifier

Figure 4.25: Implementation of Classifiers

4.2.6 Summary

Method 1 for object detection using computer vision was presented in this subsection. This method include several steps that can be seen in figure 4.1. We can distinguish two main parts in this method, namely Region of Interest (RoI) detection and Classification of this RoI. Since the aim of object detection is not only detecting the objects but also classify them both part are important. The algorithm presented for detecting the RoI consists of three main steps; first the horizon detection, second background subtraction (sea and sky) and finally foreground segmentation which in this case uses morphological operations. The background modeling approach for ship detection was the Method B which consists of using the value of the pixel for background modeling. In the case of buoys the method implemented includes a manually selection of the sea background region. Foreground segmentation includes morphological operations that aims to erased small objects in the image and enclosed different parts of an object that can be segmented. Once the RoI are defined, a pre-trained classifier has the duty of classifying the different bounding boxes contained in an image.

4.3 Object detection using Method 2

Method 2 is described in this section. As it was described previously, the main difference between both methods is that detection and classification takes places internally in the case of Method 2. This section containing ACF detector and Faster RCNN was extracted from Eduardo Vazquez thesis and the following is a summary of his results.

4.3.1 Detection using ACF

An ACF detector is a single class detector that is trained through supervised learning to identify objects of a certain class in an image. Work presented in this sub-section is based on Dollar et al. [7]. This method mainly consists of two stages: training of the detector and implementation in images. Training process is composed of channel features computation and learning process.

In the channel feature computation, a number of channels are computed (such as normalized gradient magnitude or histogram of oriented gradients) for every training image as it is shown in figure 4.26. Then, pixel blocks are summed and smoothed for every channel. Features correspond to single pixels lookups in those aggregated channels.

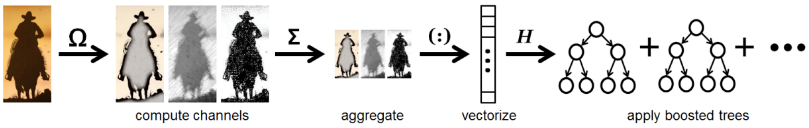


Figure 4.26: Scheme for ACF training

The learning process is considered as supervised and uses the AdaBoost (Adaptive boosting) algorithm [5]. This algorithm creates a classifier of a given category by combining weak classifiers. Weak classifiers are basically decision trees as shown in figure 4.26 (right) that from an given image it yields in a binary output, namely, whether the image belongs to that category. Each weak learner has a weight that iteratively chosen based on its accuracy on the training set. Therefore accurate decision tree models contribute more to the final estimation. This process yields in a strong learner.

Once the ACF detector is obtained is ready to be implemented. For every image, the detector will find features in order to identify objects of a certain category in each image.

4.3.2 Detection using Faster RCNN

A Faster RCNN detector is a multi-class detector. It is composed of a modified CNN, that includes a layer known as Region Proposal Network (RPN), which aims outputting Regions of Interest (RoI) from images that can later be classified by the latest layers of the CNN. Therefore, training of Faster RCNN involves four consecutive training steps, two for specifically training the RPN and two for training last layers of the CNN. Figure 4.27 shows Faster RCNN structure of how the algorithm works. One may notice that the way the algorithm detect objects is similar to the method presented in the previous section (Method 1).

The algorithm consists of 4 main steps that are now described. First, there are a series of convolutional layers that extract features from input images, creating activation maps. Then based on these activation maps, RPN outputs regions of interest. Next, all

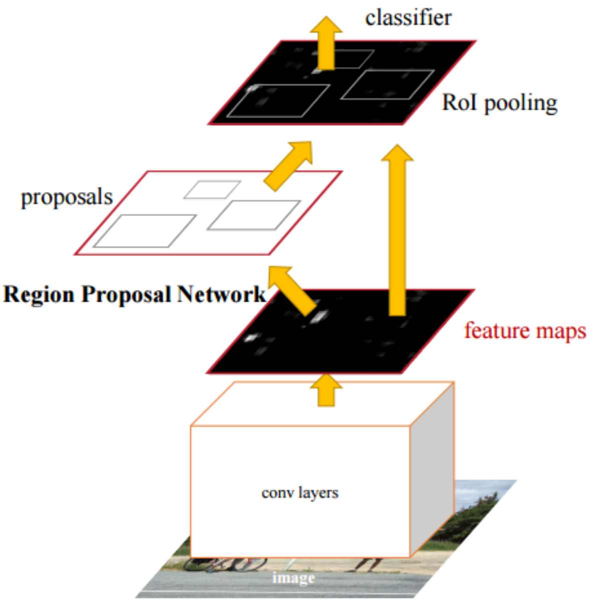


Figure 4.27: Faster RCNN scheme

this RoI are passed to other convolutional layers, that finally output the classification scores for every RoI.

Regarding training, it is supervised and takes place in the following steps:

- Initial RPN training: all parameters of the RPN are optimized using a set of training imags with annotated bounding boxes.
- Initial CNN training: parameters belonging to the rest of the layers of the CNN is trained for classification of the image regions proposed by the RPN.
- RPN retraining: this step constitutes a fine-tuning of the parameters obtained from the first of the training.
- CNN retraining: this step is the equivalent to the third step for the CNN layers that are not part of the RPN. The parameters of those layers are initialized to the values obtained at the second step.

4.3.3 Summary

This section presents the two main detectors algorithms used in this section. One, ACF detector training consists of computing several features for each input image and then used weak classifiers that yields in a binary output, namely, whether the image belongs to that category. On the other side, Faster RCNN detector is composed o a modified CNN where the RoI is computed in an other layers of the neural network.

4.4 Results

Results for horizon and object detection are presented in this section.

4.4.1 Horizon detection results

In order to give verify the effectiveness of the proposed method, two different tests were compared from the different results available. First, a comparative with other methods used in [25]. Secondly, an evaluation of the precision and recall scores is accomplish to analyze the performance of the method proposed.

Comparative among other methods
A qualitative comparison of the methods employed in [25] and the method proposed in this project is provided in table 4.5. For quantitative comparison on Singapore Maritime dataset, the representation of horizon shown in fig. 4.28 is used. Where Y is the distance between the center of the horizon and the upper edge of the frame and α is the angle between the normal to the horizon and the vertical axis of the frame. Then it is used the position error $\mid Y_{GT} - Y_{est} \mid$ and the angular error $\mid \alpha_{GT} - \alpha_{est} \mid$ as performance metric. Where the sub-index GT makes reference to the real horizon line and est to the estimate horizon.

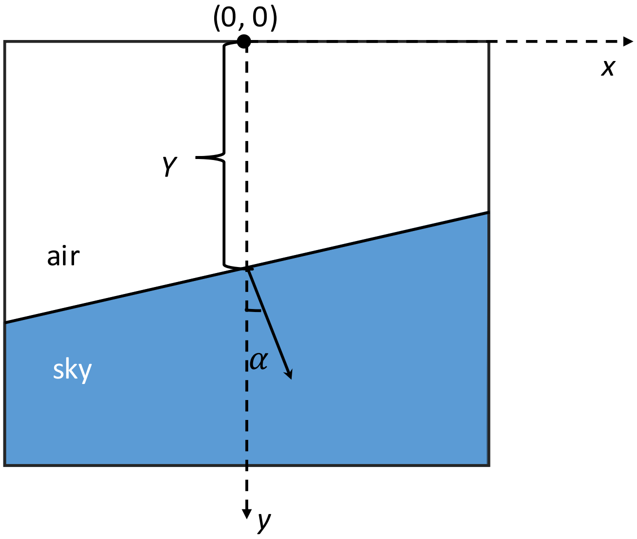


Figure 4.28: Representation of horizon for quantitative comparison of horizon detection

The results provided by [25] include Hough transform [14] (referred to as Hough), Radon transform [15] (Radon), multi-scale median filter [4] (MuSMF), Ettinger et al.'s method [8] (ENIW), and Fefilatyev et al.'s method [12] (FGSL). Hough and Radon are projection based, MuSMF, ENIW and FGSL are alternative methods to projection based method. It is clearly see that the method proposed has a better performance for both on-shore and on-board videos among the other methods. Time of detection was also compared as an average of the processing time per frame for each method. In the case of the method proposed the real-time performance was assessed on an Intel i5 Macbook Pro with 8GB RAM. As noted, method proposed has the shortest average processing time among the other methods.

	Position error (pixels) $\mid Y_{GT} - Y_{est} \mid$				Angular error ° $\mid \alpha_{GT} - \alpha_{est} \mid$				Time/ frame (s)
	Mean	Q25	Q50	Q75	Mean	Q25	Q50	Q75	
On-board videos									
Hough	219	131	229	295	2.6	0.6	1.7	3.4	0.3
Radon	372	213	362	517	40.6	1.5	3.4	87.7	2.7
MuSMF	269	156	283	379	1.8	0.5	1.2	2.5	0.9
ENIW	120	63	116	166	1.9	0.5	1.2	2.5	hours
FGSL	120	63	117	165	1.8	0.5	1.2	2.5	12.8
Method Proposed	**7.8**	**0.1**	**21.12**	**3.94**	**0.44**	**0.11**	**0.22**	**0.42**	**0.26**
On-shore videos									
Hough	208	26	194	354	1.2	0.2	0.7	1.5	0.26
Radon	313	28	359	549	32.9	0.2	0.4	88.1	2.0
MuSMF	60	25	49	85	1.2	0.2	0.4	1.1	0.9
ENIW	121	15	94	163	1.2	0.2	0.4	1.3	hours
FGSL	112	12	91	162	1.2	0.2	0.4	1.1	12.3
Method Proposed	**16.4**	**1.6**	**3.5**	**11.5**	**0.3**	**0.13**	**0.19**	**0.31**	**0.16**

Table 4.5: Quantitative Comparison of methods for horizon detection. The smallest error in each column is indicated in bold.

Evaluation of the precision and recall rates

In order to verify the effectiveness of the proposed horizon detection, precision and recall rate are adopted:

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{4.31}$$

where N_{TP} is the number of true positives, N_{FP} is the number of false positive, and N_{FN} is the number of false negatives. Here, when the method estimates the horizon detection either that estimation corresponds to the real one (TP) or that estimation does not matched the real horizon (FP). On the other hand, if the methods estimates that it did not matched the horizon either that estimation is actually true (TN) or the estimation is corrected but the algorithm thinks that it did not matched the horizon line (FN). Clearly, the higher the precision and recall are, the better the horizon detection performance is.

Table 4.6 depicts the results of the set of values that determines the precision and recall for different datasets. The selection of those that sets include images taken in Hundested and Singapore where the different scenarios in the images contain ships, land, several weather conditions (images containing haze, clouds) and several objects that challenged the horizon detection. Singapore dataset also includes NIR images.

The average precision and recall are 90.1% and 98.4%, respectively as listed in table 4.6. This means that from the total 1210 number of images a 98.4% of the horizon lines were detected correctly and that the algorithm achieves a 90.1% of horizon detection precision. It is noticeable, that approximately 8% of the horizon are false positive meaning. This is due to the fact that some of the images (Data Sing (on shore)) contain a bunch of ships that prevent the correct detection. However nearly a 90% of the estimation are TP and that gives us a good values for precision and recall rates. An assessment of the time is also performed which results in 0.13 seconds per image. Such an effective horizon detection method, provided a solid basis for the following background modelling and object detection.

Database	number of images	TP	TN	FP	FN	Prec.	Recall	Time(s)
Data collection 1	126	105	2	14	5	0.882	0.955	0.164
Data collection 2	46	40	3	3	1	0.930	0.976	0.185
Data collection 3	50	40	0	10	0	0.800	1.000	0.177
Data Sing. (on board)	290	281	0	9	0	0.969	1.000	0.075
Data Sing. (on shore)	399	340	5	53	1	0.865	0.997	0.083
Data Sing. (on shore) NIR	299	273	8	11	7	0.961	0.975	0.087
Total	1210	1079	18	100	14	0.901	0.984	0.128

Table 4.6: Precision and recall evaluation.

4.4.2 Object detection results

In order to evaluate the performance the detections enclosed in bounding boxes are compared against the ground truth objects and judged to be true or false positives by measuring bounding box overlap. A detection is considered as correct (true positive) if the intersection over union (IOU) ratio of the bounding box eq 4.32 is larger than 50% [9]:

$$IOU = \frac{Area(B_d \cap B_{gt})}{Area(B_d \cup B_{gt})}, \tag{4.32}$$

where $Area(B_d \cap B_{gt})$ denotes the intersected area (number of pixels) of the detected and ground truth bounding boxes, $Area(B_d \cup B_{gt})$ denotes their union. Associated objects with IOU smaller than 50% or unassociated objects are labelled as false positives. False negatives are ground truth objects that are not matched. Figure 4.29 shows the overlapping area between the ground truth region and the detected region. Here it is also used the precision and recall rates from equation 4.31 to measure the performance of the model. Clearly, a high value of precision means a low value for false alarms, e.g, false detection of a ship can be derived from a low precision.

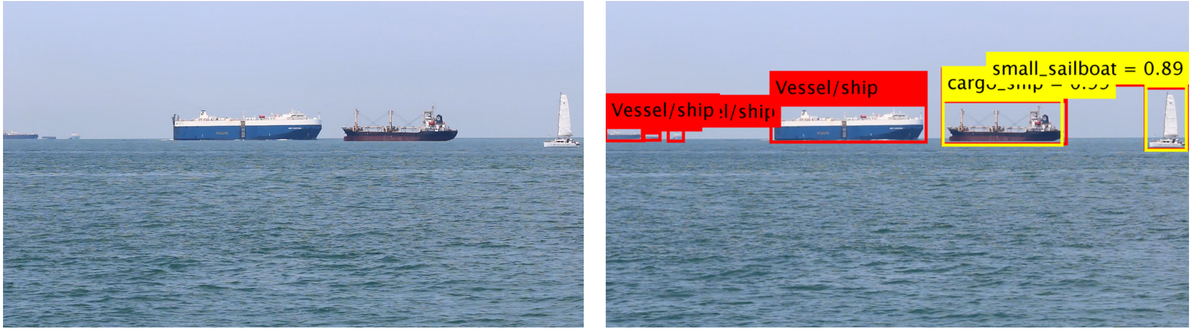


Figure 4.29: Illustration of IOU for object detection

Original Image (left). On the right picture ground truth objects (red boxes) and detection results (yellow box with name of the object and score)

The way results are presented are in a comparative structure. To elaborate, two different datasets were used including visual and NIR images. One of the datasets contains images taken in Hundested (Figure 2.6) and the other dataset is a selection from the Singapore database (Figure 2.4). Results of the 4 approaches include an average of the precision and the recall for each of the datasets and some relevant precision-recall curves.

A brief repetition of the 4 different approaches is made for convenience. Method 1 includes Method 1a and Method 1b. They both consists of a first detection of the RoI and second classification. Detection is the same for both methods while classification is different. Method 2 includes Method 2a (ACF detector) and Method 2b (Faster RCNN detector).

The comparison results are shown in Table 4.7. The table shows the results for the localization of the objects, i.e, the ROI finder part of Method 1 and the detection in Method 2. Therefore the values for classification are not included in the table due to the great amount of categories employed in the classification. It can be seen that Method 1a and 1b will have the same results, that is why results for that detection is fused in the same raw.

It is clearly, that the best performance for object detection is using Method 2b. The Method 2b outperforms the other two methods for detecting a RoI in terms of precision and recall rates. From the Average column, we can see that Method 2b achieves the highest precision (83%) for the test images but the other two methods only obtain a 1.5% and 0.4%. Further, in the case of the NIR images it has a 95% of precision. However, it can be also seen in the Average column the low value for the recall rate for all 3 methods.

Method	**Hundested**		**Singapore**				**Average**	
			Visible		**NIR**			
	Prec.	Recall	Prec.	Recall	Prec	Recall	Prec	Recall
Method 1 (RoI finder)	0,8%	6,0%	**3,0%**	**20,0%**	0,6%	0,4%	**1,5%**	**8,8%**
Method 2a	0,1%	9,0%	0,4%	**20,0%**	**0,6%**	8,0%	**0,4%**	**12,3%**
Method 2b	60,0%	7,0%	94,0%	**24,0%**	**95,0%**	10,0%	**83,0%**	**13,7%**

Table 4.7: Results for object detection

In the case of Method 2b, only a 13.7% of the objects are detected and with the other two detectors only obtain a 8.8% and 12.3% of the objects, respectively.

In order to understand those results a discussion of the results is done. In the case of Method 1 , the poor performance is due to the background content variety in the test images. The different changes in the background may have cause the poor performance. These results shows how sensitive is this method for changes from one background to another and the need of an appropriate tuning of the parameters as K (number of distribution in the GMM) and T_b (the threshold to select the number of Gaussian distributions). High dynamic backgrounds, e.g, large waves, reflections, sheen (in the case of sea), and clouds, sun rays, lands, mountains (in the case of sky background modeling) will need a large K whereas for flat waters and a color uniform color distribution K will be smaller. In this experiment, K was set to 3 for both sea and sky since for some test images was working perfectly Further, morphological operations relies on the Structuring element employed in the performance. Therefore the amount of parameters and thresholds of this method make difficult the achievement of a good performance. Certainly, these set of parameters needs to be update depending on the maritime scenario.

Method 1 and Method 2a perform almost similar, neither providing adequate precision and recall.

Finally method 2b has better performance in terms of precision. This state of the art technique shows good results compared with the other conventional method for detecting objects. The suitable and large dataset used for training provides high precision score. However, the low value in the recall is due to the fact that in some images farther ships are not detected by the algorithm (Figure 4.31). Therefore when the method is evaluated and compared with the ground truth object yields a poor recall result. This can be seen in Figure 4.30, which depicts the precision-recall curve for the Singapore visible dataset.

When Method 1 detects something that it considers as an object, that object is then classified. The performance of these classifiers with the tests showed in Table 4.7 are difficult to evaluate due to the low scores in precision and recall for Method 1. However during the experiments, the CNN (Method1b) classifier showed a better performance than the SVM (Method1a) classifier. This can be seen in the figure 4.31 where some far ships are classified as buoys for Method 1a meanwhile Method 1b classifies those ships

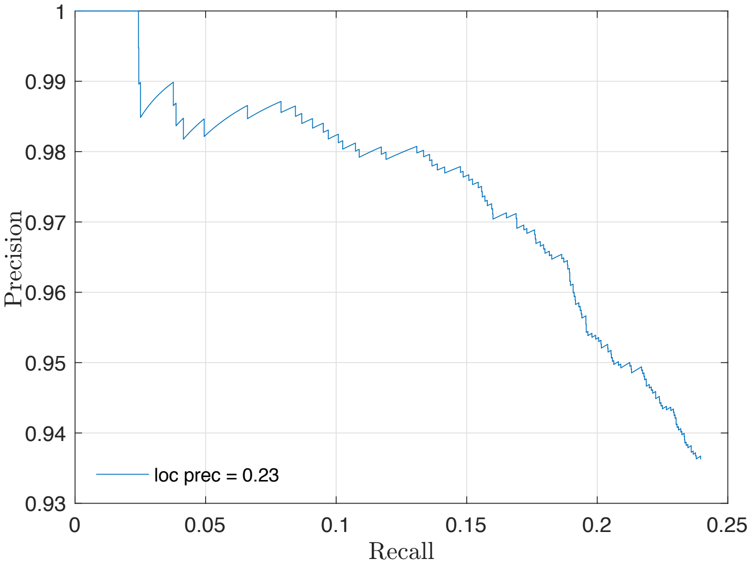


Figure 4.30: Precision-recall curve of the Method 2b for object detection from the dataset of the visible camera from Singapore

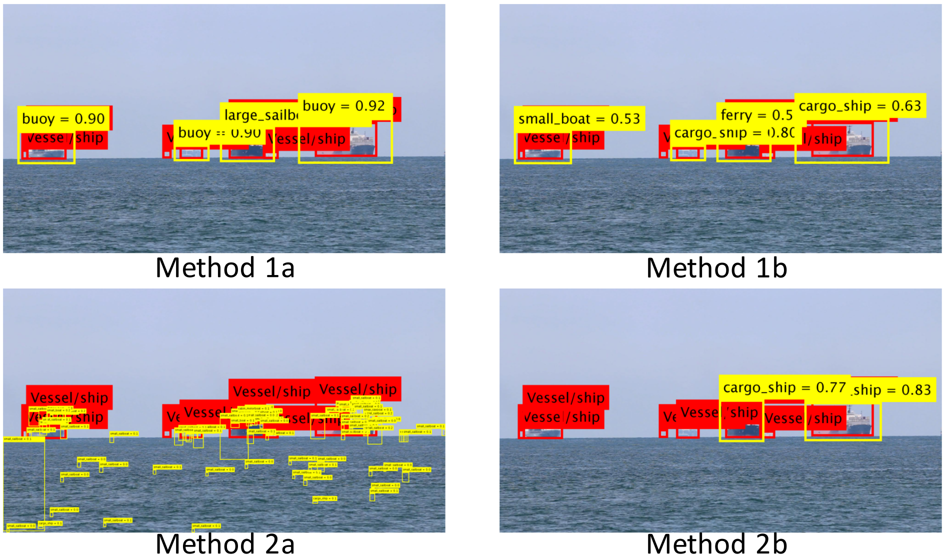
The ending point of the curve indicates the final precision and recall rates

within the ships categories.

Figures 4.31, 4.32 and 4.33 presents the comparison of varying scene content using the four object detection methods. Boxes in red color represent the ground truth region and the classification of the object. Yellow boxes shows the objects detected by the algorithm.

Method 1a, 1b and 2a all suffer from false detections caused by bad visibility, wakes, waves. The different background and shapes in waves in images produces these false detections. As a result, some small regions in the sea-surface background are modelled incorrectly as foreground, due to the fact that waves introduce a high contrast with the respect surroundings. This problem becomes more serious in the case of Method 2a where the number of RoI detected are far from detecting relevant objects in the images. This might be caused by the process the ACF detector uses for detecting objects. The lack of a algorithm inside the detector that deals with the correct location of the object may produced the poor result in detecting the object. The use of the AdaBoost algorithm may cause the poor performance when it needs to classify the object.

In comparison, Method 2b does not label any sea region as an object what explains its good results for the precision score. Consequently this method success in detecting correctly objects and in its posterior classification. This confirms that the Faster RCNN using neural networks will lead to a good performance with an appropriate training. However, this method fails in detecting objects that are far as it is shown in figure 4.31. Far ships are in a low resolution what may affect to the detection and a posterior precise



Method 1a

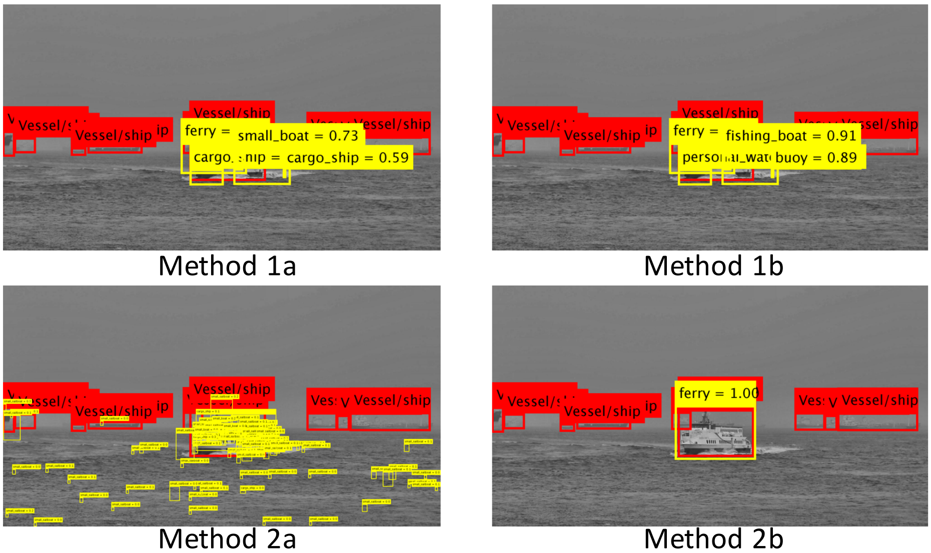
Method 1b

Method 2a

Method 2b

Figure 4.31: Detection results by the 4 methods. Image from Singapore visible camera dataset

classification.



Method 1a

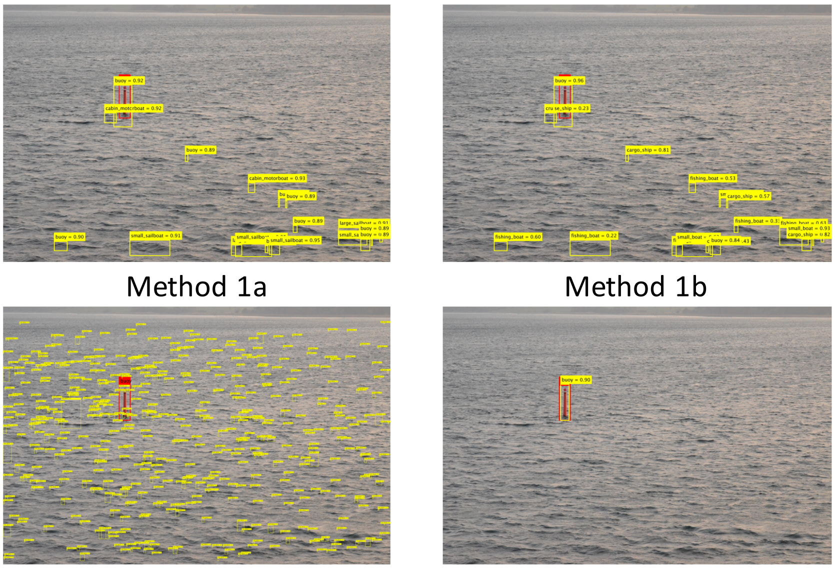
Method 1b

Method 2a

Method 2b

Figure 4.32: Detection results by the 4 methods. Image from Singapore NIR camera image dataset

We asses the real-time performance on an Intel Core i7 with 8GB RAM. All the algorithms were implemented in Matlab. Table 4.8 gives the average processing time per image by the 4 detectors and classifiers algorithms. As it can be seen, Method 2a consumes the longest time whereas Method 1a and Method 2b processing time are



Method 1a

Method 1b

Method 2a

Method 2b

Figure 4.33: Detection results by the 4 methods.Image from Hundested dataset

shorter among the other two Methods. Such a large time for Method 2a arises from one reason: the algorithm that process and extract features suffers from false detections that subsequently have to be classified.

	Method 1a	**Method 1b**	**Method 2a**	**Method 2b**
time (ms)	**3586**	5340	11636	3743

Table 4.8: Average processing time per image

4.5 Summary

Four different approaches for object detection and classification at sea has been designed and compared. According to the different approaches they can divided in to classes depending on the technique employed, namely RoI finder and classifier (Method 1) or detector (Method 2). The input (image) and the output (object detected and category of the object) are the same for both methods.

Method 1 first, finds a region where an object is detected trough the RoI algorithm. To elaborate, an algorithm to detect the region of interest is designed and tested. The experimental results with various scenes have demonstrated that the proposed horizon detection approach can achieve both improved horizon detection accuracy and enhanced real-time performance in comparison to traditional horizon detectors. Once the horizon is detected the background is subtracted. In order to find an appropriate background subtraction, 3 different approaches have been presented and considered. Due to the complexity to adjust on of the approaches, only were implemented two of them. When the background is suppressed in the image and after some morphological operations to enhance the detection, classification is performed. To proceed, two different classifiers were used, namely SVM and CNN classifiers. These techniques consists of extracting features from the RoIs and classify them according to some parameters trained off-line. The experimental results for object detection led into poor results due to the need of tuning the parameters for detection.

Method 2a is a single class detector that is trained to detect the object of certain class in an image. However the results show that the process inside the algorithm is not accurate enough for detecting objects. This might be caused by the process the ACF detector uses for detecting objects. The lack of a algorithm inside the detector that deals with the correct location of the object may produced the poor result in detecting the object. The use of the AdaBoost algorithm may cause the poor performance when it needs to classify the object.

Method 2b is a multi-class detector that is composed of a modified CNN that includes a layer that outputs the region of interest that can be later classified in the latest layers of the CNN. This method outperforms among the other approaches due to the intensive training with a pre-trained CNN that aims to detect objects. This techniques achieves a 83,0% and 13,7% of averaged precision and recall for the 3 datasets used for training.

CHAPTER 5
Electronic Sea Chart

When a mariner navigates into an unfamiliar, he/she uses a nautical chart to familiarize him/herself with the environment, determine the locations of hazards, and decide upon a safe course of travel. An autonomous surface vehicle (ASV) would gain a great advantage if, like its human counterpart, it can learn to read and interpret the information from a nautical chart.

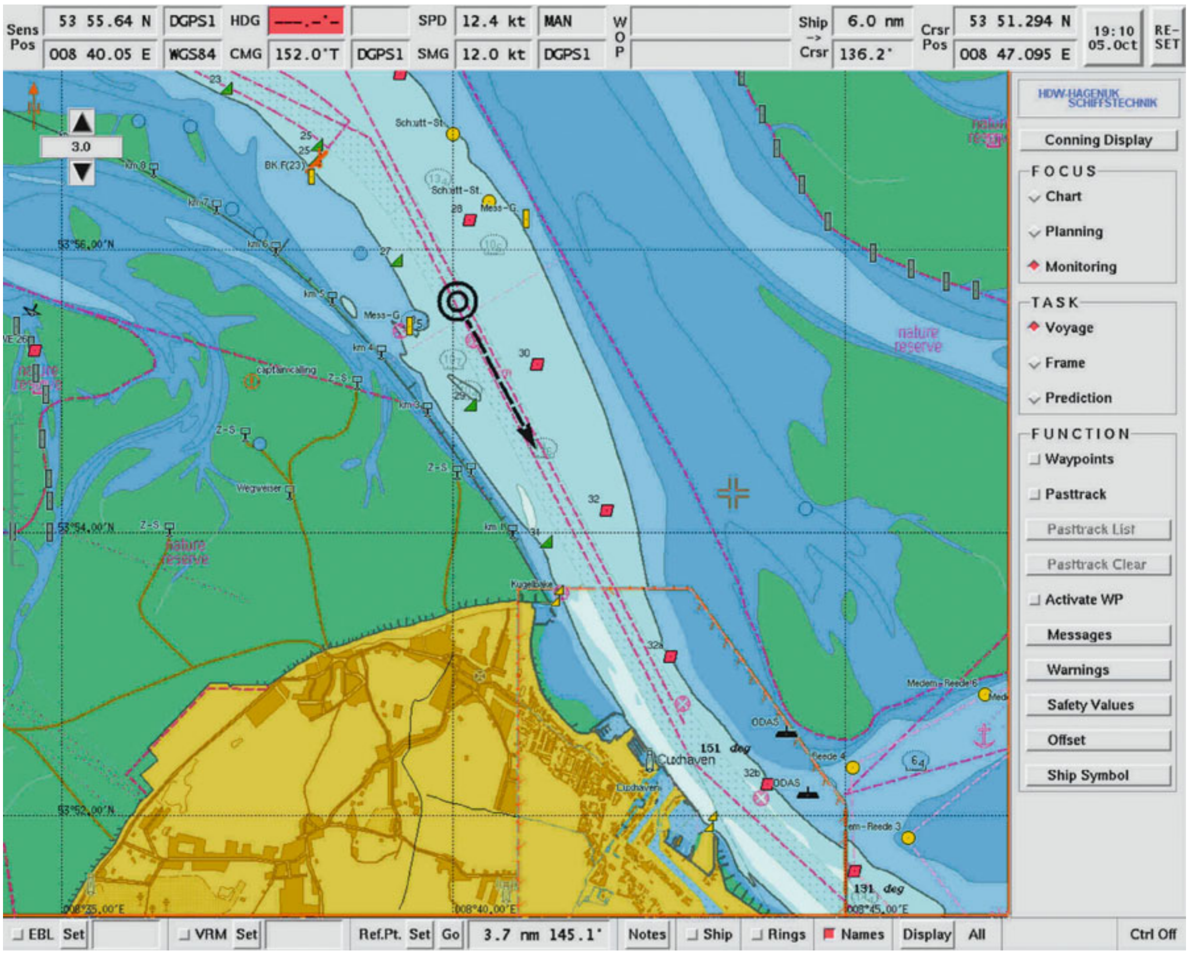
In the past, knowledge and information about the sea was storage in paper sea charts that provide seafarers a safer navigation. Nowadays, all that information has been collected in what is called Electronic Nautical Chart (ENC). ENCs contain extensive information on an area, providing indications of rocks and other obstructions.

Due to some complications to obtain an ENC, ENC are only treated theoretically. Main points presented in this chapter are shown below.

- Section 5.1 presents what is an ENC and why it is so important for a safer navigation.
- Section 5.2 briefly presents sensor fusion architecture technique.
- Section 5.3 describes how ENC information can be integrated to enhance situation awareness.

5.1 What is an ENC?

Navigating properly is mainly dependent on 3 main points, namely, knowing where is the position of the ship, what are the conditions in that position and in the vicinity and which direction should the ship follow to reach its destination. These points can be deal with a proper combination of the correctly ship's position with available information about the local environment. An electronic chart ensures the availability of both of these at sea.



Sens Pos 53 55.64 N 008 40.05 E DGPS1 WGS84 HDG ---.-° CMG 152.0°T DGPS1 SPD 12.4 kt SMG 12.0 kt MAN DGPS1

Ship -> Crsr 6.0 nm 136.2° Crsr Pos 53 51.294 N 008 47.095 E 19:10 05.Oct

EBL Set VRM Set Ref.Pt. Set Go 3.7 nm 145.1° Notes Ship Rings Names Display All Ctrl Off

Source: Vetter et al. [33] (try to find the corret one)

Figure 5.1: A typical human–machine interface (HMI) of an electronic navigational chart

The introduction of digital technology on-board vessels marked the beginning of electronic navigational charts development. As it was said before, digitalization of nautical charts are still being developed and some are very difficult to get access to them. Unlike land-base navigation technology, marine equipment undergoes strict supervision by national and international bodies. Reason of this strict supervision is that shipping is international in nature, consequently shipping nations of the world cooperated based on regulations agreed under the umbrella of the International Maritime Organization (IMO) and in the case of the ENC the International Hydrographic Organization (IHO) [33]. Electronic chart display and information system (ECDIS), as it is also known this technology, had become a mature technology and is about to send paper charts into sea navigation history.

The general functions of navigation are route planning, route monitoring, and route documentation. Electronic chart systems introduce a new level of performance compared with conventional paper nautical chart. Figure 5.1 shows an example of an ENC including the graphical chart, the main menu for control (right-hand side), an information line (status bar) to display numeric information (top), and a function line to activate common navigational functions (bottom).

Generally, the functions of an electronic chart are [33]: to determine the optimal route, with navigational and economical viewpoints taken into consideration and to ensure that the route can be safely sailed, e.g., by identifying navigational aids, marking position lines, and fixing the ship's position, course corrections, and speed.

This project treats mainly with identifying navigational aids. A navigational aid is any sort of marker which aids the traveller in navigation. Common types of such aids include lighthouses, buoys, fog signals, and day beacons, Appendix A. These navigational aids will be used to contextualized and correlate information from sensors.

5.2 Sensor Fusion

This section aims to give the reader an insight of the sensor fusion method. This part is only presented theoretically and without going into specifics due to the lack of time of the project.

Sensor data fusion techniques have been extensively employed on multisensor environments with the aim of fusing and aggregating data from different sensors and for obtaining a lower detection error probability and a higher reliability by using data from multiple distributed sources [6]. In our case those sensors are the cameras and the radar.

In the case of sensor data fusion for automotive applications, the task consists of creating and all-around detection system to overcome the deficiency of an individual sensing device. There are many works that have been carried out this problem through several approaches. In this project and to the lack of information $serial fusion$ technique has been employed.

Serial fusion is an architecture that uses the information from one sensor and then that information is complemented with the other one. An application of this method is shown in [30] and [21] for classifying pedestrians using a laser and a camera. First the laserscanner segments the scene and then provide some Region of Interest (RoIs), which are confirmed to match pedestrians by means of a vision based classifier.

The architecture presented is extracted from [30] and [21] and the following is a rewrite of their method but using a radar instead of a laser. This approach contains some modules subdivided in four systems: radar-based and vision-based systems, coordinate transformation and classification. The radar-based detects the objects in the radar space, estimates its position and size, and classifies them. The position of the objects is then converted to the camera coordinates in order to define a RoI in the image space. Then the vision-based system classifies the RoI received from the coordinate transformation module. Finally the classification module process the information from the others systems and outputs the class of the objects and its position (Figure 5.2).

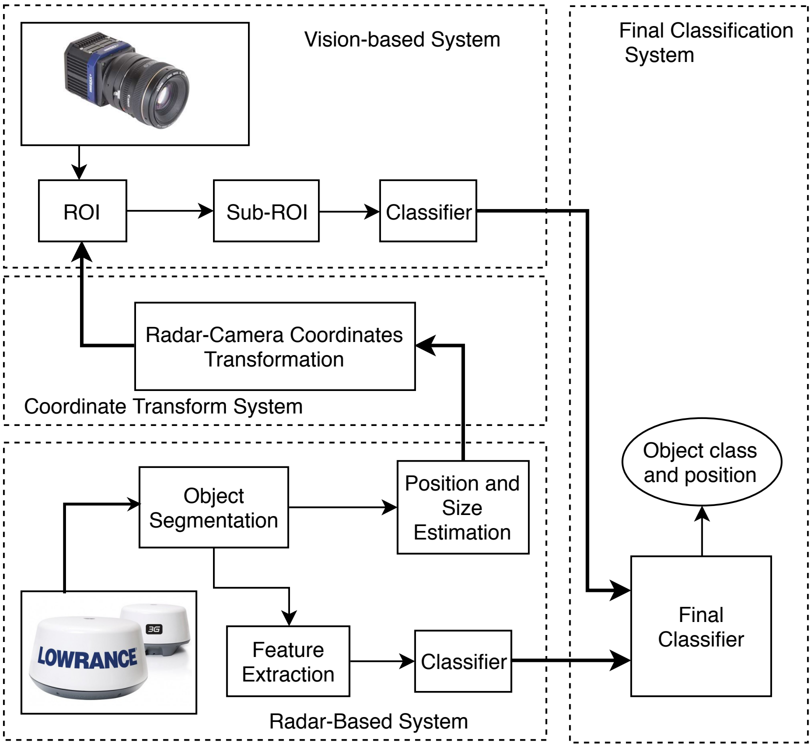

Vision-based System
ROI
Sub-ROI
Classifier
Final Classification System
Radar-Camera Coordinates Transformation
Coordinate Transform System
Object class and position
Object Segmentation
Position and Size Estimation
Final Classifier
Feature Extraction
Classifier
LOWRANCE
Radar-Based System

Figure 5.2: Module architecture using radar and vision information for detection and classification of objects

Since the information from radar is not available and description for object detection using cameras has been discussed in previous chapter, a brief description the coordinate transformation system and the final classification system is described.

Coordinate Transformation System

The task of fusing sensor information leads to establishing a correspondence between the measurements gathered by distinct sensors. In this case, it is necessary to find a correspondence between the camera and the radar. The coordinate transformation system shown in figure 5.2 calculates this correspondence. From the result of object's position and size estimation, in the radar space, is used to construct a ROI in the image frame by means of a set of coordinates transformations. This correspondence between objects detected in radar and the objects in image frame is used to create this ROI in the image plane. This process facilitates the process segmentation and detection in the vision-based system and accelerates the computational speed of the classifier.

Combining Classifiers
The final classifier shown in figure 5.2 takes as an input the results from the classifiers of the camera and the radar, which are then combined by means of another classifier in order to produce an unique decision rule. This final classifier works with a vector of structured data (feature array) as it is described in the next section.

The way this classifier works is by using a sum rule to combine (fuse) the classifiers outputs from radar and camera. This decision rule inspired in [17] uses the Bayesian framework to ultimate classify an object based on the outputs of each classifier.

Let us consider the number of classifiers as NC, and the feature vector used by the ith classifier by Ω_i. Let us assume that each class q_i is represented by a class-conditional probability density function $p(\Omega_i \mid q_i)$ and its as priority probability of detection $P(q_i)$. Given the probability density function and the priori probability, the classical decision rule can be stated as:

$$\begin{aligned}&assign\ Object \rightarrow if\\ &P(q_j \mid \Omega_1, ..., \Omega_{NC}) = max_k P(q_k \mid \Omega_1, ..., \Omega_{NC})\end{aligned} \tag{5.1}$$

Assuming that the features vectors are conditionally statically independent, and that the posterior probability of each classifier do not deviate dramatically from the prior probability, after some mathematical formulations [17], a ”practical” combinational Bayesian decision rule is stated as:

$$\begin{aligned}&assign\ Object \rightarrow if\\ &(1 - NC)P(q_j) + \textstyle\sum_{i=1}^{NC} P(q_j \mid \Omega_i) =\\ &max_{k=1}^{N} = [(1 - NC)P(q_k) + \textstyle\sum_{i=1}^{NC} P(q_k \mid \Omega_i)]\end{aligned} \tag{5.2}$$

This ”sum” decision rule depends on the prior probability of occurrence each class q_i and the posterior probabilities yielded by the respective classifiers.

5.3 Contextualizing ENC information with sensor measurements

This section describes how ENC information can be used to enhance object estimation. To elaborate, a description of the methodology used is presented. As mentioned before, this section is treated theoretically due to the difficulties to obtain an ENC. Therefore, assumptions about the format on how information is presented in ENC are made based on knowledge provide by maritime experts. First, an introduction to the hypothesis (assumptions) of how information from the navigational aids is display on ENC is described. Then the proposed Kalman Filter model for enhance object estimation is presented.

5.3.1 Assumptions

Navigational aid position estimation such as beacons, buoys or lighthouses is considered as the target of this section. The way as these sea marks are represented in a map is by a type of map projection called Mercator. Mercator projection is widely used in maritime navigation an it consists of using mathematical formulas to relate spherical coordinates on the globe to planar coordinates (Figure 5.3). In the spherical system, horizontal lines, are lines of equal latitude,whereas vertical lines,are lines of equal longitude. Latitude and longitude values are traditionally measured either in decimal degrees or in degrees, minutes, and seconds (DMS). Therefore, to refer the location of a navigational aid, longitude and latitude are required.

Assumptions of the format of the sea marks in an ENC are:

- Longitude and attitude are given as planar coordinates.
- Accuracy of these values are provided so an estimation of the variance of these can be made.
- These objects are considered statics which means that position is independent of the time. In the case of buoys, tide changes can alter the position of them. This fact, is translate as a decrease of the accuracy of the buoys position.

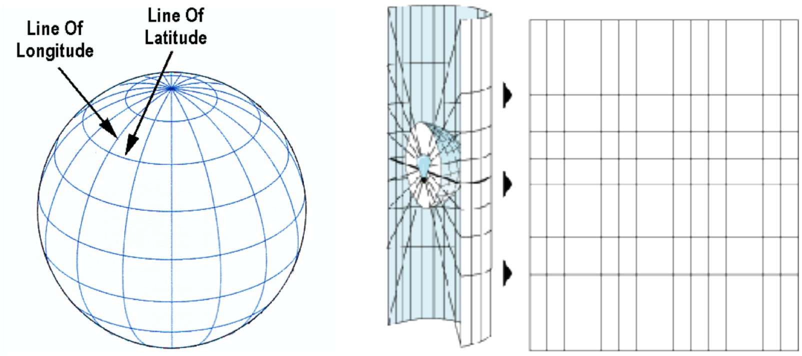


Figure 5.3: Illustration of coordinate systems. (Left) Spherical system representation. (Right) Map projection; from spherical system to planar coordinate

5.3.2 Kalman Filter model for object detection

This section aims to present a model that allows enhancement in object detection using values from the ENC as well as from sensors. To elaborate, Kalman Filter is used. Figure 5.4 shows the general pipeline followed in the method that now is presented. First, sensors, ie, radar and cameras compute from raw data to features an estimation of the objects. Then both estimations are fused, so a lower error probability. The fused estimation is complemented with ENC information by using a Kalman filter.

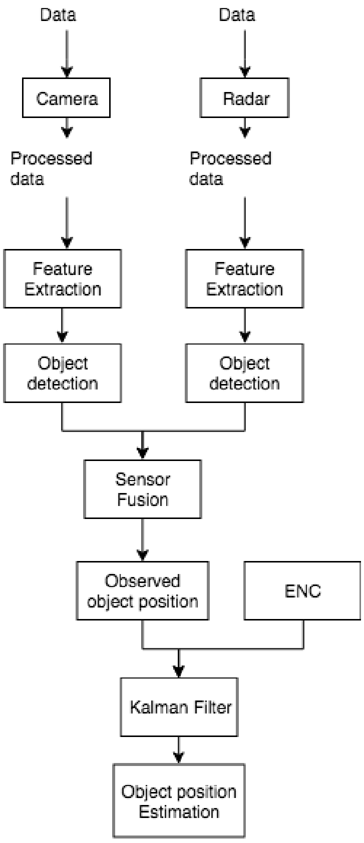


Figure 5.4: General Pipeline for object detection correlation with the ENC

First a brief introduction to the model and notation used in the method are presented. Then, the description of the method is introduced.

The general notation and derivation presented below is inspired by Thrun's book "Probabilistic Robotics" [31]. Despite the fact this book is mainly for land-robot applications it can be a good aid to give an idea how the problem could be focused.

Introduction

In general, sensors models are based on extracting features from the measurements. If we can denote the feature extractor as a function of f, then the features extracted from a range measurement at time t are given by $f(z_t)$. One of the main advantages of this technique is the enormous reduction of time complexity. Example of this features varies depending on the sensors; for range sensors, it is commonly to find lines or corners whereas for cameras, relies on computer vision techniques to extract an specific feature as edges, patterns or objects with distinct appearance.

The most common model for processing sea marks assumes that the sensor can measure the range and the bearing of the sea mark relative to the ship's local coordinate frame. Moreover the feature extractor might generate a signature. Signature might be either numerical value, an integer that characterizes type or a multi-dimensional vector characterizing a landmark. All this information can be represented as below:

$$f(z_t) = \{f_t^1, f_t^2, f_t^3 ...\} = \{ \begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^2 \\ s_t^2 \end{pmatrix}, \begin{pmatrix} r_t^3 \\ \phi_t^3 \\ s_t^3 \end{pmatrix} ...\} \tag{5.3}$$

where r, ϕ, s are the range, bearing and signature, respectively.

The number of features identified at each time step is variable. However, many probabilistic robotic algorithms assume conditional independence between features, that is,

$$p(f(z_t)|x_t, m) \quad = \prod \quad p(r_t^i, \phi_t^i, s_t^i|x_t, m) \tag{5.4}$$

In order to use this sensor model, we need a variable that establishes correspondence between the feature f_t^i and the sea mark m_j in the map. This variable called correspondence variable c_t^i is the true identity of an observed feature. It is established that $c_t^i \in \{1, ..., N+1\}$ where N is the number of sea marks in the map m. If $c_t^i = j \leq N$ then the ith feature corresponds to the jth mark. When $c_t^i = j \leq N+1$ a feature observation does not correspond to any feature in the map m.

In practice, is rarely the case when correspondences can be determined with absolute certainty. Most implementations therefore determine the identity of the sea mark during localization. One of the strategies to cope with this problem is known as *maximum likelihood* correspondence. This technique will be treated in detail below.

Once, the object detection and the correspondence is presented is time to describe the predictor of the sea mark. The Kalman filter is a technique for filtering and prediction in linear systems. It represents beliefs by the moments representation: the belief is represented by the mean μ and the covariance Σ. Algorithm 1 depicts the Kalman Filter algorithm. At time t, the belief is represented by the mean μ_t and the covariance Σ_t. The input of the Kalman filter is the belief at time t_1, represented by μ_{t-1} and Σ_{t-1}. These parameters are updated by using control u_t and the measurement z_t. The output is the belief at time t, represented by μ_t and Σ_t.

In lines 2 and 3, the predicted belief $\bar{\mu}$ and $\bar{\Sigma}$ is calculated representing the belief $\bar{bel}(x_t)$ on time step later, but before incorporating the measurement z_t. This belief is obtained by incorporating the control u_t. The mean is updated using the deterministic version of the state transition function 5.5, with the mean μ_{t-1} substituted for the state x_{t-1},

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \tag{5.5}$$

Algorithm 1 Kalman filter algorithm

```
1: input $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$
2: $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
3: $\bar{\Sigma}_t = A_t \bar{\Sigma}_{t-1} A_t^T + R_t$
4: $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$
6: $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
7: return $\mu_t, \Sigma_t$
```

where x_t and x_{t-1} are the state vectors, and u_t is the control vector time at t, A_t and B_t are state and control matrices that assure linearity between states. ϵ_t is a Gaussian random vector that models the randomness in the state transition. In Line 3 covariance is updated considering the fact that states depend on previous states through the linear matrix A_t.

The belief $\bar{bel}(x_t)$ is transformed into the desired belief $bel(x_t)$ in Lines 4 through 6, by incorporating the measurement z_t. The variable K_t, Kalman gain, specifies the degree to which the measurement is incorporated into the new estimate. Line 5 manipulates the mean, by adjusting it in proportion to the Kalman gain K_t and the deviation of the actual measurement, z_t, and the measurement predicted, $\bar{z}_t$. Finally, the new covariance is calculated in Line 6. This algorithm will be adapted to our problem and then incorporated to the method present below.

Method

Problem description

Featured based maps consists of a list of features ($l = m_1, m_2 ...$) where each one posses a signature and a location coordinate. The resulting measurement model is formulated for the case where a feature at time t corresponds to a sea mark in the map. As usual, the ship pose is given by $x_t = (x, y, \theta)^T$ where x and y determines the latitude and longitude and θ the yaw.

Figure 5.5 shows the framework of the ship. There are two coordinate systems; one of the coordinate framework of the global map, O^G, and the other relative to the ship location, O^{ship}. In this case, the sea mark m and the ship are represented by global coordinates (x_m^G, y_m^G) and $(x_s^G, y_s^G, \theta_s^G)$, respectively. Sensors that provide information of the location of sea marks brings that position in terms of the range, bearing and a signature as shown in equation 5.3. Range and bearing $(r_{m,sensor}^{ship}, \phi_{m,sensor}^{ship})$ need to be transformed from local to global coordinates $(x_{m,sensor}^{ship}, y_{m,sensor}^{ship})$. This is achieved by the equation showed below,

$$\begin{aligned} x_{m,sensor}^G &= r_{m,sensor}^{ship} \cdot cos(\phi_{m,sensor}^{ship} + \theta_s^G) + x_s^G \\ y_{m,sensor}^G &= r_{m,sensor}^{ship} \cdot sin(\phi_{m,sensor}^{ship} + \theta_s^G) + y_s^G \end{aligned} \tag{5.6}$$

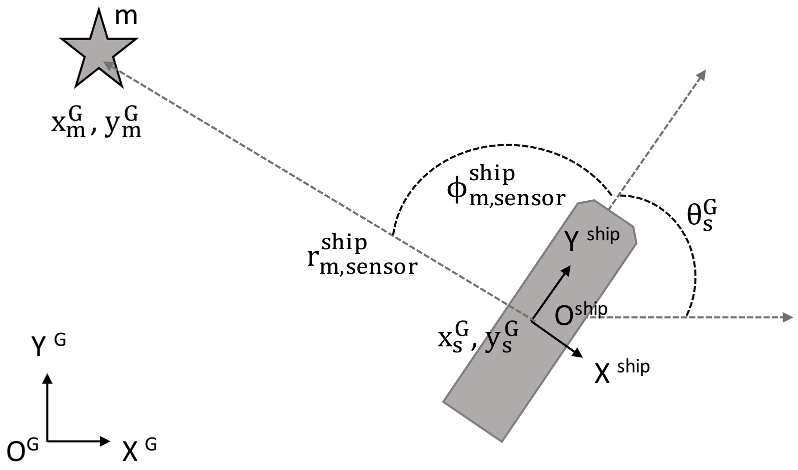


Figure 5.5: Illustration of a coordinate system of the global map

Now that Kalman filter algorithm is given, description of the model used for localization of obstacles using sensor fusion and ENC information is presented. Figure 5.6 shows a scheme for estimate the position of the sea marks. In this case, sea mark k is initialized from the ENC information. Then when the ship senses, fused the information from sensors and estimates the correspondence. Correspondence, as explained below gives a value of how good a measurement match with an object in the map. Therefore if the sea mark m corresponds to the measurement j, Kalman filter is implemented and new position of the sea mark updated.

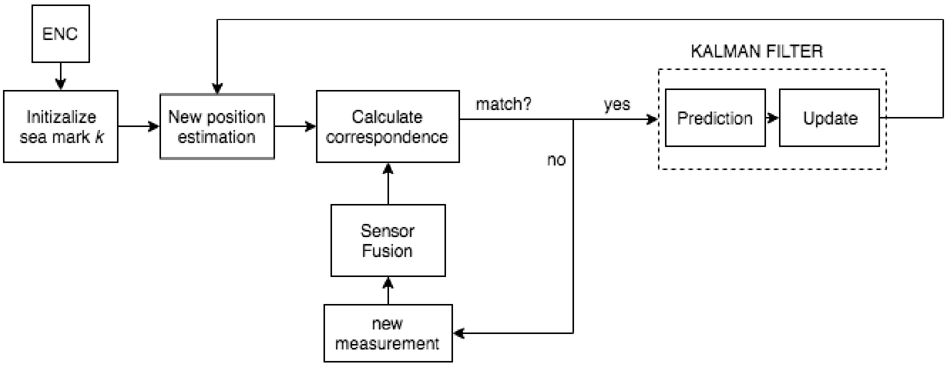


Figure 5.6: Block diagram illustration of the algorithm proposed for estimate sea mark position

As mentioned before, the aim of this section is estimation of sea marks. This is equal to obtain the belief $bel(x_t)$, where x_t is the state that represents the position and signature of the sea mark, i.e, latitude (x_m^G), longitude (y_m^G) and type of sea mark ($type$),

$$\begin{pmatrix} x_m^G \\ y_m^G \\ type \end{pmatrix} = \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{pmatrix} = x_t \qquad \begin{bmatrix} \sigma_{x,m}^2 & 0 & 0 \\ 0 & \sigma_{y,m}^2 & 0 \\ 0 & 0 & \sigma_{s,m}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{1,t}^2 & 0 & 0 \\ 0 & \sigma_{2,t}^2 & 0 \\ 0 & 0 & \sigma_{3,t}^2 \end{bmatrix} = \Sigma_t, \quad (5.7)$$

where Σ_t represents the covariance matrix of the state x_t and $\sigma_{1,t}^2$, $\sigma_{2,t}^2$ and $\sigma_{3,t}^2$ are the variances of the latitude, longitude and the sea mark type of the mark m at time t, respectively.

As before, each feature vector (measurement) contains three elements, latitude, longitude and signature,

$$z_t = \begin{bmatrix} x_{m,sensor}^G & y_{m,sensor}^G & s_{m,sensor} \end{bmatrix}^T \quad (5.8)$$

The measurement probability $p(z_t \mid x_t)$ model is given by the following expression,

$$\bar{z}_t = C_t \bar{x}_t + \mathcal{N}(0, Q_t) \quad (5.9)$$

where C_t is a matrix of size 3×3 and $\mathcal{N}(0, Q_t)$ is Gaussian noise and $\bar{x}_t$ is given from the prediction state from equation 5.13. In equation 5.15 are the given the matrices used in the system.

Initialization

First, the algorithm initializes first belief $bel(x0)$ from the information provided form the ENC.

$$\begin{bmatrix} x_{m,ENC} \\ y_{m,ENC} \\ s_{m,ENC} \end{bmatrix} = \begin{bmatrix} x_{1,0} \\ y_{1,0} \\ s_{1,0} \end{bmatrix} = x_0 \qquad \begin{bmatrix} \sigma_{x,m,ENC}^2 & 0 & 0 \\ 0 & \sigma_{y,m,ENC}^2 & 0 \\ 0 & 0 & \sigma_{s,m,ENC}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{1,0}^2 & 0 & 0 \\ 0 & \sigma_{2,0}^2 & 0 \\ 0 & 0 & \sigma_{3,0}^2 \end{bmatrix} = \Sigma_0 \quad (5.10)$$

where $x_{m,ENC}$, $y_{m,ENC}$ and $s_{m,ENC}$ are the position and signature of the sea mark from the ENC and Σ_0 is the initial covariance matrix given from the variances of the ENC.

Correspondence estimation

The algorithm proposed for estimating correspondences is performed via a maximum likelihood estimator. The maximum likelihood estimator determines the correspondence that maximizes the data likelihood. Then a correspondence variable is choosen, by minimazing a quadratic Mahalanobis distance function defined over the measure feature vector z_t^i and expected measure $\hat{z}_t$ for the sea mark m_k in the map.

Maximum likelihood correspondence, first determines the most likely value of the correspondence variable and then takes this value for granted. This is equal to solve equation,

$$\bar{c}_t = \underset{c_t}{\operatorname{argmax}} \;\; p(z_t \mid c_{1:t}, l, z_{1:t-1}). \tag{5.11}$$

Here c_t is the correspondence vector at time t, $z_t = \{z_t^1, z_t^2, ...\}$ is the measurement vector that contains the list of features, or sea marks observed at time t. Equation 5.11 after some mathematical derivation [31] yields:

$$\bar{c}_t(i) = \underset{c_t}{\operatorname{argmax}} \;\; det(2\pi S_t)^{\frac{1}{2}} exp\{\frac{-1}{2}(z_t^i - \bar{z}_t)^T [S_t]^{-1}(z_t^i - \bar{z}_t)\} \tag{5.12}$$

which selects the maximum correspondence $\bar{c}_t$ that maximizes the likelihood of all measurements z_t. Here, S_t is the uncertainty corresponding to the measurement $\bar{z}_t$ and the Jacobian H_t of the measurement model.

Kalman filter

Once the correspondence is made the Kalman filter is implemented. It is divided into two parts; prediction and update. First, it takes the last state to predict the next pose for the sea mark. This value is then used to update the posterior states and covariance matrix. Finally the output is the belief represented by x_t ans Σ_t. In this case, prediction is given by this expression,

$$\bar{x}_t = A_t x_{t-1} + B_t u_{t-1} \qquad \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T. \tag{5.13}$$

The equations for the measurement update yields,

$$\begin{aligned} K_t &= \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \\ x_t &= \bar{x}_t + K_t(z_t - \bar{z}_t) \\ \Sigma_t &= (I - K_t C_t)\bar{\Sigma}_t \end{aligned} \tag{5.14}$$

$$A_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad C_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Q_t = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{bmatrix} \tag{5.15}$$

One may notice, that the elements in the input matrix B_t are zero due to the fact that there is not control in the system.

The algorithm proposed is shown in Algorithm 2. First the algorithm initialize the sea mark k with the information from the ENC of that sea mark (Line 1). Once the position is initialized, the algorithm waits until the sensors match that sea mark by calculating the maximum correspondence along all the measurements (Lines 2 to 5). Then the measurement that has been match is used to update the last prediction.To elaborate, first prediction is computed (Line 6 to 10) and then last prediction ($\bar{z}_t$) is computed as explained in equation 5.12. Finally, this measurement is used to update last position estimation of that sea mark (Lines 11 to 14).

Algorithm 2 Contextualizing ENC information with sensor measurements

```
initialization $x_0, \Sigma_0, \bar{z}_0$
input $x_{t-1}, \Sigma_{t-1}, z_t^i$
for all observed features $z_t^i$ do
$\bar{j}(i) = \operatorname{argmax}_{c_t} \;\; det(2\pi S_t)^{\frac{1}{2}} exp\{\frac{-1}{2}(z_t^i - \bar{z}_t)^T [S_t]^{-1} (z_t^i - \bar{z}_t)\}$
endfor
Prediction
$\bar{x}_t = A_t x_{t-1}$
$\bar{\Sigma}_t = A_t \bar{\Sigma}_{t-1} A_t^T + R_t$
$\bar{z}_t = C_t \bar{x}_t + \mathcal{N}(0, Q_t)$
$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
Update
$x_t = \bar{x}_t + K_t(z_t - \bar{z}_t)$
$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
return $x_t, \Sigma_t$
```

5.4 Summary

In this chapter a theoretical description of the implementation of the model for object detection is presented. First an introduction of the different aids for navigation available in an ENC are presented. Sea marks (aid for navigation) are useful for seafarer due to the fact that depending on the feature (color, shape...) of that sea mark it gives a different information. Then the sensor fusion architecture is presented. This technique first takes the information from radar and created some ROI to classify the information from camera. Both classifiers (radar and camera) are then combined in a final one that provides the information of that object detected. Finally, an approach to have a better estimation of sea marks using sensor fusion and the ENC is presented. The approach consists of a Kalman filter implemented for each of the sea marks.

CHAPTER 6

Test of object searching

This chapter presents the results and discussion for the experiment carried out with the setup that will be used in the real implementation. The experiment aims to validate results obtained with the previous datasets and how methods performed in a system that could be used in cooperation with the rest of the sensors (radar and IMU).

- Section 6.1 describes the system setup and where the images were taken.
- Section 6.2 presents the results and discussion for the experiment carried. out with the real setup.

6.1 Object system setup

The design of the system for measuring was carried out by DTU. However, it is interesting to present how the system looks like to make easier the reader the understanding of the setup. Figure 6.1 shows the construction for performing the experiment. It is composed of a radar (R), a set of cameras (C1L, C1R, C2, C3), an IMU with the GPS integrated and a computer that collects the data. The important setup regarding this project was the set of cameras for collecting images, since the IMU and radar was not used in this project. Only a monochrome camera (C2) and two color cameras (C1L and C1R) were used for this test. FLIR camera (C3) was not used due to some problems with the battery.

The experiment was carried out in Helsingør in daylight. The information of the dataset content is shown in table 6.1. Color in the table concern to the captured images from the two color cameras and Monochrome refers to the images from the monochrome camera. Figure 6.2 shows a collection of images taken from camera 1 and 2 in the

	number of images	number of objects	
		ferry	cargo ships
Color	866	1006	105
Monochrome	134	134	0

Table 6.1: Dataset content from Helsingør experiment .

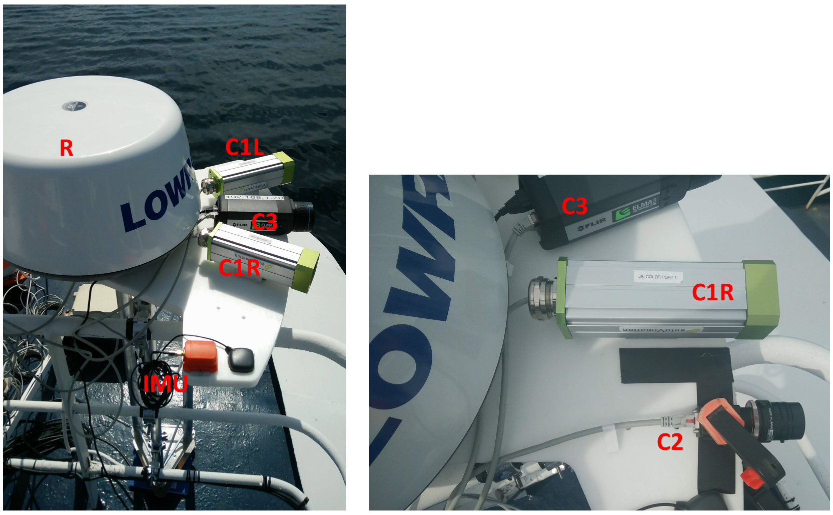


Figure 6.1: Platform used for measurement. In red are annotated different sensors. Radar (R), Inertial Measurement Unit and GPS (IMU), two color cameras (C1L, C1R), one monochrome camera (C2) and FLIR camera (C3).

Helsingør expedition. These images were captured in different environments (harbour, open sea).

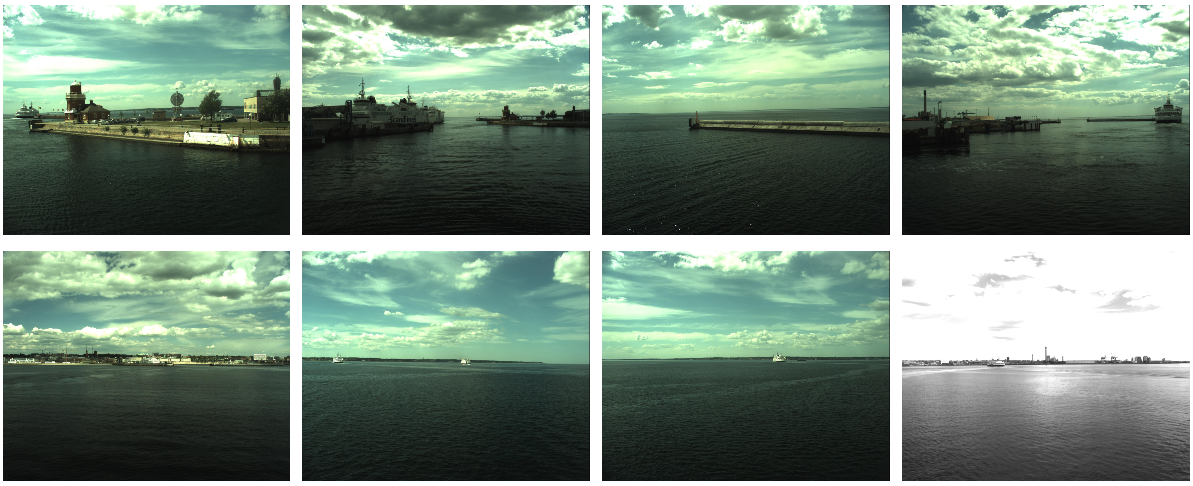


Figure 6.2: Set of images from from Helsingor experiment.

6.2 Results

Results for the experiment are shown in this section. As it was presented in the test experiments, first the horizon line algorithm is evaluated and then the object detection.

Horizon detection evaluation

Table 6.2 gives the results for the precision and recall for the experiment in Helsingør. For 1000 images, the average precision and recall are 98,84% and 96,27%, respectively. The proposed algorithm achieves a 100% precision and recall averaged over 134 monochrome images. Such a high scores indicate the effectiveness of the proposed horizon detection method.

	Number of images	TP	TN	FP	FN	Precision	Recall	Time(s)
Color	866	756	32	18	61	97,67%	92,53%	0,1076
Monochrome	134	134	0	0	0	100,00%	100,00%	0,0982
Total	1000	890	32	18	61	98,84%	96,27%	0,1029

Table 6.2: Precision and recall rates by the proposed horizon line detection in Helsingør experiment

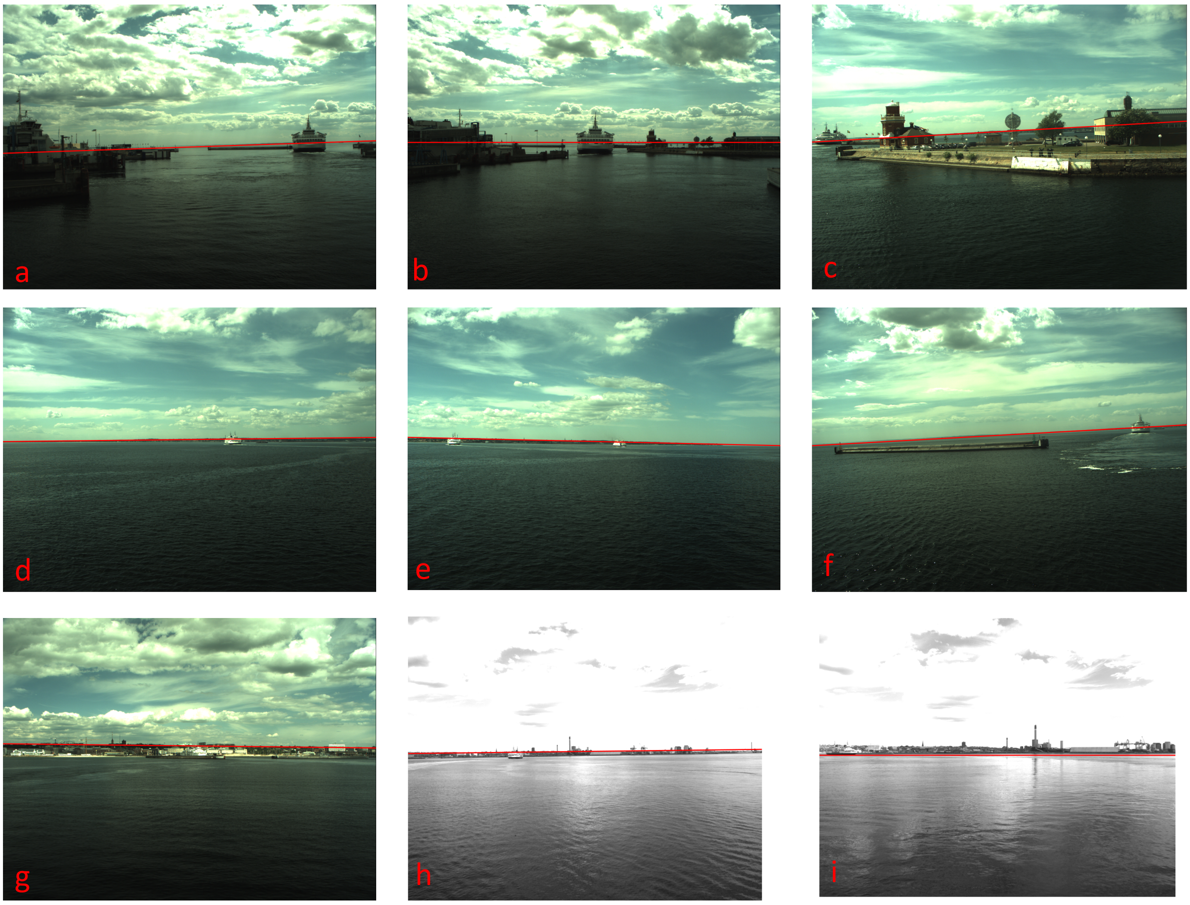


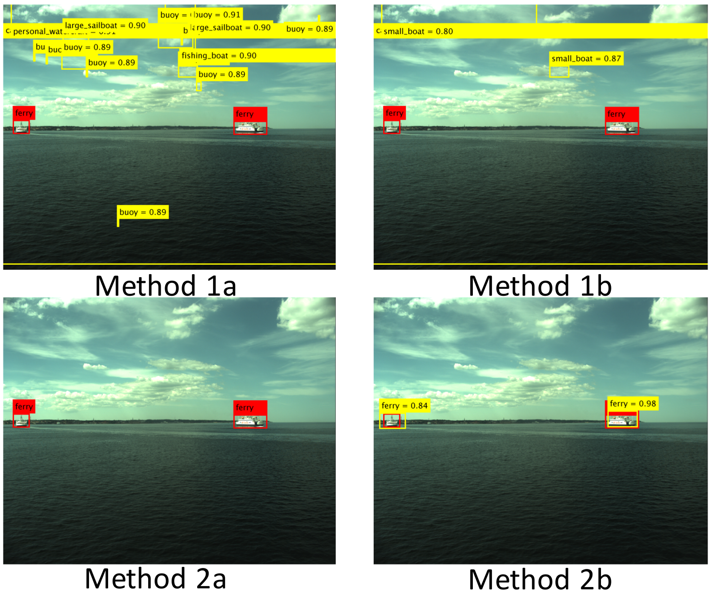
Figure 6.3: Horizon detection results in Helsingør using the proposed algorithm.

Further, this algorithm shows a good performance in both scenarios: open seas and close to port/harbour. Figure 6.3 depicts the horizon line marked in red color in a set of images. The line detected in figures 6.3(a, b, c, g) confirms the adaptability to scenarios were the horizon line is not well defined. Further, images captured in open sea also presented a good performance for both color (figures 6.3(d, e, f)) and monochrome (figures 6.3(h, i)) cameras. In the case of figures 6.3 (g, h, i) where region between sea and sky includes buildings and land, horizon detections appear above the land (figure 6.3 (g, h)) or below the land (figure 6.3 (i)). Both estimations are considered correct since the aim of this algorithm is to find a line that segments sky from sea, and in scenarios like those, that problem is can be solved in the following background modeling step.

Object detection and classification evaluation

Table 6.3 lists the average precision and recall scores for color and monochrome cameras. The Method 2b (Faster RCNN) outperforms the other two methods in terms of both precision and recall rates. This method achieves a 88,52% and 40,30% in precision and recall rates in monochrome pictures. In comparison, this method results the most effective among two other methods with a 35,6% of ships detected in over 1000 images with a total of 1245 object.

Methods 1 and 2a performs almost similar, neither providing adequate precision and recall. In Method 1 we note that the presence of clouds contributes the presence of false detections (figures 6.4 and 6.5). Further, the images with the monochrome camera were captured in a harbour scenario where the presence of harbour made difficult the object detection.



Method 1a

Method 1b

Method 2a

Method 2b

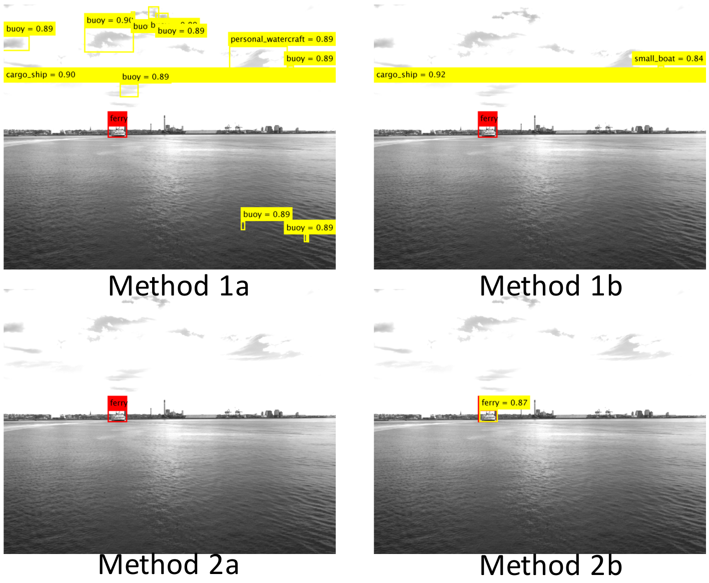
Figure 6.4: Object detection for Helsingør using color camera

Classify algorithms (Method 1a and 1b) were difficult to evaluate in terms of precision and recall due to the fact that classification was mainly based on the RoI and the poor

Method	Color		Monochrome		Average	
	Precision	Recall	Precision	Recall	Precision	Recall
Method 1 (RoI finder)	**0,30%**	**0,30%**	0,00%	0,00%	**0,26%**	**0,26%**
Method 2a	5,40%	0,18%	**6,00%**	**1,50%**	**5,48%**	**0,36%**
Method 2b	79,21%	30,87%	**88,52%**	**40,30%**	**80,46%**	**32,13%**

Table 6.3: Precision and recall rates in object detection for Helsingør experiment.

results on the RoI finder made difficult this task.



Method 1a
Method 1b
Method 2a
Method 2b

Figure 6.5: Object detection for Helsingør using monochrome camera

6.3 Summary

In order to validate results obtained with the previous datasets and also study how the proposed methods perform with the camera system setup this chapter presents the results of the experiment carried out in Helsingør with the real system. To elaborate, a color and a monochrome camera where used to collect a total of 1000 images with 1245 objects including (ferries and cargo ships).

Results for method 1 involves the evaluation of both horizon line detection and the RoI finder. Horizon line detection achieves a 98,84% and 96,27% averaged precision and recall scores, respectively. These results confirm the effectiveness of this horizon detection method that is crucial in the following background modeling subtraction.

Object detection assessment shows that the method 2b (Fast RCNN) outperforms among 3 other proposed methods with a 80,46% and 32,13% of precision and recall rates, respectively. The other two methods (method 1 and method 2a) performs almost similar, neither providing adequate precision and recall.

CHAPTER 7

Conclusion

This chapter concludes the thesis, summarizing most important findings and presenting some future challenges.

- Section 7.1 gives a brief overview of the work presented in this thesis.
- Section 7.2 underlines the most important findings in this thesis.
- Section 7.3 presents some future work that might be worth investigating.

7.1 Overview

This thesis has investigated four different approaches for object detection an classification at sea using computer vision. This work was carried out to provide robust detection with optical sensor (camera) with the aim of aiding maritime collision avoidance. The detector uses 4 different approaches; three of them are conventional approaches whereas the other uses the state of the art object detection classification framework Faster R-CNN.

Two of the conventional approaches consist of a Region of Interest finder. This multiscale algorithm includes an horizon line detector, a background subtraction process and finally a foreground segmentation. First, a projection horizon detection method is proposed. Second, three different methods for background modelling are presented and discussed. One of the methods consists of selecting the region of the background manually to proceed with the subtraction. The other two techniques include a Gaussian Mixture Modeling; one applied to the value of the pixel whereas the other detects objects by extracting texture features from DCT blocks. The next step is to apply a set of morphological operations so the small object can be erased and the objects can be detected properly. Once the object has been detected two classifiers categorize the object according to the different trained classes. The classifiers include a SVM and a CNN classifier.

Other conventional approach (ACF detector) consists of computing several features for each input image and the used classifiers that yields in an output, namely, whether the image belong to that category. Finally, Faster R-CNN is composed of a modified CNN where the RoI's and classifiers are defined and computed in different layers of the CNN.

Further, this work includes a brief introduction of how sensor fusion can be used to enhance situation awareness using information from different sensors. In addition, some of the important aspects of the ENC are presented so information from sensor fusion can be correlated with an electronic sea chart.

7.2 Findings

The results includes first test with images from different datasets and finally with images captured with the cameras that will be used in the real implementation. In both experiments two elements are evaluated, i.e, horizon detection and object detection.

Horizon detection is evaluated in the RoI finder algorithm. The horizon detection method can extract the sea regions accurately for complex background modeling. It achieves a 98,84% and 96,27% of precision and recall in the final experiment which confirms its effectiveness to segment sky from sea. Background subtraction using two of the techniques presented did not have a good performance due to the complexity of the background and the sensitivity in tuning parameters of the background modelling. As a result, the best result for this technique only achieves a 3% and 20% of precision and recall. The main contribution of this approach for detecting a RoI is the horizon detection algorithm proposed and the different proposals for background subtraction.

On the other hand, results show that the Faster R-CNN outperforms among the other approaches due to the intensive training with a pre-trained CNN that aims to detect objects. This techniques achieves a 80,46% and 32,13% of precision and recall in the final experiment confirming the effectiveness of this method among the three other. ACF detector performs similar to the RoI finder algorithm, providing an inadequate precision and recall rates.

7.3 Future work

The presentation of 4 different approaches gives an idea of which one is better for an application in a maritime environment. Regarding the RoI finder, the different background subtraction techniques can be enhanced by studying the performance with different parameters. Future work should also include improvement the recognition ability to discriminate ships from other foreground objects such as buildings, harbor...

Concerning the findings in this thesis there are aspects which should be explored further. Building a larger and more diverse training and test data set is one thing that is necessary in order to evaluate the true robustness of such a detectors. Collecting image data on the same path at different times of year with different targets would be ideal. In the case of evaluating robustness, not enough data can be obtained. Another possibility that should be explored is adding some background-classes to the training-data. Adding classes such as house, harbor and mountain to name a few, could help eliminate some of the false positive detection.

In chapter 5 a theoretical introduction of how sensor fusion and ENC can be fused is presented so the situation awareness of the vessel can be improved. In the tracking pipeline it is also necessary to estimate the Cartesian coordinates of the detected vessels coordinates by using sensor fusion and track them.

Appendices

APPENDIX A
Aids to Navigation

Classification of different marks extracted from [22]:

- lateral marks
- cardinal marks
- isolated danger marks
- safe water marks
- special marks
- marking new dangers
- other marks

A.1 Lateral Marks

The conventional direction of buoyage, which must be indicated in appropriate nautical charts and documents, may be either:

- The general direction taken by the mariner when approaching a harbour, river, estuary or other waterway fr5om seaward

- The direction determined by the proper authority in consultation, where appropriate, with neighbouring contruis. In principle, it should follow a clockwise direction around and masses.

There are two international Buoyage Regions A and B, where lateral marks differ. The current geographical division os these two Regions are shown on the world map.

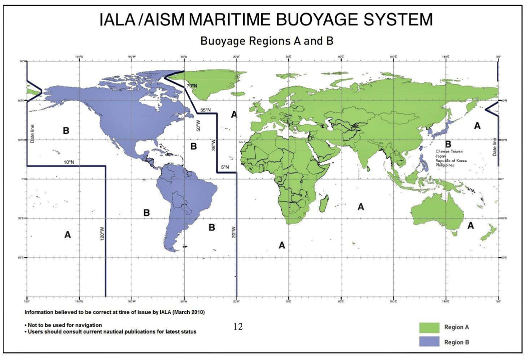
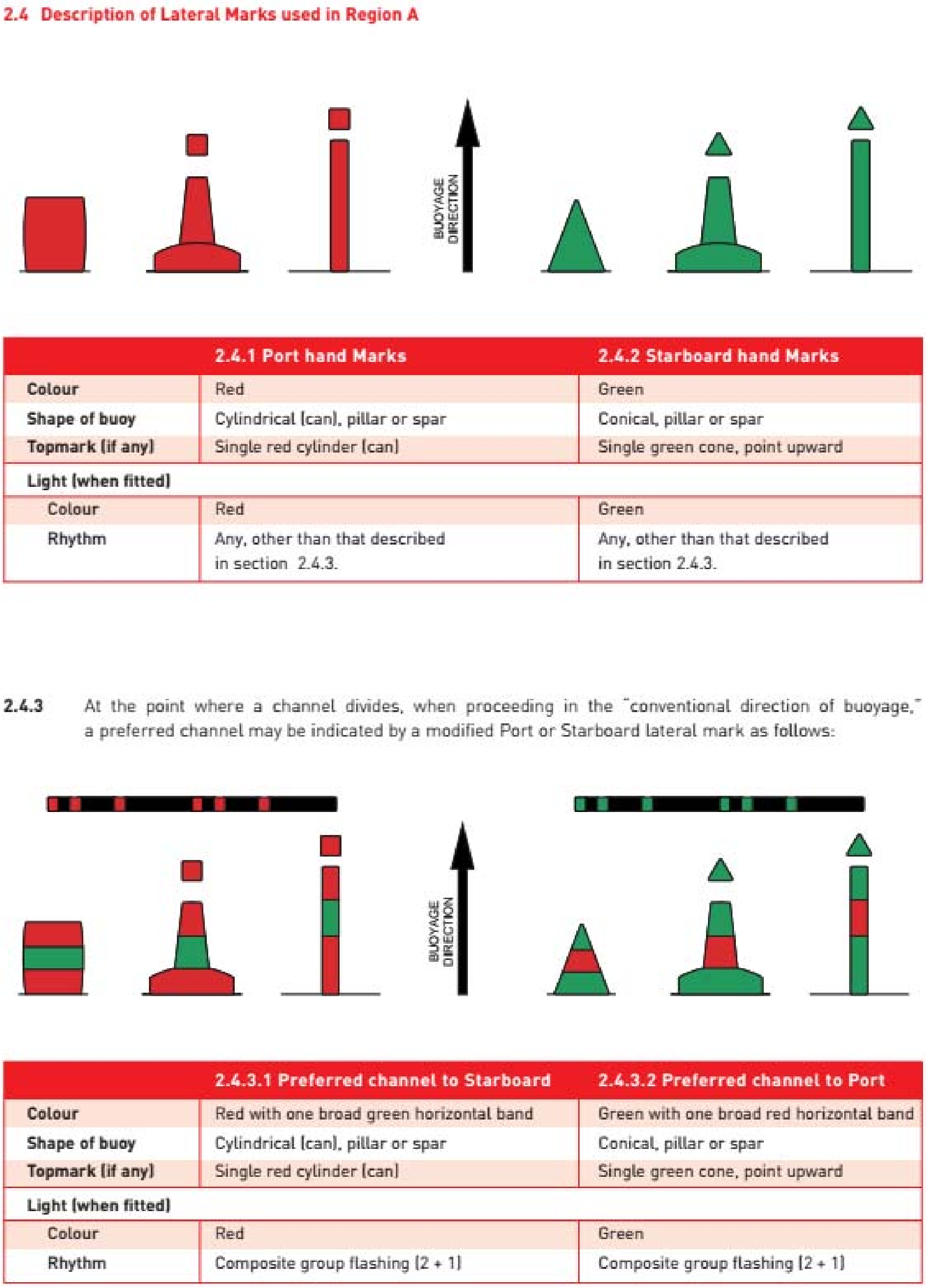

IALA / AISM MARITIME BUOYAGE SYSTEM
Buoyage Regions A and B
70°N
55°N
50°W
35°W
5°N
10°N
120°W
20°W
Date line
A
B
Chinese Taiwan
Japan
Republic of Korea
Philippines
Information believed to be correct at time of issue by IALA (March 2010)
• Not to be used for navigation
• Users should consult current nautical publications for latest status
12
Region A
Region B

Figure A.1: World map with Buoyage Systems A and B

Description of lateral Marks used in Region A are shown below:



2.4 Description of Lateral Marks used in Region A

	2.4.1 Port hand Marks	2.4.2 Starboard hand Marks
Colour	Red	Green
Shape of buoy	Cylindrical (can), pillar or spar	Conical, pillar or spar
Topmark (if any)	Single red cylinder (can)	Single green cone, point upward
Light (when fitted)		
Colour	Red	Green
Rhythm	Any, other than that described in section 2.4.3.	Any, other than that described in section 2.4.3.

2.4.3 At the point where a channel divides, when proceeding in the "conventional direction of buoyage," a preferred channel may be indicated by a modified Port or Starboard lateral mark as follows:

	2.4.3.1 Preferred channel to Starboard	2.4.3.2 Preferred channel to Port
Colour	Red with one broad green horizontal band	Green with one broad red horizontal band
Shape of buoy	Cylindrical (can), pillar or spar	Conical, pillar or spar
Topmark (if any)	Single red cylinder (can)	Single green cone, point upward
Light (when fitted)		
Colour	Red	Green
Rhythm	Composite group flashing (2 + 1)	Composite group flashing (2 + 1)

Figure A.2: Description of lateral Marks used in Region A

A.2 Cardinal Marks

Ther four quadrants (North, East, South and West) ared bounded by the true bearings (NW-NE, NE-SE, SE-SW, SW-NW). The name of a Cardinal mark indicates that it should be passed to the named side of the mark. A Cardinal mark may be used, for example:

- to indicate the deepest water in that area is on the named side of mark
- to indicate the safe side on which to pass a danger
- to draw attention to a feauture in a channel such as a bend, a junction, a bifurcation or the end of a shoal.
- competent authorities should consider carefully before establishing too many cardinal marks in water-way or areas as this can lead to confusion, given their white lights of similar charecteristics.

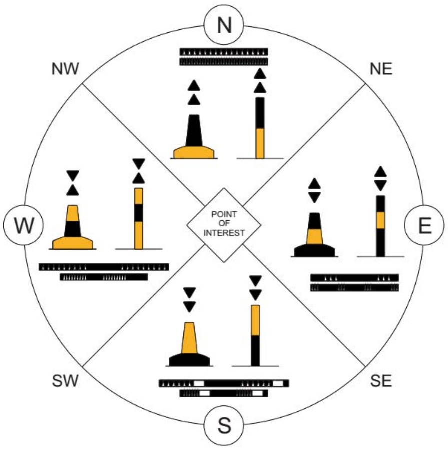


Figure A.3: Cardinal Marks

A.3 Isolated Danger Marks

An isolated Danger mark is a mark erected on, or moored on or above, an isolated danger which has navigable water all around it.

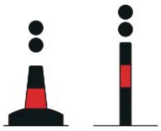


Figure A.4: Isolated Danger mark

A.4 Safe Water Marks

Safe Water marks serve to indicate that there is navigable water all around the mark. These include center line marks and mid-channel marks. Such a mark may also be used to indicate channel entrance, port or statuary approach, or landfall. The light rhythm may also be used to indicate best point of passage under bridges.

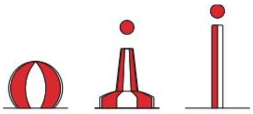


Figure A.5: Safe Water Marks

A.5 Special Marks

Marks used to indicate a special area or feature whose may be apparent from reference to a chart or other nautical publication. They are not generally intended to mark channels or obstructions where other marks are more suitable.

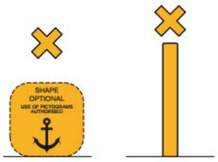


Figure A.6: Special Marks

A.6 Marking New danger

The term ”New Danger” is used to describe newly discovered hazards not yet shown in nautical documents. ”New Dangers” include naturally occurring obstructions such as sandbanks or rocks or man-made danger such as wrecks.

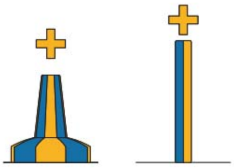


Figure A.7: New Dangers Mark

A.7 Other Marks

A.7.1 Leading Lines/Ranges

A group of two or more marks or lights, in the same vertical plane such that the navigator can follow the leading line on the same bearing.

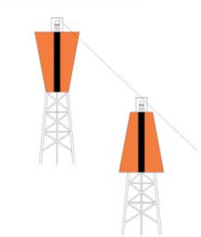


Figure A.8: Leading Lines/Ranges

A.7.2 Sector Lights

A sector light is a fixed aid to navigation that displays a light of different colours and/or rhythms over designated arcs. The colour of the light provides directional information to the mariner. A sector may be used:

- to provide directional information in a fairway
- to indicate a turning poin, a junction with other channels, a hazard or other items of navigational importance
- to provide information on hazard areas that should be avoided
- in some cases a single directional light may be used

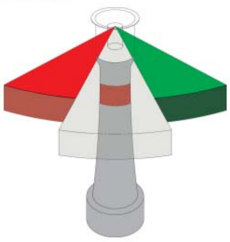


Figure A.9: Sector Lights

A.7.3 Lighthouses

Definition of lighthouse is a tower, or substantial building or structure, erected at a designated geographical location to carry a signal light and provides a significant day mark. It provides a long or medium range light for identification by night.

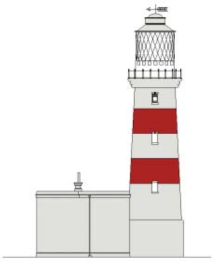


Figure A.10: Lighthouses

A.7.4 Beacons

A fixed a-made navigation mark that can be recognised by its shape, colour, pattern, top-mark, or light character, or a combination of these.

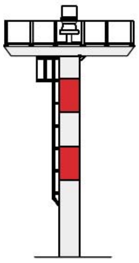


Figure A.11: Beacons

A.7.5 Major-Floating Aids

Major floating aids include lightvessels, light floats and alarge navigational buoys. Major floating aids are generally deployed at critically locations, intended to mark approaches from off-shores areas, where shipping traffic concentrations are high. It may provide a platform for other AIDS to Navigations such as AIS an Aids to Navigation to assist marine navigation.

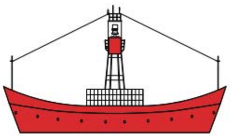


Figure A.12: Major-Floating Aids

A.7.6 Auxiliary Marks

These marks are usually outside of defined channels and generally do not indicate the port and starboard sides of the route to be followed or obstructions to be avoided. They also include those marks and shall be promulgated in appropriate nautical charts and documents. Should not generally be used if a more appropriate mark is available within the MBS.

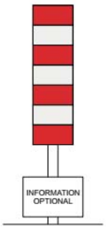


Figure A.13: Auxiliary Marks

A.7.7 Port or Harbour Marks

Mariners should be careful to take account of any local marking measures that may be in place and will often be covered by Local Regulation or by-lays. Before transiting an area for the first time, mariners should make themselves aware of local marking arrangements. Local Aids to Navigation may include, but not be restricted to, marking of:

- breakwater, quays and jetties
- bridges and traffic signals
- leisure areas

and other rivers, channels, canals, locks and waterways marked within the responsibilities of competent authorities.

Bibliography

[1] S. Ahvenjärvi. "Unmanned ships and the maritime education and training". In: volume 1. cited By 0. 2017, pages 245–254.

[2] G.-Q. Bao, S.-S. Xiong, and Z.-Y. Zhou. "Vision-based horizon extraction for micro air vehicle flight control". In: *IEEE Transactions on Instrumentation and Measurement* 54.3 (2005). cited By 34, pages 1067–1072. DOI: 10.1109/TIM.2005.847234. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-20544471218&doi=10.1109%2fTIM.2005.847234&partnerID=40&md5=6085187925c878383513e24a482f48f8.

[3] D. Bloisi and L. Iocchi. "ARGOS - A video surveillance system for boat traffic monitring in venice". In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.7 (2009). cited By 28, pages 1477–1502. DOI: 10.1142/S0218001409007594.

[4] H. Bouma et al. "Automatic detection of small surface targets with electro-optical sensors in a harbor environment". In: volume 7114. cited By 28. 2008. DOI: 10.1117/12.799813.

[5] J. Brownlee. "Boosting and AdaBoost for Machine Learning". In: (2016). URL: https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/.

[6] F. Castanedo. "A review of data fusion techniques". In: *The Scientific World Journal* 2013 (2013). cited By 89. DOI: 10.1155/2013/704504. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84888882639&doi=10.1155%2f2013%2f704504&partnerID=40&md5=5240b8a31c1d990f044fb8c7eace30bb.

[7] P. Dollar et al. "Fast feature pyramids for object detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014). cited By 594, pages 1532–1545. DOI: 10.1109/TPAMI.2014.2300479. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84903622275&doi=10.1109%2fTPAMI.2014.2300479&partnerID=40&md5=083c8799e252abd9586da9c3e9d83a5b.

[8] S.M. Ettinger et al. "Towards flight autonomy: Vision-based horizon detection for micro air vehicles". In: *Florida Conference on Recent Advances in Robotics* (2002). cited By 26.

[9] M. Everingham et al. “The Pascal Visual Object Classes Challenge: A Retrospective”. In: *International Journal of Computer Vision* 111.1 (2014). cited By 415, pages 98–136. DOI: 10.1007/s11263-014-0733-5.

[10] S. Fefilatyev et al. “Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system”. In: *Ocean Engineering* 54 (2012). cited By 18, pages 1–12. DOI: 10.1016/j.oceaneng.2012.06.028. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84865731573&doi=10.1016%2fj.oceaneng.2012.06.028&partnerID=40&md5=c2319bd339e3eb72a2ae5ce6fe5775f1.

[11] S. Fefilatyev et al. “Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system”. In: *Ocean Engineering* 54 (2012). cited By 19, pages 1–12. DOI: 10.1016/j.oceaneng.2012.06.028.

[12] S. Fefilatyev et al. “Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system”. In: *Ocean Engineering* 54 (2012). cited By 19, pages 1–12. DOI: 10.1016/j.oceaneng.2012.06.028.

[13] D. Frost and J.-R. Tapamo. “Detection and tracking of moving objects in a maritime environment using level set with shape priors”. In: *Eurasip Journal on Image and Video Processing* 2013 (2013). cited By 8. DOI: 10.1186/1687-5281-2013-42.

[14] E. Gershikov, T. Libe, and S. Kosolapov. “Horizon Line Detection in Marine Images: Which Method to Choose?” In: *International Journal on Advances in Intelligent Systems* 6.1-2 (2013). cited By 11, pages 79–88.

[15] C. Gonzalez and R.E. Woods. In: *Digital Image Processing* (2013). cited By 1522, page 738.

[16] K.M. Gupta et al. “Adaptive maritime video surveillance”. In: volume 7346. cited By 6. 2009. DOI: 10.1117/12.818330.

[17] J. Kittler et al. “On combining classifiers”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.3 (1998). cited By 3727, pages 226–239. DOI: 10.1109/34.667881. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0032021555&doi=10.1109%2f34.667881&partnerID=40&md5=ab668661a2570340d6758c88fa4a310b.

[18] G. Lemoine et al. “An open source framework for integration of vessel positions detected in spaceborne SAR imagery in operational fisheries monitoring and control”. In: *Proceedings of ENVISAT/ERS Symposium* 572 (2005). cited By 1, pages 252–258.

[19] H. Liu et al. “Omni-directional surveillance for unmanned water vehicles”. In: *Proc. Eighth Int. Workshop Visual Surveillance* (2008). cited By 4.

[20] M. Ludvigsen and A.J. Sørensen. “Towards integrated autonomous underwater operations for ocean mapping and monitoring”. In: *Annual Reviews in Control* 42 (2016). cited By 9, pages 145–157. DOI: 10.1016/j.arcontrol.2016.09.013.

[21] Gonçalo Monteiro et al. "Video Source Object Segmentation ROI AdaBoost Classifier Sub-Window Tracker Bayesian Classifier Voting Scheme Objects Repository Final Classifier Laserscanner Vision-Based System Ladar-Based System Feature Extraction Position and Size Estimation Laser-Camera Coordinate Transformation Object Class and Position Tracking and Final". In: 2006.

[22] International Association of marine aids to navigation and lighthouse authorities. "NAVGUIDE AIDS TO NAVIGATION MANUAL". In: (2014).

[23] T. Perez. "Ship seakeeping operability, motion control, and autonomy - A Bayesian perspective". In: *IFAC-PapersOnLine* 28.16 (2015). cited By 3, pages 217–222. DOI: 10.1016/j.ifacol.2015.10.283.

[24] D.K. Prasad et al. "MSCM-LiFe: Multi-scale cross modal linear feature for horizon detection in maritime images". In: cited By 1. 2017, pages 1366–1370. DOI: 10.1109/TENCON.2016.7848237. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85015419724&doi=10.1109%2fTENCON.2016.7848237&partnerID=40&md5=23d0ad60ee279c67716825846a2dd345.

[25] D.K. Prasad et al. "Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 18.8 (2017). cited By 3, pages 1993–2016. DOI: 10.1109/TITS.2016.2634580. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85009889602&doi=10.1109%2fTITS.2016.2634580&partnerID=40&md5=0889776d27e524d36793054102066b2a.

[26] G. Saur et al. "Detection and classification of man-made offshore objects in TerraSAR-X and RapidEye imagery: Selected results of the DeMarine-DEKO project". In: cited By 13. 2011. DOI: 10.1109/Oceans-Spain.2011.6003596.

[27] M. Seibert et al. "SeeCoast port surveillance". In: volume 6204. cited By 28. 2006. DOI: 10.1117/12.666980.

[28] Roland. Siegwart, Illah Reza Nourbakhsh, and Davide. Scaramuzza. *Introduction to autonomous mobile robots.* eng. MIT Press, 2011, xvi, 453 s. : ill. ISBN: 0262015358, 9780262015356.

[29] M.D.R. Sullivan and M. Shah. "Visual surveillance in maritime port facilities". In: volume 6978. cited By 22. 2008. DOI: 10.1117/12.777645.

[30] M. Szarvas, U. Sakait, and J. Ogata. "Real-time pedestrian detection using LIDAR and convolutional neural networks". In: cited By 49. 2006, pages 213–218.

[31] Sebastian Thrun. "Probabilistic Robotics". In: *Commun. ACM* 45.3 (March 2002), pages 52–57. ISSN: 0001-0782. DOI: 10.1145/504729.504754. URL: http://doi.acm.org/10.1145/504729.504754.

[32] R. Veal and M. Tsimplis. "The integration of unmanned ships into the lex maritima". In: *Lloyd's Marit Commer Law Q* (2017). cited By 1, pages 303–335.

[33] L. Vetter et al. *Marine geographic information systems*. cited By 1. 2012, pages 743–793. DOI: 10.1007/978-3-540-72680-7_23. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84947038079&doi=10.1007%2f978-3-540-72680-7_23&partnerID=40&md5=f9ccac0e9ba2e48aa0c15c7987b59344.

[34] Weiqiang Wang et al. "Modeling background from compressed video". In: *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. October 2005, pages 161–168. DOI: 10.1109/VSPETS.2005.1570911.

[35] Y. Wang et al. "Aquatic debris monitoring using smartphone-based robotic sensors". In: cited By 5. 2014, pages 13–24. DOI: 10.1109/IPSN.2014.6846737.

[36] H. Wei et al. "Automated intelligent video surveillance system for ships". In: volume 7306. cited By 16. 2009. DOI: 10.1117/12.819051.

[37] Paul Westall et al. "Evaluation of machine vision techniques for aerial search of humans in maritime environments". eng. In: *Proceedings - Digital Image Computing: Techniques and Applications, Dicta 2008* (2008), pages 4700018, 176–183. DOI: 10.1109/DICTA.2008.89.

[38] Y. Zhang, Q.-Z. Li, and F.-N. Zang. "Ship detection for visual maritime surveillance from non-stationary platforms". In: *Ocean Engineering* 141 (2017). cited By 1, pages 53–63. DOI: 10.1016/j.oceaneng.2017.06.022. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020426362&doi=10.1016%2fj.oceaneng.2017.06.022&partnerID=40&md5=df0a708af7ff26781e550aebbb24879e.

DTU Electrical Engineering
Department of Electrical Engineering
Technical University of Denmark

Ørsteds Plads
Building 348
DK-2800 Kgs. Lyngby
Denmark
Tel: (+45) 45 25 38 00

www.elektro.dtu.dk