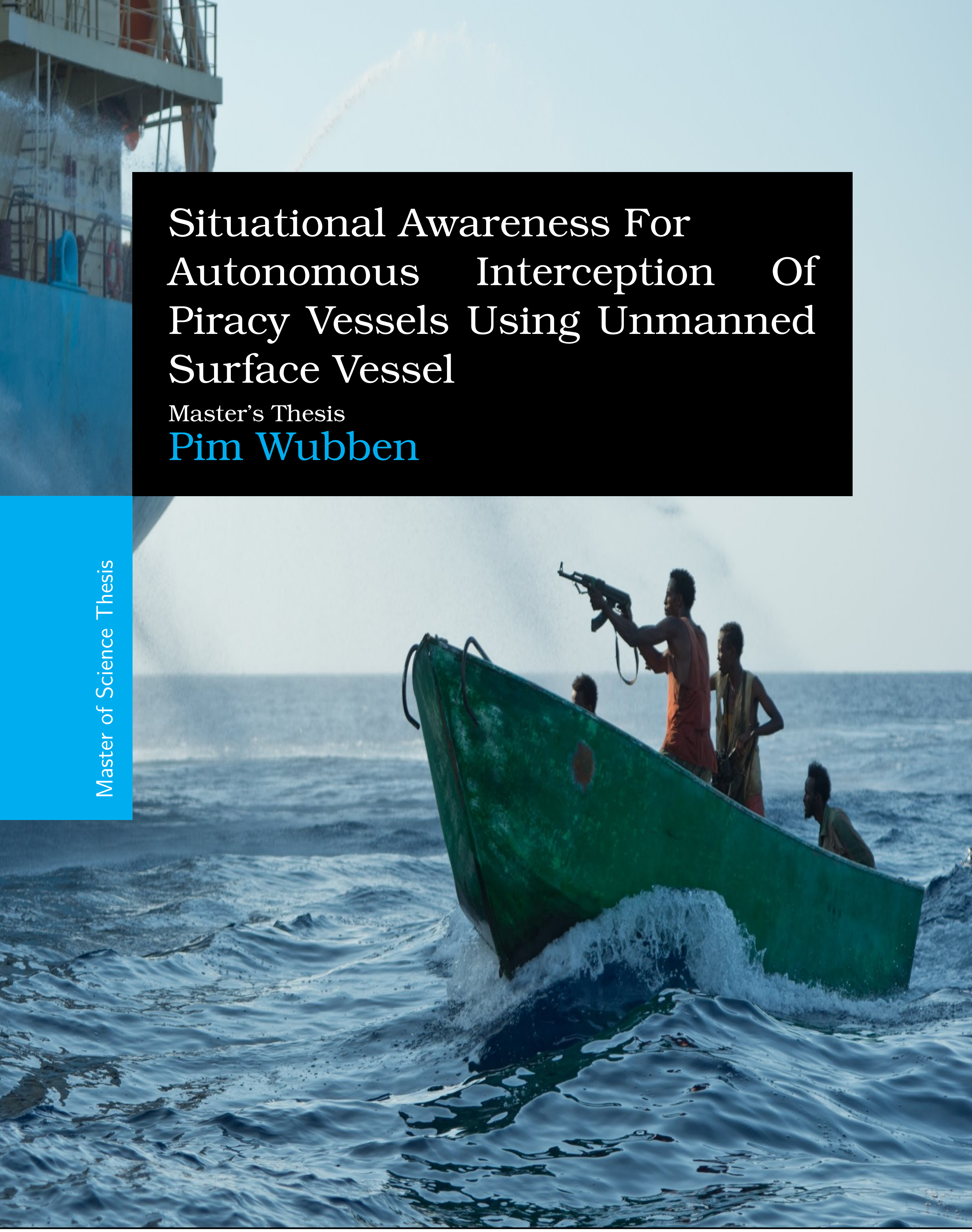


Situational Awareness For Autonomous Interception Of Piracy Vessels Using Unmanned Surface Vessel

Master's Thesis

Pim Wubben

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Situational Awareness For Autonomous Interception Of Piracy Vessels Using Unmanned Surface Vessel

Master's Thesis

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Pim Wubben

May 23, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by SeaState5. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Situational Awareness For
Autonomous Interception Of Piracy Vessels Using Unmanned Surface Vessel

by

Pim Wubben

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: May 23, 2019

Supervisor(s): ____________________
dr. ing. S. Grammatico

____
ir. W. Zwetsloot

Reader(s): ____________________
dr. ir. M. Kok

____
dr. F. Fabiani

____
dr.ir. R.G. Hekkenberg

Abstract

Globally, a growing number of kidnappings executed by pirates occur each year. The current protection consists of mercenaries which are flown in by helicopter to protect merchant ships in threat areas. This is not only expensive, sometimes innocent fishermen are seen for pirates, in some cases with a lethal ending for the fishermen. An alternative could be in the form of an autonomous unmanned vessel, which intercepts potential pirates in a non-lethal manner.

This thesis presents the design of a situational awareness system of such a vessel. The autonomous interception vessel utilizes radar and vision systems to obtain information about the environment. The camera detection algorithm makes use of a modified version of boolean map saliency to segment the target from the background. The radar and camera detections are passed to a global nearest neighbor data association algorithm. The states of the interception vessel are also estimated using sensor data using an EKF. This estimation is used to compensate for the self movement of the interception vessel in the estimation of the states of the target or targets. The target states are are tracked by an interacting multiple model framework, using extended Kalman filters. The algorithm is tested on scenario-specific simulations, which include a visual feed and sensor data. The simulated scenarios are case studies with staged pursuits of the interception vessel and its targets. In the evaluation, multiple algorithms and design parameters are tested and compared.

Next to the design of the algorithm, some modifications and improvements to the existing literature are proposed. A modification to the boolean map saliency algorithm is made to make it less sensitive to target disappearance from the image frame. Also, a motion model is proposed, based on ship dynamics. It is shown that for the generated data set, this model obtained better performance in estimating the velocity and heading of the target. Finally, the measurements are obtained in polar coordinates, while the motion models are defined in Cartesian coordinates. The coordinate change is done using the converted measurement Kalman filter. To convert the single bearing obtained by the camera, the predicted range is used. In order to deal with the increasing uncertainty in the range prediction, a time dependency in the range variance is proposed.

Table of Contents

Preface ix

1 Introduction 1

1-1 Thesis Outline 2

2 Problem Description 3

2-1 Design Requirements 3

2-2 Sensor Application 4

2-3 General Framework 5

2-4 Sensor Data 6

3 Detection 9

3-1 Radar Target Detection 10

3-1-1 Nautis Radar 10

3-1-2 Radar Reconstruction 11

3-2 Performance indicators 11

3-3 Camera Target Detection 13

3-3-1 Wavelet-based Approach 13

3-3-2 Saliency-Based Approach 14

3-3-3 Gaussian Mixture Model 16

3-3-4 Kernel Density Estimation 18

3-3-5 Comparison 18

3-4 Modification of SB algorithm 19

3-5 Camera Bearing Computation 21

3-6 Implementation 22

3-7 Results 23

3-7-1 Test and Algorithm Parameters 23

3-7-2 Computational Efficiency 25

3-7-3 Algorithm Evaluation 26

3-8 Discussion 27

3-9 Sensor Accuracy 28

4 Tracking **31**
4-1 Data Association 32
4-2 Coordinate System 34
4-3 Dynamic State Estimation 37
4-3-1 Basic Tracking 37
4-3-2 Kalman Filter 38
4-3-3 Particle Filter 40
4-3-4 Interacting Multiple Model 41
4-4 Motion Models 42
4-4-1 Standard Motion Models 42
4-4-2 Ship Model 45
4-5 Implementation 49
4-5-1 Target Tracking 49
4-5-2 Beagle Tracking 50
4-6 Coordinate Conversion 51
4-7 Results 56
4-7-1 Test Cases 56
4-7-2 Simulation Limits 59
4-7-3 Evaluation 60

5 Conclusion and Recommendations **71**
5-1 Conclusion 71
5-2 Recommendations 73

Appendices **75**

A Feature Extraction **76**

B Sensor Application **80**
B-1 Thermal Camera 80
B-2 Visible-light camera 82
B-3 Passive Sonar 84
B-4 Radar 85
B-5 Lidar 86
B-6 Sensor Fusion 88

C Detailed Results **89**

Bibliography **91**

List of Acronyms **97**

List of Figures

2-1 General framework of Situational Awareness System (SAS) 5
2-2 Visual feed of the Nautis software 7

3-1 Location of the detection algorithm in the system block scheme 9
3-2 Radar image provided by simulation software Nautis 10
3-3 Discrete wavelet transform on gray scale ocean image [1] 14
3-4 Process of creating the saliency map 15
3-5 Gaussian Mixture Model (GMM) example for one of the pixels [2] 17
3-6 PR curves for original SB algorithm using the norm, and its modification 20
3-7 Virtual screen plane and camera 21
3-8 Angle projection on virtual screen plane 21
3-9 Comparison between bin size 10 and 25 for SB algorithm in sea state 3 24
3-10 Computation speed of algorithms 25
3-11 PR plots for all four scenarios 26
3-12 Histogram of camera detection error values 29
3-13 Wake of vessel causes small bearing offset 29

4-1 Location of tracking algorithm in system block scheme 31
4-2 Simple nearest neighbor and global nearest neighbor 33
4-3 Longitude and latitude 35
4-4 Tracking in the Beagle coordinate system 36
4-5 Tracking in global polar coordinates 36
4-6 Tracking in global Cartesian coordinates 37
4-7 Linear Kalman filter operation [3] 39
4-8 IMM framework 41
4-9 Motion along curved trajectory by vessel 45

4-10 Kinematic bicycle model [4] 46
4-11 Body-fixed coordinate system of vessel [5] 47
4-12 Increasing variance on range between radar updates 54
4-13 Position RMSE result of incorporating time dependent range variance 55
4-14 Trajectories of Beagle and target for scenario 1 57
4-15 Scenario 1 with two targets 57
4-16 Trajectories of Beagle and target for scenario 1 58
4-17 Trajectories of Beagle and target for scenario 3 58
4-18 Visual feed with detected bounding box (green), and sensor data position (red diamond) 60
4-19 RMSE of the position estimate 61
4-20 Position RMSE for different filter and motion model approaches, with the green diamond representing the average Root Mean Square Error (RMSE). From left to right the filter models: Constant-Velocity (CV) - without and with camera resp. (KF), Coordinated-Turn, Constant-Velocity (CTCV) - without and with camera resp. (Extended Kalman Filter (EKF)), Interacting Multiple Model (IMM) - Coordinated-Turn (CT) and CV - without and with camera resp., IMM - CV and Constant Rudder (Ship Model) 62
4-21 Heading RMSE for different filter approaches, with the green diamond representing the average RMSE. From left to right the filter models: CV - without and with camera resp. (KF), CTCV - without and with camera resp. (EKF), IMM - CT and CV - without and with camera resp., IMM - CV and Constant Rudder (Ship Model) 64
4-22 Velocity RMSE for different filter approaches,with the green diamond representing the average RMSE. From left to right the filter models: CV - without and with camera resp. (KF), CTCV - without and with camera resp. (EKF), IMM - CT and CV - without and with camera resp., IMM - CV and Constant Rudder (Ship Model) 64
4-23 Heading errors cannot always be observed using only the bearing 65
4-24 Yaw rate estimation using the CTCV-C model 66
4-25 Average algorithm performance for certain radar drop-out rate 67
4-26 Multiple target tracking, the zoomed-in figures correspond to the black rectangles in the upper images. 67

A-1 HOG depiction of a simulated vessel 77
A-2 Example of strong Haar-like features in face detection 78
A-3 Scale Invariant Feature Transform (SIFT) match between two parts of image sequence with small time interval 79

List of Tables

3-1 Advantages and disadvantages of presented segmentation methods 18
3-2 Qualitative comparison on both algorithms in terms of performance indicators . . 27

4-1 RMSE while tracking multiple targets, track 1 and 2 respectively 67
4-2 Computation time per frame for different filter approaches 68
4-3 Position RMSE [m] comparison of time-dependent Converted Measurement Kalman Filter (CMKF) and mixed-coordinate EKF 69

B-1 Factors determining range of thermal cameras 81
B-2 Price quotations for certain thermal camera models 81
B-3 Detection range (km) given a temperature difference between target and background of 10 °C 82
B-4 Indication of visible-light security cameras 83
B-5 Cost of parts in measurement setup 85
B-6 Cost of commercial Frequency Modulated Continuous Wave (FMCW) radar systems 86
B-7 Maximum detection ranges for different lidar models 87
B-8 Range and angular accuracy for different lidar models 87
B-9 Cost of different Lidar systems 88
B-10 Evaluation of sensor combinations 88

C-1 Results for different filter models, corresponding to Figure 4-20, 4-21, and 4-22 respectively 89

Preface

Based on expectations within the company, the project started as the design of a situational awareness system using a passive sonar system. Only a small amount of literature was available for the tracking of a surface vessel, with a passive sonar system, measured from another surface vessel. To investigate the feasibility, a test setup was produced, consisting of a large triangle with hydrophones on the corners. Based on the difference in arrival time of the signals, the direction of a close-by target can be determined. A test campaign on a nearby lake was conducted. Although the approaching summer was a pleasant addition, we concluded from these tests that the use of a passive sonar was not well suited for this application due to a low detection range and inaccurate bearing estimation. Therefore, the focus of the research shifted towards tracking based on a fusion of radar and computer vision. Fortunately, the course did not have to be changed that rigorously anymore, and this is what became the research direction of my thesis. I did not have experience in computer vision, or in programming in C++. However, I am glad that we took this direction, since it are valuable skills which will probably be incredibly useful later on in my career.

Hereby I would like to thank SeaState5 for giving me the opportunity to work on this socially relevant topic, and for the support trough-out the company. I have really enjoyed working in the environment of the SeaState5 office. Special thanks to my supervisor within SeaState5, Willem Zwetsloot, for taking the time to have fruitful discussions on the topic. Furthermore, I would like to thank my thesis supervisor Sergio Grammatico, for the useful input during our meetings, and for taking the time to give feedback on my work. Lastly, I would like to thank the members of the thesis committee Manon Kok, Robert Hekkenberg and Filippo Fabiani, for showing interest in this thesis. Also, their input on the topics closely related to their research field is greatly appreciated.

Chapter 1

Introduction

Globally, around 50,000 merchant ships are in service. These ships are regularly exposed to the risks of piracy, especially in West Africa, Somalia and the Philippines. Kidnappings, involving demand of ransom, are gaining in popularity. In 2016, the total number of incidents was 191, in which 151 crew members were taken hostage and 62 were kidnapped, tripling the number of cases in 2015.

At the moment, the main protection against pirates are private military teams which are flown in by helicopter when the ships pass the piracy threat areas [6]. This is an expensive operation. Also, it occurs that innocent fishermen are mistaken for pirates. In case of oncoming pirate vessels, an alternative could be to send out an unmanned autonomous vessel from the merchant ship, designated to intercept the pirates. This vessel could be non-lethal to temper the effect of erroneous identification of the vessels. The purpose of such a system is to provide a long-range, non-lethal alternative for current anti-piracy systems. A conceptual design of this idea has been developed within Seastate5, called the "Beagle".

The Beagle was initially conceived as an after-sales add-on for rescue boats. Every merchant ship is, according to the international convention 'Safety of Life at Sea' (SOLAS) of the International Maritime Organization (IMO), required to have at least one rescue boat on board [7]. By equipping this rescue boat with sensors and a smart controller, the rescue boat could be used without a crew to hamper oncoming pirate boats. Possibly the unmanned rescue boat can be equipped with a tear gas cannon or a pepper spray gun to temporarily disable pirates. The Beagle can also become a special purpose boat. This would require the installation of the boat, and its lifting system on the merchant vessel, additional to the rescue vessels already installed. However, this lowers the economic feasibility due to the higher cost. It is not yet determined within the existing concept whether it should be a special purpose vessel or an add-on to existing rescue boats.

Several systems have been developed in the control of boats. Most larger boats and ships have an autopilot on board, which is able to track a desired course based on the measured course and position. For damage-free ship navigation, it is essential to possess accurate knowledge of the environment, called situational awareness. It is the perception of environmental elements and events occurring within relevant distance to the observer. Furthermore, it incorporates the comprehension of the meaning of these elements, events and the prediction of their future state. At the moment, there is no reliable autonomous situational awareness for small vessels, although obstacle detection at sea is discussed in literature. In [8], an obstacle detection for an Unmanned Surface Vessel (USV) was described. Automotive radar and visible-light camera was used to detect obstacles. The movement of the vessel itself was corrected using both the IMU and a horizon detection algorithm to correct for rolling motion. The obstacles are tracked using an Extended Kalman Filter (EKF). However, the obstacles were static and brightly colored, which makes detection and tracking significantly easier. Close range, static obstacle detection was done in [9], using monocular and stereo camera. The presented system operated at 12Hz. Due to the limited suitability of stereo vision for longer range, targets were located only 30 to 100 meters from the USV. As stated in [10], is fully autonomous situation awareness for USVs is still scarce. Especially for situational awareness considering self movement, with dynamic targets, literature seems to be lacking.

At the moment, there are two commonly used systems that tell something about the surroundings of a boat: radar and the Automatic Identification System (AIS). Navigation radar systems of large merchant vessels work with radio waves and therefore have a wide range, but unfortunately small objects (under 10m) are usually filtered out. Otherwise, the radar screen would be ’polluted’ with reflections of waves [11]. This means the objects are not detected until they are very close to the ship. AIS is a system which broadcasts the current position, heading and velocity to other ships. Based on the received AIS data, it checks whether the surrounding vessels’ courses are not on collision with their own course.

However, pirate vessels do not have the AIS system, and have a size smaller than 10m. Therefore, either of the two systems are not sufficient to reliably detect these vessels. Because of this, there is the need for a Situational Awareness System (SAS), which provides reliable information regarding small targets, allowing the Beagle to perform a successful interception.

1-1 Thesis Outline

This thesis is structured as follows.

Chapter 2 presents the problem description. A main design goal is proposed, followed by three design requirements. Furthermore, the sensor fusion choice is explained. To narrow the scope of the project, two assumptions are made. A general framework for the system is presented, which will be used through-out the thesis. Finally, the method which is used for the generation of data is presented.

Chapter 3 presents the methods which were used to detect potential targets in the sensor data. For the camera, two algorithms are implemented and evaluated. For one of the algorithms, a modification is proposed to deal better with target disappearance.

Chapter 4 presents the tracking approaches. Multiple methods are considered and implemented. The algorithm is evaluated on a generated data set, which is scenario specific. Multiple tracking approaches are compared, followed by a reflection on the design requirements.

Chapter 5 makes concluding remarks on the thesis, discusses the work presented in this thesis and identifies topics for further research.

Chapter 2

Problem Description

In this chapter, the problem description is presented. Below, the main design goal is defined. In Section 2-1, three design requirements are proposed. Then, in Section 2-2, the sensor choice is summarized, followed by a general framework for the situational awareness system in Section 2-3. To test the algorithm, a data set is created. The method used for generating a relevant data set is discussed in Section 2-4.

As described in the introduction, piracy is an increasing problem in certain areas of the world. An alternative is sought for the current protection strategy, which consists of flown-in mercenaries. The alternative, proposed by Seastate5, is to equip merchant vessels with an autonomous, unmanned vessel which is capable of non-lethal interception of the pirates. This vessel is called the Beagle.

Information provided by a Situational Awareness System (SAS) can be used by the guidance and control modules of the Beagle to determine which actions should be taken. More detailed state estimation leads to an increase of the probability for interception success. A SAS for such a vessel is to be designed, which leads to the main design goal:

- Design a detection and tracking system for an unmanned surface vessel using radar and vision systems

2-1 Design Requirements

An algorithm is designed to process the data generated by the sensors. To fulfill the main design goal, three design criteria are proposed:

- Target vessel should be adequately detected and tracked, even when the target vessel and the Beagle are manoeuvring.
- The system should be robust to fluctuating target appearance due to waves. Small open boats can disappear from the radar image up to 50% of the time [11]. Therefore, the

SAS should be able to maintain tracks even if the target does not appear in the sensor data for a period of time.

- Real-time operation. Since this is dependent on computation power, it is merely something to keep in mind. The choice of the computing platform has to be made in the final design, as a trade-off additionally involving performance and budget.

2-2 Sensor Application

An observation of the surrounding environment can be made by sensors. Commonly used sensors in environment mapping are Radio Detection And Ranging (RaDAR), Light Detection And Ranging (LiDAR), Sound Detection And Ranging (SoNAR), visible-light camera's and thermal camera's [10]. During the literature study prior to the project, these sensors were evaluated according to performance indicators, such as range, accuracy, cost condition robustness and practicality for the final product. This resulted in a clear overview of sensor advantages and disadvantages. A summary of the sensor evaluation according to the performance indicators is added in Appendix B. Since every sensor has its pros and cons, a fusion of multiple sensors was proposed. Multiple fusion approaches were evaluated according to the same performance indicators as for the individual sensors. The recommendation following from this evaluation proposed a sensor fusion between a Frequency Modulated Continuous Wave (FMCW) radar system with visible-light cameras.

This sensor combination outperformed the individual sensors in terms of the performance indicators. The FMCW radar system performs significantly better in terms of detection range and range accuracy, compared to the other sensors [12]. Compared to a conventional pulse radar system, the FMCW radar is less sensitive to clutter, due to its better range accuracy. However, radar systems have an update frequency of about 0.5 Hz, which is quite low in case of a high speed pursuit. Also, it does not provide any additional appearance information about the potential target. Therefore, the addition of a camera system is proposed, which provides high angular detection frequency. Although this is outside the scope of this thesis, the camera does provide the opportunity to obtain additional appearance information. For instance, it could be determined whether a radar detection is indeed a target, using a pre-trained classifier. Another possibility could be to detect evasive manoeuvres of a target at a sooner stage, based on appearance changes.

The movement of the Beagle is also tracked, for which three types of sensors are used. It is desired to know the location of the Beagle, as will be discussed in Section 4-2. Highly accurate position measurements can be obtained using Differential Global Positioning System (DGPS). A DGPS module provides an accurate global location on the Earth. This location is accurate to about 2 meters. Furthermore, the heading and yaw rate are measured. The heading is measured by a Digital Compass (DC), which uses the calibrated internal electronics to measure the response of the device to the magnetic field of the Earth. The yaw rate is measured by the Inertial Measurement Unit (IMU) of the vessel. The IMU uses a combination of gyroscopes and accelerometers to, among other things, measure the yaw rate. Sometimes, a DC is also incorporated in the IMU.

2-3 General Framework

The general framework of the SAS can be seen in Figure 2-1. The three colored blocks represent the three main divisions within this thesis. Of the three, the sensor division is relatively small, and is discussed in this chapter. The other two are discussed in separate chapters.

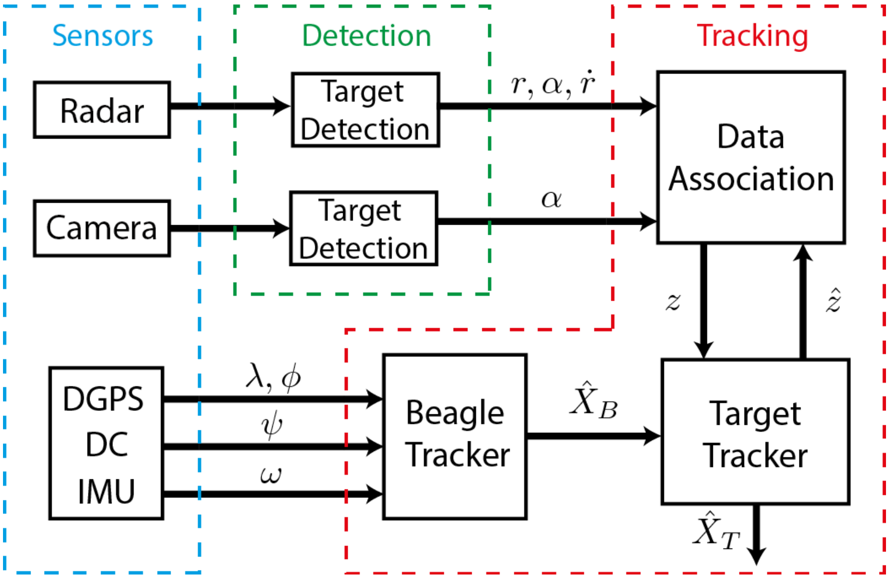


Figure 2-1: General framework of SAS

Sensors - The radar image and camera image are passed to the detection algorithm. Measurements concerning the movements of the Beagle are done using the other three sensors. Measurements are done using a DGPS module, a DC and an IMU, to obtain longitude λ and latitude ϕ, heading ψ and yaw rate ω respectively.

Detection - The radar image is processed to obtain an initial potential target detection. This detection is in the form of the measurement vector $z = \begin{bmatrix} r & \alpha & \dot{r} \end{bmatrix}$, which represents the range, bearing and range rate. The bearing is the angle relative to the Beagle heading. The camera detection algorithm receives a camera image from the sensor system in every time step. Potential targets are detected in the image, which is transformed into a bearing measurement.

Tracking - The data association algorithm determines which detections are most probable to correspond to a certain track. This is determined using the predicted measurement $\hat{z}$ provided by the target tracker. The most probable detection z is assigned to a track, and is used to update the target state estimate $\hat{X}_T$. The Beagle state estimate X_B is used to determine the global position of the target. If necessary, tracks can also be initiated or terminated. When not all detections can be associated to a track, a new track is initiated. If a track is not associated with any detections for some time, the track is terminated.

2-4 Sensor Data

The previous section presented the three main divisions within the thesis. In this section, the method which was used to obtain the sensor data is presented, corresponding to the blue division in Figure 2-1.

The implementation of most marine computer vision projects have been tested on small data sets. These data sets are mainly focused on near-field targets or large merchant ships, such as the MarDCT data set [13]. The PETS2016 data set contains videos of small targets, recorded from a large ship [14]. However, no ground truth position is incorporated. Moreover, we require data from a relatively small USV. To obtain data of small, far-field and near-field vessels, in different sea and weather conditions would require an extensive measurement campaign. All data has to be collected or reconstructed in a synchronized manner. The project is currently in a conceptual phase. Therefore, the system and algorithms do not necessarily have to be designed and tested in a real environment. Results can also be shown in a simulated, fully controlled environment, which is assumed to be close to a real-world scenario.

Motivated by this, we propose to use the maritime simulation software Nautis, created by the company V-Step. Their software is currently used to train people to operate vessels in all sorts of conditions. The software can provide 360° visual information around the ship, of which a snapshot of the rotating camera is shown in Figure 2-2. Furthermore, the system provides sensor data such as radar, Automatic Identification System (AIS), DGPS, and IMU. The software is used to simulate a realistic scenario using the proposed sensors in the desired environmental conditions. The vessels are assigned to predefined tracks, which is used to simulate pursuits in different environmental conditions. Camera stabilization is done in the software, and is therefore not implemented separately. Ideally, the algorithm is implemented in cooperation with the simulation software, such that it is possible to repeat the simulations many times, each time with a different (random) sea motion. Unfortunately, this is not possible due to the data exporting protocol of the software. Only one vessel can export its data in each session. Therefore, multiple sessions have to be recorded and synchronized to obtain a complete data set. To design and test the algorithm, it is necessary to obtain data of relevant scenarios. The required data is:

- Radar image data sequence. This information is used for initial target localization on a relative map. Extracted parameters are range, bearing and range rate.
- Video sequence from the camera on the Beagle. This data is used to design and test algorithms for target detection, feature extraction and tracking.
- Beagle sensor data. Data from the IMU, DC and DGPS is used as input to the state estimation filter of the Beagle.
- DGPS data of the vessels. The DGPS data can be used to generate a ground truth reference, which is used to evaluate the algorithm.

Stabilization of a camera image sequence is a widely studied subject in recent years. Multiple approaches and algorithms are available for this [15][16][17]. Since the simulation can be set-up with built-in camera stabilization, this is not implemented separately. Furthermore, it is possible to rotate the camera on the Beagle. This could be beneficial in case of the vessels manoeuvring close together, as the target is more prone to leave the Field Of View (FOV) in such a case. However, for this thesis the FOV of the camera is kept constant, with the center of the image right in front of the Beagle.

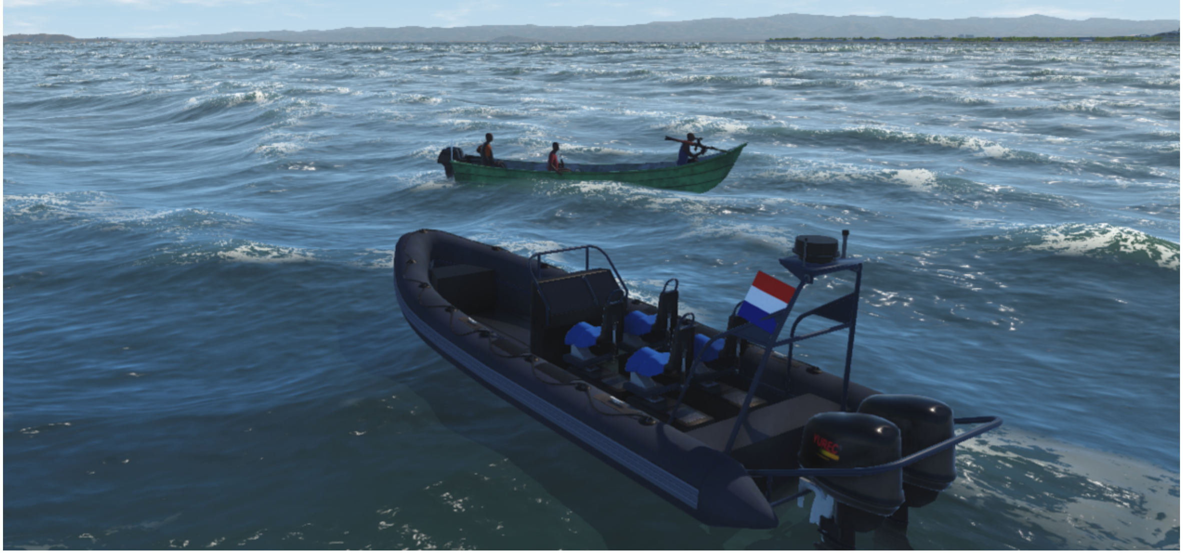


Figure 2-2: Visual feed of the Nautis software

Chapter 3

Detection

The first step towards tracking a target is determining if and where it is for a certain time step. The detection algorithm processes the images from the radar and camera to obtain potential target detections, as illustrated in Figure 3-1. The radar processing has a lower complexity than the camera processing. Therefore, the radar processing will be briefly discussed first in Section 3-1. In the remainder of this chapter, the camera detection algorithm will be discussed, starting with the proposal of some performance indicators in Section 3-2. In Section 3-3, multiple detection approaches are discussed for the camera detection algorithm. The two most promising detection algorithms have been implemented. For one of the algorithms, a modification is proposed in Section 3-4. Using the position of the classified pixels, the bearing can be computed, which is discussed in Section 3-5. In Section 3-6, the implementation of the algorithms is explained in more detail. Finally, some results are discussed in Section 3-7, followed by a short discussion which reflects on the performance indicators in Section 3-8.

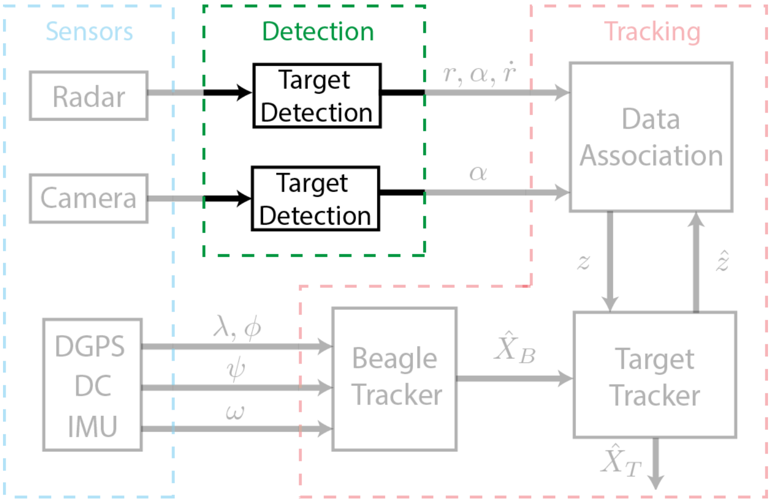


Figure 3-1: Location of the detection algorithm in the system block scheme

3-1 Radar Target Detection

Radar detections are used to initialize tracks. An initial radar detection is required to initiate a track. The information of the radar detection is communicated to the data association algorithm. An FMCW radar system generally provides the measurement vector $z = \begin{bmatrix} r & \alpha & \dot{r} \end{bmatrix}$, where r is the range, α is the bearing and $\dot{r}$ is the range rate. The latter is obtained by a so called Doppler measurement. If a target moves towards or from the Beagle, the frequency of the received radar echo increases and decreases respectively. The magnitude of the relative velocity difference in the target direction determines the amount of increase or decrease of the frequency. Therefore, this can be converted to a relative range rate. The simulation software creates a radar image, based on specifications of a basic navigation radar system. A target detection algorithm has been implemented for this image, which will be discussed first in Subsection 3-1-1. However, to obtain the desired accuracy and range rate measurements, sensor data has been used to reconstruct the radar data. This will be discussed in Subsection 3-1-2.

3-1-1 Nautis Radar

The radar image provided by the simulation software Nautis is a replica of a basic navigation radar system. Figure 3-2 shows a close-by target vessel, and the recording vessel. The two dots in the middle of the image are the bow and stern of the recording vessel, of which the reflections are also picked up by the radar. The stronger the reflection of a target, the higher the intensity of the pixel on the radar image. This means we can use an intensity threshold to obtain targets of a certain reflection strength. This results in a mask containing all possible targets, and the location of the recording vessel. Note that this radar image is relative to the recording vessel heading, not to the world map. So when a target is located north of the center, it is located in front of the Beagle.

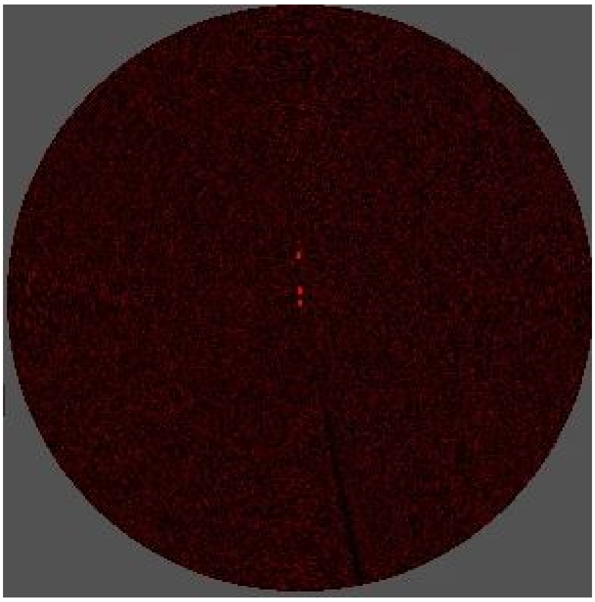


Figure 3-2: Radar image provided by simulation software Nautis

The location of the recording vessel is taken as the mean of the bow and stern detections. The relative target range and bearing can simply be computed using the Pythagorean theorem. In general, the detection is relatively robust. However, in increasing sea state, the target might be difficult to detect. Due to waves, the signal to noise ratio decreases. Therefore, the camera image is used to obtain more information.

The radar image is checked for new detections at 15Hz. However, the rotation frequency of the radar is about 0.37Hz, i.e. the radar is updated every 2.7 seconds. Also, the Nautis radar does not provide range rate measurements.

3-1-2 Radar Reconstruction

As discussed in the previous subsection, the simulated radar in Nautis has a couple of drawbacks. As discussed in Section 2-2, it was decided to use a FMCW radar system. However, the simulated radar in Nautis does not provide Doppler measurements. Also, the simulated radar is a standard navigation radar. Therefore, the variance on the measurements is relatively large. Furthermore, the update frequency is 0.37Hz, instead of 0.5Hz.

To overcome these drawbacks, separate radar data is created using the DGPS data of the vessels, which is exported by the software. The DGPS data provides accurate coordinates of the vessels. Therefore, the computation of the relative polar coordinates of the radar is straightforward. Measurement noise is added, with a range and bearing standard deviation $\sigma_r = 5m$, and $\sigma_\alpha = 0.6°$ respectively [12]. The Doppler measurement requires the velocity of the two vessels. However, no velocity measurement is provided by the simulation software. Therefore, the velocities are reconstructed using an Extended Kalman Filter (EKF). This is done using the approach which will be discussed later in Subsection 4-5-2. By combining the reconstructed velocities and the vessel headings and positions, the relative range rate can be computed as

$$\dot{r} = v_T \cos(\alpha + \psi_B - \psi_T) - v_B \cos(\alpha), \tag{3-1}$$

where $\dot{r}$ represents the range rate, v_T and v_B the velocities of the target and the Beagle respectively, ψ_T and ψ_B the heading of the target and the Beagle respectively, and α the relative bearing between the Beagle and the target. In [12], the range rate resolution of the used FMCW radar is less than 1 m/s. Since the velocities are reconstructed and the error standard deviation cannot be determined, no extra measurement noise was added. The reconstructed radar data is used for the remainder of this thesis.

3-2 Performance indicators

Some design requirements have been presented in Section 2-1. To conclude on the performance of the algorithms according to these requirements, some performance indicators have been set. The camera detection approaches are compared according to the following performance indicators

- Computation intensity
- Precision and Recall

- Suitability for static and dynamic targets
- Robustness to clutter
- Ability to overcome camera movement

The computation intensity of the algorithms is quantified by measuring the time it takes to process one frame, averaged over the entire data set. The total data set consists of 4 separate sets, in total containing around 6000 images. Precision and recall are performance metrics which are explained below. At the initial moment of detection, a hostile vessel might move straight towards the merchant vessel. Since the Beagle is released from this position, the two vessels will move straight towards each other. Therefore, the target will appear visually static. At some point, the hostile vessel might start evasive manoeuvres, making it visually dynamic. Therefore, the algorithm should be suitable for both situations. Clutter robustness is derived from the requirement stating the system should be robust to fluctuating target appearance due to waves. It is important that when the target disappears, the system correctly recognizes this disappearance. Furthermore, when a lot of clutter detections occur, it is more difficult for the data association algorithm to match the correct detection to the correct track. Finally, the algorithm should be able to deal with movement of the camera, since the Beagle, and thus also the camera, will inevitably move.

As stated above, one of the performance indicators is the precision and recall. The meaning of this commonly used metric is discussed in the following paragraph [18].

Precision - Recall - The performance of a computer vision algorithm can be measured in terms of precision and recall. Precision attempts to answer the question: What proportion of the identifications was actually correct? This is done by computing

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3-2}$$

where TP represents the number of true positives, which is the number of times a target is correctly classified as target. FP represents the false positives, which is the number of times a presumed target detection is induced by clutter. Recall attempts to answer the question: What proportion of actual positives was identified correctly? This is determined by computing

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3-3}$$

where FN represents the number of False Negatives. This is the number of times the target was not detected, while in fact it was there. A precision-recall (PR) curve shows the trade-off between the two for different thresholds. High scores for both show that the algorithm is returning accurate results (high precision) and returning a majority of the actual targets as foreground (high recall). Thresholds over a certain range are used during the performance evaluation. The threshold range is chosen such that the recall ranges from 0 to 1. This corresponds to the target being detected in none of the images, and the target being detected in all of the images.

3-3 Camera Target Detection

Significant amount of research has been done in the field of computer vision. Especially the segmentation of static backgrounds with dynamic foreground is widely studied [19]. However, in this case also the background is highly dynamic. The algorithm has to deal with the dynamic nature of the waves, foam and wakes, illumination changes, dynamic and (visually) static targets and unknown target appearance. The algorithm is made and tested using the simulated data extracted from Nautis. The camera image can be split up in color channels. These channels represent primary colors, of which other colors can be constructed. An image from a digital camera will have a Red, Green and Blue channel (RGB). These channels contain the primary color intensities of the pixels. These channels can be used as direct input to detection algorithms. It is also possible to extract more information out of these channels, often called feature extraction. In this section, the processing of the RGB color channels is discussed. In Appendix A, some feature extraction approaches are discussed. However, to meet the real-time processing requirement, no feature extraction approaches have been used in this thesis.

The camera image is processed to obtain potential target locations in the image frame. These potential target locations are found by segmenting the image, which can be done in many different ways. Object detection is generally done in two main steps: background subtraction and foreground segmentation. The challenge in background subtraction arises due to the dynamic nature of the water background, in the form of waves and wakes. Background subtraction can in turn be split up in two parts: static background subtraction and dynamic background subtraction. Static background subtraction assumes the background remains the same over time.

The camera detection algorithm is used to separate the noisy ocean background from the objects of interest. The bearings of potential targets are passed to the data association algorithm. Methods that are promising to implement are wavelet-based filtering approaches [1], Saliency-Based (SB) approaches [20], Gaussian Mixture Model (GMM) or Kernel Density Estimation (KDE) [13][10]. These approaches will be discussed in the following subsections.

3-3-1 Wavelet-based Approach

This method relies on the assumption that the presence of a target causes a certain offset or spike in the wavelet domain, compared to the more repetitive background. Using the discrete wavelet transform, an image is decomposed into four frequency bands: low-low (LL), low-high (LH), high-low (HL), and high-high (HH). These frequency bands are obtained by applying analysis filters h and g, which correspond to a particular type of wavelet, followed by a downsampling step. These filters are low and high pass filters [21]. The decomposition provides sub bands corresponding to different resolution levels and orientations. Figure 3-3 depicts this two step process.

In [1], the authors propose to use the discrete wavelet transform and support vector machine on low frequency wavelets were used to detect objects. This operation was followed by a correlation step over 5 frames, and adaptive segmentation. Low frequency wavelets were assumed to contain less information of clutter. Therefore, the uncluttered background would not correlate over the frames, so both clutter and background could be taken care of.

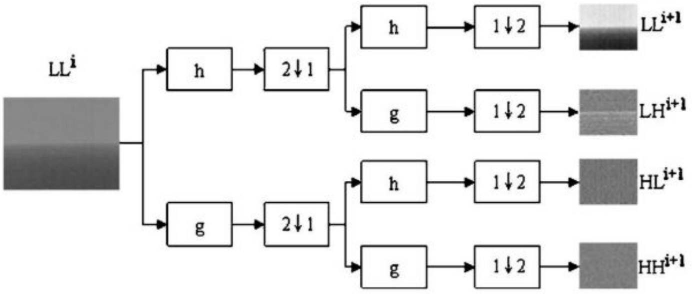


Figure 3-3: Discrete wavelet transform on gray scale ocean image [1]

Applicability to the data

Due to the nature of the method, it is expected that it will not perform very good in the presence of wave crests. This is because it will filter out details in the image. Also, small, distant objects are difficult to detect, since they appear as details in the large image. Therefore, the method is suitable in case of flat water, but will not be robust enough to handle higher sea states.

3-3-2 Saliency-Based Approach

This approach makes use of the assumption that the target appears as a relatively small salient point in the image, where the rest of the image should contain relatively large areas with similar characteristics. The approaches of [22] and [20] make use of Boolean Map Saliency (BMS). The boolean maps are generated by thresholding of the input image feature maps. Any feature channel could be used (color, orientation, motion, etc.). However, this implementation only used the color channels. A color whitening step is used to obtain equal importance of different color channels. After this, each of the channels R, G and B are normalized to the range [0,255] and sampled using a threshold function at intervals with a step size δ. The function assigns a 1 to a pixel with a value above the threshold, and 0 otherwise. This results in a set of N boolean maps, $B_i{}_{i=1}^{N}$. Activation maps are created from these boolean maps by identifying surrounded regions. The activation map $A_i(B_i)$ is created by setting pixels to 1 if it lies in a surrounded region of B_i, and 0 elsewhere. The activation maps are splitted into the sub-activation maps $A_i^+(B_i)$ and $A_i^-(B_i)$:

$$\begin{aligned} A_i^+(B_i) &= A_i(B_i) \wedge B_i \\ A_i^-(B_i) &= A_i(B_i) \wedge \neg B_i, \end{aligned} \tag{3-4}$$

where $\wedge$ is a pixel-wise conjunction operator and $\neg B_i$ is the negation of B_i. An intuitive interpretation is that $A_i^+(B_i)$ activates the surrounded peaks above the threshold and $A_i^-(B_i)$ activates the surrounded valleys below it. The resultant activation maps are normalized so that maps with small concentrated active areas will receive more emphasis. This is done by L2-normalization. Sub-activation maps with small, scattered active areas, the maps are dilated with a kernel K of size ω_1 before normalization.The sum of all activation maps gives the normalised activation map $\bar{A}$.

$$\bar{A} = \sum_{i=1}^{N} \frac{A_i^+(B_i) \oplus K_{\omega_1}}{\left\|A_i^+(B_i)\right\|_2} + \frac{A_i^-(B_i) \oplus K_{\omega_1}}{\left\|A_i^-(B_i)\right\|_2}, \tag{3-5}$$

where $\oplus$ represents the morphological dilation operation. The intermediate saliency map M is obtained by performing min-max normalization. The final saliency map S is obtained by a second dilation operation using a kernel K of size ω_2, followed by Gaussian smoothing G with a standard deviation σ.

$$M = \frac{\bar{A} - \min(\bar{A})}{\max(\bar{A}) - \min(\bar{A})}, \tag{3-6}$$

$$S = G_\sigma(M \oplus K_{\omega_2}). \tag{3-7}$$

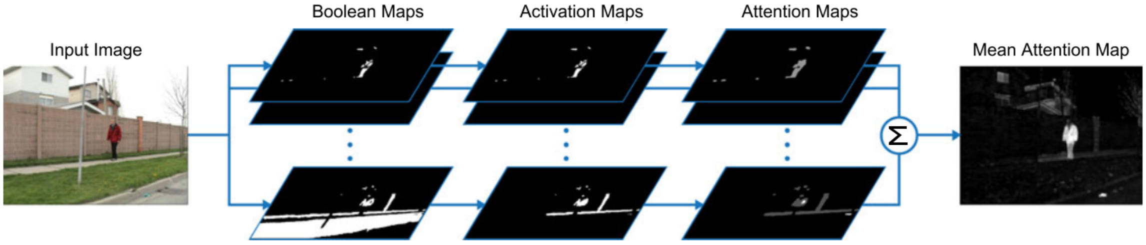


Figure 3-4: Process of creating the saliency map

It was argued in [22] that stopping here gives a tendency of highlighting the wakes of the boats and specular reflections in the water. Therefore, the authors propose to use broadly tuned, intensity decoupled red, blue and green color channels. The channels are decoupled from intensity by dividing the RGB channels of the image by the intensity channel I by computing

$$I = \frac{r+g+b}{3}. \tag{3-8}$$

The channels are set to zero for pixels where I is less than $\frac{1}{10}$ of the maximum value I_{max}. The broadly tuned color channels are defined as

$$\begin{aligned}
\bar{R} &= \begin{cases} \frac{r-(g+b)/2}{I} & \text{if } I > \frac{I_{max}}{10} \\ 0 & \text{otherwise} \end{cases} \\
\bar{G} &= \begin{cases} \frac{g-(r+b)/2}{I} & \text{if } I > \frac{I_{max}}{10} \\ 0 & \text{otherwise} \end{cases} \\
\bar{B} &= \begin{cases} \frac{b-(g+r)/2}{I} & \text{if } I > \frac{I_{max}}{10}. \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{3-9}$$

The obtained saliency map is binary thresholded to obtain candidate object regions. To be able to generalize for different regions, hysteresis thresholding can be applied. An upper and lower bound on the threshold is selected. The saliency map is binary thresholded at the upper bound and a flood-fill algorithm is then used to grow regions and add connected pixels which are above the lower bound. Candidate objects are then found by labelling connected components and creating bounding boxes. In [22], the authors propose a lower bound on the bounding boxes to get rid of false detections from the background.

Applicability to Data

The framework presented in the previous subsections builds on the assumption that the salient parts in the image are substantially different in intensity on at least one of the channels, opposed to most of the background. This means that small objects which have a different color compared to the background will be easily detected. However, objects with nearly the same color as the background are more difficult to detect, since no emphasis is created on the object due to the normalization of that activation map. It is beneficial that the color distribution of the ocean is relatively uniform and spatially repetitive. Therefore, this will not obtain a lot of emphasis. However, as the authors in [22] state in the conclusion, distant and dark objects are difficult to detect. On the other hand, intuitively it is rather unlikely that a vessel, including persons, taking along shadows and lighting, will be able to camouflage themselves good enough to be the same color as the water or background. Therefore, the BMS algorithm is expected to be applicable for the majority of the targets.

3-3-3 Gaussian Mixture Model

Since we are dealing with a background which changes over time, it is desired that the algorithm is not sensitive to this change. This can be done by utilizing an adaptive background model. It is not known beforehand what the change in background will be exactly. Multiple factors influence the background, such as sea state, weather conditions and lighting.

The GMM is a probabilistic method which corresponds to a mixture distribution that represents the probability distribution of certain observations or states. The pixel intensity values are assumed to be Gaussian distributed random variables. Wakes and foam make that images of the ocean are invariably multimodal, since these have a different color compared to the waves. Therefore, we have to describe the development of the color intensities in the pixel using multiple distributions. GMM are suitable for representing multimodal backgrounds. The probability $P(I)$ that a pixel, which belongs to the background, has intensity I is given as:

$$P(I) = \sum_i \omega_i G(\mu_i, \sigma_i), \tag{3-10}$$

where $G(\mu_i, \sigma_i)$ represents the i^{th} Gaussian distribution with mean μ_i and standard deviation σ_i. The weight ω_i represents the portion of the data which is accounted for by the specific Gaussian. As much Gaussian distributions as desired can be added, although that will not necessarily improve the result. Pixels which have an intensity which has an Euclidean distance of a certain times the standard deviation to either one of the means, are classified as foreground. The GMM is fitted for the RGB values of the pixels over multiple frames [23]. If the pixel process would be a stationary process, this fit could be done using a so called Expectation-Maximization (EM) algorithm, which is an iterative method to find the maximum-likelihood estimates for the model parameters [24]. However, the pixel values will vary over time. Running the EM algorithm in each step is computationally expensive. Therefore, a recursive approximation of this algorithm has been proposed. A mixture model exists for every pixel. All new pixel intensity values x_I, are checked against the existing D distributions for that pixel. When the pixel value is within 2.5 standard deviations of one of the distributions, it is considered a match. An example is illustrated in Figure 3-5. If none of the distributions match the pixel value, a new distribution is added. The D distributions are weighed. These weights $\omega_{d,k}$ are updated as follows

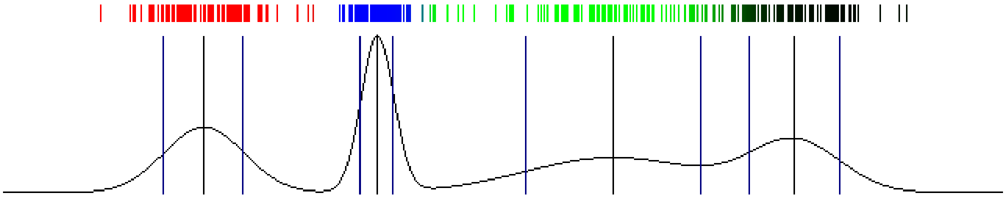


Figure 3-5: GMM example for one of the pixels [2]

$$\omega_{d,k} = (1-\alpha)\omega_{d,k-1} + \alpha(M_{d,k}), \tag{3-11}$$

where α is the learning rate, d represents the distribution, k is the current time step and $M_{d,k}$ is 1 for for a matched model, and 0 otherwise. The weights are normalized after this step. The μ and σ parameters of the matched models are updated as follows

$$\begin{aligned} \mu_t &= (1-\alpha)\mu_{t-1} + \alpha x_I \\ \sigma_t^2 &= (1-\alpha)\sigma_{t-1}^2 + \alpha(x_I - \mu_t)^T(x_I - \mu_t), \end{aligned} \tag{3-12}$$

A newly added distribution is not allowed to become part of the background model until it becomes the K^{th} most probable distribution. This way, an outlier in the pixel intensity does not destroy the background model. This allows for a foreground object to remain at the same place for a short amount of time, without immediately being mistaken for background.

Applicability to Data

The downside of the method is that in general, it works on movement. The rotate and zoom action discussed in the previous chapter results in the fact that at the start of the recording, the target might already be in the image at initialization. This makes that the vessel might be learned as background, resulting in poor segmentation performance. However, the vessel will not remain exactly at the same place relative to the Beagle. If the vessel is visible at all, this means it differs from the environment in some sense. Therefore, when the vessel moves into image pixels which are fitted on ocean background, it is classified as foreground. It would be possible to insert different features as well. This should emphasize the relatively abrupt transition of the sea colors to the vessel colors, this would be helpful when the vessel color is close to that of the waves.

3-3-4 Kernel Density Estimation

Similar to the GMM, the KDE uses probability distributions to describe the system. However, it does not necessarily use Gaussian distributions, but allows the kernels to be asymmetric or have suitable statistical properties. In general, a few frames are used to compute the probability distribution and the kernels are fit on these. Because of the asymmetric kernels, the computational cost is higher than for a GMM. The probability $P(I)$ is modeled as:

$$P(I) = \frac{1}{n}\sum_{i=1}^{n} K(I, \kappa_i), \tag{3-13}$$

where $K(I, \kappa_i)$ represents the kernel function for the pixel intensity I and the kernel parameters κ_i for the i^{th} kernel. The variable n represents the total number of kernels [25]. The asymmetric kernels make that the KDE can deal with illumination changes and wakes better than GMM.

Applicability to Data

In theory, KDE models should be able to better fit the presented data, due to the ability of the use of asymmetric kernels. Therefore, it should be able to deal with foam and wakes better than GMM. However, the computational complexity is an issue. In [26], images of (320 x 240) resolution are processed in a rate up to 0.97 s/frame. In [25], the authors show that the KDE produces a lower false alarm rate than the GMM. However, it is only capable of running 7 fps processing (160 x 120) resolution images. To improve this performance, a lot of computing power will have to be added. For this application, a higher resolution and frame rate will be necessary. Therefore, the KDE is not suitable for this application without changing the design requirements.

3-3-5 Comparison

The comparison of the pros and cons of the different algorithms is made in Table 3-1.

Table 3-1: Advantages and disadvantages of presented segmentation methods

Method	Advantages	Disadvantages
Wavelet-based approach	- Simple - Computation efficient	- Cannot handle large wave dynamics
Gaussian Mixture Model	- Adaptive to changing conditions - Computation efficient	- Static targets can be learned as background
Kernel Density Estimation	- Can deal with wakes better than GMM	- Computation intensive
Saliency-Based Approach	- Can deal with foam and waves - Not influenced by camera motion	- Addition of temporal information makes it computation intensive - Small, dark objects might be difficult to detect

The algorithm should be able to deal with waves, wakes and foam. The wavelet-based approach is not able to handle large wave dynamics. Therefore, the wavelet-based approach is not considered suitable for our needs. The KDE algorithm has high computation intensity, which conflicts with the requirement of real-time operation. Because of the relatively large range of the target, it is expected that the image resolution will have to be increased compared to presented literature, which was already unsuitable for real-time operation. From this point of view, the KDE is not suitable for the application. The two remaining approaches, GMM and SB, both do not appear to have large fundamental issues. Slight camera motion makes it likely that a target will never be completely visually static. This decreases the probability that the target is learned as background by the GMM. The difficulty in segmentation of small, dark objects using the SB approach is not expected to be a large issue, but this will be investigated. Therefore, these two algorithms have been implemented. To meet the real-time operability constraints, no additional feature extraction methods have been implemented in the Situational Awareness System (SAS).

3-4 Modification of SB algorithm

In this thesis, a modification to the SB algorithm is proposed, which will be discussed in this section, accompanied by a performance comparison. The modification decreases the false-positive detection rate in case of a fluctuating target appearance, or fully disappeared target. It is shown that the proposed modification improves the precision-recall performance.

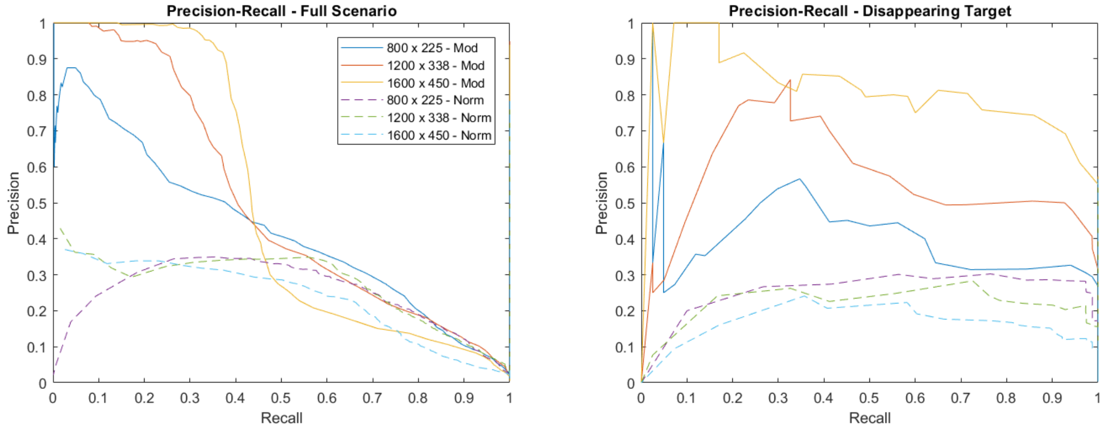
In the original SB algorithm, the final result map is normalized using min-max normalization. This means the map is normalized relative to the largest and lowest intensity value in the resulting map, as described in Equation (3-6). This accounts for deviations in the image resolution and saliency of objects. However, this approach inherently makes the assumption of the presence of a target. When the target disappears from the image, the resulting saliency map is amplified by the normalization. This results in false positives when thresholding is done using a static value. In [22], thresholding is done using adaptive hysteresis thresholding. However, this still poses the same issue, since the background pixels will now contain the highest intensity values, and are thus classified as foreground. Furthermore, changes in the environment, target size or lighting make it difficult to determine a proper threshold.

If the final normalization step is omitted, the resulting map does no longer fluctuate in case of a disappearing target. However, the resulting map does change with changing environment or image resolution. This has to be handled by the thresholding step. Assuming a target will not disappear behind waves for longer than a certain amount of time, we could use a threshold based on the maximum pixel value of a number of past frames. However, in case of longer target disappearance, the threshold value would be corrupted, resulting in false positives. Also, the offset from the maximum value, which results in the threshold, would have to scale with the maximum value. Another option is to assume the resulting map to be Gaussian distributed. From the fundamental basis of the algorithm, we expect the target to be at least somewhat more salient than its surroundings. Thus, the outliers are classified as objects of interest. At each time step, the mean and standard deviation of the resulting map is computed. The threshold is computed as a certain times the standard deviation from the mean:

$$T_k = \mu_k + n\sigma_k, \tag{3-14}$$

where μ_k is the mean value of the resulting map at time k, σ_k the standard deviation at time k and n being the variable determining the height of the threshold. This thresholding approach scales with the result map. In case of a target disappearance, the standard deviation decreases. However, due to the relatively small target size compared to the image resolution, this is not significant.

In Figure 3-6, the comparison between the original and modified version of the algorithm can be seen. Using the simulation software a data set is created. The data set contained both partial target occlusion behind waves, as complete target disappearance from the Field Of View (FOV) of the camera. The target first approaches the camera, and leaves the frame around 75% of the total time. In Figure 3-6(a), we observe the result for the complete scenario, including approaching target vessel. Figure 3-6(b) shows the result while testing only on the end of this scenario, which involves the target vessel moving out of the frame. The test data for (b) contains a target vessel in the frame for about half of the frames. This is done to emphasize the influence of the target disappearing from the frame.



(a) **(b)**

Figure 3-6: PR curves for original SB algorithm using the norm, and its modification

In Figure 3-6(a) we see that the performance of the modified algorithm is close to the original algorithm for lower thresholds, where we have low precision and high recall. However, for higher thresholds, which inevitably result in lower recall, the precision performance is significantly better. Therefore, the modified algorithm is more robust to false positives.

Figure 3-6(b) shows the influence of complete target disappearance. In this case, the algorithm modification performs significantly better than the original over the entire threshold span. The false positive rate of the original algorithm is significantly higher, resulting in a lower precision. The performance advantage of the modified algorithm increases for increased image resolution. For the (1600 x 450) image resolution, the precision performance is nearly four times as high, for the same recall.

This can be explained by inspection of the algorithm. Since the boolean maps are normalized over all activated pixels, the resulting maps have a lower mean and variance for larger images. In case of target disappearance, this causes substantial amplification of the resulting map, which results in more false positives than occur for smaller images. The opposite happens for the modified algorithm, in which the false positive rate decreases with lower variance in the resulting map. This makes outliers easier to detect, which corresponds to the larger performance improvement.

3-5 Camera Bearing Computation

The Nautis software renders the image using a virtual screen plane, which is illustrated in Figure 3-7. The camera has a position in the world, and the virtual screen plane is located in front of this camera. Everything in the world is projected on this plane, which forms the final image displayed to the user. In navigation, the bearing is the horizontal angle between the direction of an object and another object, which can be computed from the position of the the target in the image.

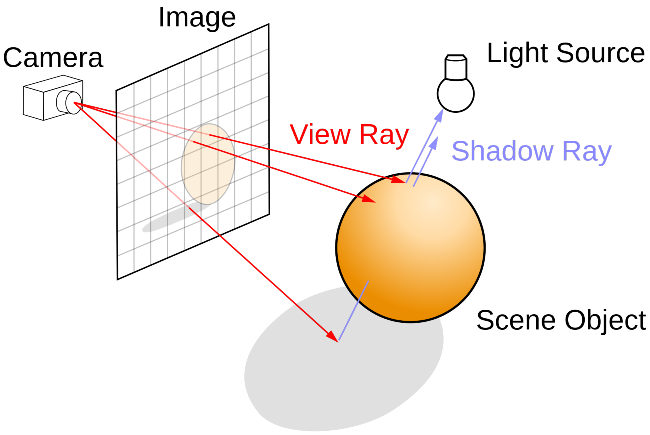


Figure 3-7: Virtual screen plane and camera

The position of the virtual screen plane creates a certain FOV. Within this FOV, the camera detection algorithm aims to detect targets. These potential targets are located at a certain position in the world, which is projected on the virtual screen plane. From the detection location on the image, we obtain the relative bearing to the Beagle. However, the projection causes the world to be depicted somewhat warped. As can be seen in Figure 3-8, the projected area from 60° to 50° is larger than the area between 50° to 40°, etcetera.

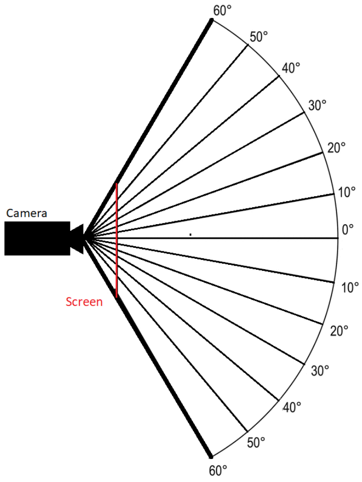


Figure 3-8: Angle projection on virtual screen plane

Therefore, we need a conversion to compute the relative bearing to the target location. This is found by defining an expression for the FOV of the camera. For better readibility, this value is denoted as β.

$$\text{FOV} = \beta = 2 \arctan\left(\frac{x_{pixels}}{2y}\right), \tag{3-15}$$

where x_{pixels} is the total amount of horizontal pixels on the image plane, y is the distance between the screen and the camera. Since we know the FOV and the total amount of horizontal pixels, we can compute y, which can then be used to compute the corresponding bearing ϕ to any given horizontal target location x_T, which is measured in pixels.

$$y = \frac{x_{pixels}}{2\tan\left(\frac{\beta}{2}\right)}. \tag{3-16}$$

This can now be substituted in the equation for the bearing of interest α

$$\begin{aligned} \alpha &= \arctan\left(\frac{x_T}{y}\right) && (3\text{-}17) \\ &= \arctan\left(\frac{2x_T \tan(\frac{\beta}{2})}{x_{pixels}}\right), && (3\text{-}18) \end{aligned}$$

where the horizontal position of the target x_T is measured from the horizontal center of the image. The found bearing is passed to the data association algorithm.

3-6 Implementation

The algorithm can be written in many different programming languages. Based on the programming experience of the author, Matlab would be an obvious choice. However, it is well known that Matlab lacks computation speed. One of the design criteria is the requirement of real-time operability. An advantage of using Matlab is that prototyping is relatively easy, because of the large amount of built-in functions. Prototyping is the process of building an initial version of the algorithm. It is used to obtain some idea of the performance. Another option is to use Matlab's counterpart Python. Python provides more or less equal prototyping convenience, while being substantially faster in terms of computation speed. A large advantage is the availability of several computer vision libraries, of which OpenCV is most popular. While this library is available for Python, the code is written in optimized C/C++. The programming languages C or C++ are well known for their computational efficiency. On the other hand, its prototyping convenience is lower. In view of the real-time operability design requirement, the choice was made to write the entire algorithm in C++.

In general, a C++ executable is computed using the CPU. However, for a large amount of small computations, which do not depend on each other, it is possible to parallelize these computations on the graphics card. CUDA is a platform developed by NVIDIA, which gives direct access to the GPU's parallel computational elements. Since the algorithms for a large part consist of pixel operations , this could give significant improvement. However, it must be noted that it does not necessarily improve computation time. Communication between the

CPU and GPU also requires time. Parallel computation using CUDA has been implemented for both algorithms. The GMM algorithm showed significant improvement in the computation speed, compared to the CPU implementation. The computation time per frame decreased with a factor 8. This algorithm allowed for a low amount of communication between the CPU and GPU, resulting in the large improvement. The SB algorithm did not show performance improvement after the CUDA implementation. It is expected that this lack of improvement is attributable to a high amount of communication between the CPU and GPU. This is necessary, since some computations can only be done on the CPU. Also, some computations are simply not computed faster on the GPU, compared to the CPU. Therefore, the GMM algorithm is used with CUDA acceleration, and the SB algorithm without.

The computer on which the tests have been conducted is an HP ZBook Studio G4, which has an Intel Core i7-7700HQ CPU @ 2.80GHz, 8GB RAM, 64-bit Windows 10 operating system, and NVIDIA Quadro M1200.

3-7 Results

In this section, the results and findings of the camera detection algorithm are presented. To compare the algorithms, some performance indicators were presented in Section 3-2. In Subsection 3-7-1, the data set used for testing is discussed, followed by the used algorithm parameters influencing performance. The computation speed performance is presented in Subsection 3-7-2. Then, the two algorithms are compared in Subsection 3-7-3.

3-7-1 Test and Algorithm Parameters

The algorithms have been tested on a created data set. These scenarios contained the following environment and camera parameters:

- **Sea state** - The sea states which are incorporated in the data set are 1 and 3. Sea state 1 contains ripples and glints on the water surface. No large wavelets, foam or wakes. Sea state 3 does contain large wavelets. Wakes and white caps are visible. On a small vessel on the open ocean, it will be difficult to maintain high velocity or board a ship.

- **Camera movement** - Of course, camera movement is directly linked to the movement of the Beagle. The settings which are considered are a static Beagle and a moving or "chasing" Beagle. For the static vessel scenario, this means the target vessel approaches the Beagle, and abruptly changes its course to one side at around 100 meters from the Beagle. The course change is representing an evasive manoeuvre. For the moving vessel scenario, the Beagle moves straight towards the target vessel, which initiates an evasive maneuver at some point. The Beagle "responds" to this by steering in the same direction, following the target vessel.

This results in a total of four data sets of about 1200 frames each. The ground truth for the data sets is hand labeled in the form of bounding boxes. The two algorithms have been tested for different algorithm parameters:

- **Image resolution** - The images are resized to a certain size before being passed to the algorithm. Since both algorithms are pixel-based, reducing the image resolution significantly influences the algorithm computation speed. The horizontal dimensions range from 600 to 1600 pixels, with 200 pixel steps in between. The vertical image dimension scales such that the image ratio remains constant.
- **GMM** - The parameters for the GMM are the learning rate and variance threshold. The learning rate determines how fast the background model adapts to the environment. The variance threshold is the squared Mahalanobis distance of the pixel intensities to the mixture components. This is chosen such that we obtain a recall ranging from 0 to 1, which is used to produce the precision-recall curves.
- **Saliency-based** - The parameters for the SB algorithm are the bin size and background threshold. The bin sizes for the boolean map generation are 10 and 25. Again, the thresholds are chosen such that we obtain the full recall span.

The bin size of the SB algorithm determines the amount of created boolean maps. A smaller bin size corresponds to more emphasis on details. Because of the relatively large number of bins, closely related pixels could still be distributed over multiple bins. This effectively results in a lot of emphasis on these pixels and thus emphasis on salient details. Large bins cause the opposite effect. It generalizes more, but is therefore more robust to noise. In low sea state, the two bin sizes perform similar. Therefore, this is not depicted. However, difference in performance can be observed for increasing sea state. Figure 3-9 shows the precision-recall curves between two bin sizes and three image resolutions, for the sea state 3 data sets.

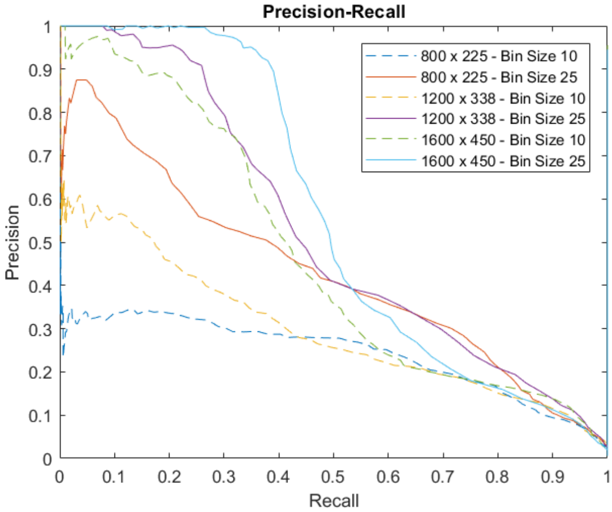


Figure 3-9: Comparison between bin size 10 and 25 for SB algorithm in sea state 3

We observe that the small bin size performs worse in terms of precision and recall, compared to the larger bin size. This is because it is likely to happen that some pixels fall just outside a certain bin containing a lot of background pixels. Such an occurrence might create a bin with a small amount of pixels, which obtains a lot of emphasis by the first normalization step in the algorithm, defined in Equation (3-5). In case of a higher sea state, more clutter and glint is present. This causes more false detections for small bin sizes than occur with larger bin sizes. Since the smaller bin size is also disadvantageous in terms of computational effort, in the following, the larger bin size is considered.

The GMM algorithm can take different values for the learning rate. However, at the moment the camera starts recording, the vessel is already in the image. Therefore, increasing the learning rate to a value corresponding to a couple of minutes shows equal performance to a learning rate corresponding to a frame history of a few seconds. A few seconds should be sufficient to cover the repetitive wave motion. Based on this assumption, a learning rate corresponding to a frame history of about 12 seconds is chosen. This corresponds to a learning rate of 0.005.

3-7-2 Computational Efficiency

One of the performance indicators presented at the beginning of this chapter is the computational efficiency of the algorithms. Since computational power is often limited for financial reasons, this is generally considered an important factor in the system design. Because of the pixel-based nature of both algorithms, image resolution is a important aspect. In the SB algorithm, the bin size used to create the boolean maps also influences computation time. In Figure 3-10, the computation time per frame versus the image resolution is presented.

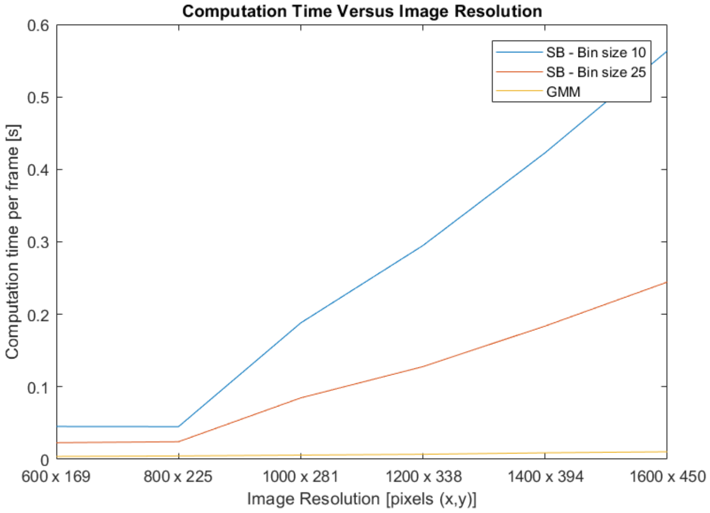


Figure 3-10: Computation speed of algorithms

Something quite noticeable is the lack of increase in the computation time of the SB algorithm from image resolution (600 x 169) to (800 x 225). In turns out that the computation power required for the increase in pixel operations is not significant to the computation power required for overhead tasks, which are independent from the pixel operations.

It can be seen that the GMM algorithm is significantly faster than the SB algorithm. The latter processes at about 40 FPS for the (800 x 225) image resolution, while the GMM processes at over 200 FPS for this image resolution. However, the video is captured at 15 FPS. This means both algorithms easily process in real-time for small image resolutions.

Since image resolutions (600 x 169) to (800 x 225) have nearly equal computational complexity. However, the latter contains more information. Therefore, the (600 x 169) image resolution will no longer be considered. The SB algorithm with the (1600 x 450) image resolution performs far from real-time on the computer processing the evaluations. However, if the performance would be substantially better compared to smaller image resolutions, it could be argued to invest in more computing power. Therefore, the (800 x 225), (1200 x 225) and (1600 x 450) images sizes will be considered in the detection performance section.

3-7-3 Algorithm Evaluation

The two algorithms have been tested on the four data sets. Using a static measuring vessel and moving measuring vessel, and in sea state 1 and 3. In Figure 3-11, the precision-recall performance can be observed.

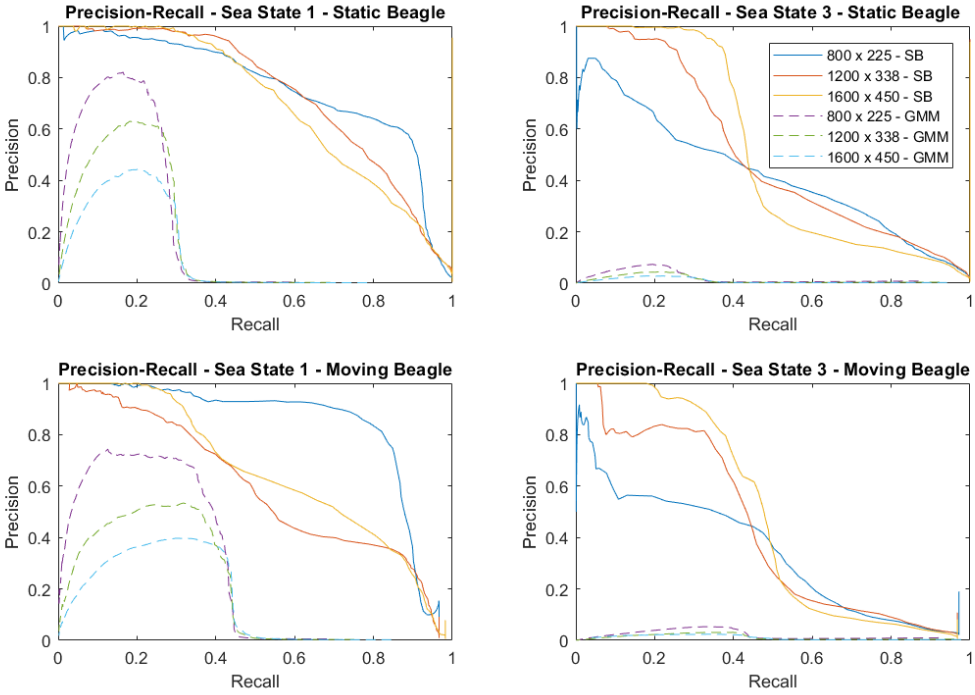


Figure 3-11: PR plots for all four scenarios

It can be seen that the GMM algorithm only obtains high precision values for low recall, in all four data sets. The first part of the scenario involves the target vessel moving straight towards the Beagle. The visually static state of the target makes that it is incorporated in the background model relatively fast. Therefore, the recall performance using a static camera is poor. It was expected that the use of a moving camera would improve performance, since the relative target position will also change. As can be seen in the moving Beagle results, this did not significantly improve the performance. We observe that the precision drops at a slightly higher recall value. However, this difference is mainly due to the difference in scenario, as the target vessel initiates its turn at a different moment. The relative movement of the target vessel in the video is too small to be consistently classified as foreground. The robustness of the GMM algorithm against clutter detections is low, which we can conclude from the performance difference between sea state 1 and 3. For the sea state 3 scenario, the GMM algorithm obtains a lot more false detections due to large wakes and sunlight glints.

The SB algorithm performs better in all four data sets, over the entire threshold span. Interesting performance differences between the different image resolutions can be observed. The smallest image resolution (800x225) shows relatively good performance for sea state 1, especially in the moving Beagle scenario. The performance for this image resolution drops significantly for the "sea state 3 - moving Beagle" scenario, while the largest image resolution (1600x450) shows better performance in this scenario. Because of the frequent partial occlusions occurring in this scenario, the small image resolution poses difficulty in detection of the small target.

In lower sea states, the largest image resolution result suffers from small sunlight glints, which result in false positives. However, in the higher sea state these glints are accompanied by wakes, which are in the same color span. This reduces the emphasis on both the glints and the wakes, decreasing the false positive rate.

Depending on environment factors, the choice can be made which image resolution is most suitable. If high precision-recall performance is required for proper tracking in all environment conditions, the image can be rescaled according to the environment conditions. The necessity for high precision, recall, and computation speed will have to be determined in combination with the tracking algorithm and the financial possibilities within the final system design.

In the way the algorithms are constructed at this point, both lack performance for targets having a color close to that of the background. The GMM scores a bit better on this indicator, since it is a local descriptor. For the SB algorithm, equal colors somewhere in the image would already be of influence. Incorporating local features could make the algorithms suitable for these kind of targets. However, it is rather unlikely that a vessel, including persons, taking along shadows and lighting, will be able to camouflage themselves good enough to be the same color as the water or background. Therefore, it will be omitted for this project. However, it would be an interesting robustification of the final algorithm in a later stage.

3-8 Discussion

Performance indicators were presented at the start of the chapter. Table 3-2 shows a qualitative comparison between the two algorithms in terms of these performance indicators.

Table 3-2: Qualitative comparison on both algorithms in terms of performance indicators

Performance Indicator	Saliency-Based Algorithm	Gaussian Mixture Model
Computation intensity	+-	++
Precision - Recall	+	-
Suitability for static and dynamic targets	+	-
Robustness to clutter	+	-
Camera movement	++	+

The computation intensity of the SB algorithm is clearly significantly higher than that of the GMM. However, for small image resolutions it is still capable of real-time computing on the test computer. By investing in more computing power, also larger image resolutions should be able to process at 15 Frames Per Second (FPS). The precision-recall performance of the SB algorithm is substantially better than the GMM, which especially lacks precision when the recall demand is high.

The GMM is not robust to targets which do not move significantly relative to the camera. Camera movement does improve this slightly, but not substantially. The SB algorithm does not have this issue, since it is a static background subtraction approach. It does not use temporal information to build a background model. This also explains its unaltered high performance for both static and dynamic Beagle scenarios. The robustness to clutter can be observed in Figure 3-11. GMM cannot deal with waves and wakes which are high. However, if the learning rate is lower, this will improve, since the wakes will be learned in the background model.

The SB algorithm outperforms the GMM in every indicator, except the computation intensity. However, the computation time performance of the SB algorithm is still acceptable in view of the real-time requirements. Therefore, the SB algorithm is used to compute the bearing which is passed to the data association algorithm.

3-9 Sensor Accuracy

Every sensor suffers from measurement noise. This measurement noise can be suppressed by the tracking filter. The measurement noise is often assumed as zero mean white Gaussian noise. This section considers the measurement noise of the radar and camera detections. This information is used later in the tracking filter.

Since the radar data is simulated, the noise variance for the range and the bearing is known. The standard deviation of the range $\sigma_r = 5m$. The bearing standard deviation $\sigma_\alpha = 0.60°$. The range rate noise is difficult to determine, since there exists no ground truth for the velocity of the vessels. For the evaluation of the algorithms, the ground truth velocity is reconstructed using an EKF which uses the measurements of the four other states. However, this reconstructed ground truth velocity will never be perfect. Therefore, it is not possible to determine the exact measurement noise characteristics for the range rate.

The camera detection accuracy is influenced by multiple factors. The range of the target is an important factor. When the target is close, it might happen that not the entire vessel is classified as foreground. Because of the small range, this leads to large bearing detection errors. As discussed in Section 3-5, the angle or bearing distribution is not uniform over the FOV of the image. Compared to the angles at the edges of the image, the angles in the middle of the image are closer together. Therefore, a target in close range, located in the middle of the image, is more prone to errors than a target in close range at the edge of the image. Another issue in close pursuit, is the classification of wakes of the target vessel as foreground, which causes bearing errors. Because of this, the algorithm is most accurate for mid-range targets. In short range, the influence of bearing errors is smaller. Therefore, no further improvement will be made to improve the short range bearing estimation of the camera.

A bounding box ground truth was created, which is used to evaluate to obtain the detected bearing error. A histogram of the resulting bearing error can be observed in Figure 3-12. A Gaussian distribution is fitted over the data. It can be seen that the data distribution somewhat resembles a Gaussian distribution. However, the obtained fit is not very accurate, because the data is not completely Gaussian distributed. The found standard deviation $\sigma_\alpha = 0.44°$.

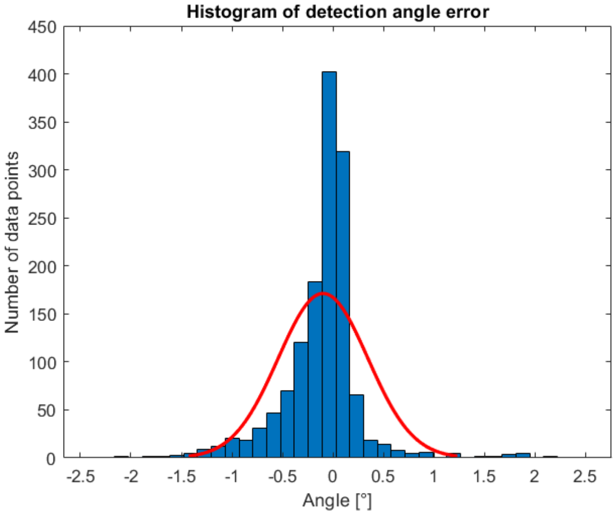


Figure 3-12: Histogram of camera detection error values

The mean error is -0.09 degrees. That the mean error is not zero, is mainly because the viewing angle of the target is not equally distributed over time. The target more often observed from one side than from the other. The wake behind the target causes a small error, which can be seen in Figure 3-13. This causes the detected bearing to be biased to one side depending on the viewing angle.

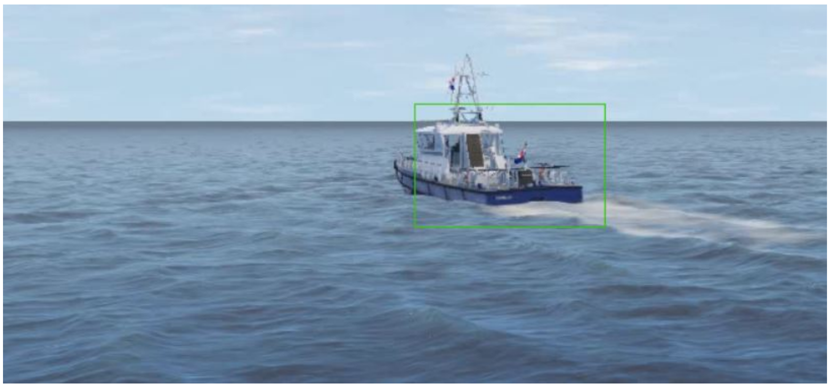


Figure 3-13: Wake of vessel causes small bearing offset

Chapter 4

Tracking

The information provided by the sensors can be used to track potential targets. Tracking methods can be very diverse, but are similar for most sensors and detection approaches. Therefore, these methods are discussed separately from the detection methods. The data association algorithm is discussed in 4-1. In Section 4-2, the coordinate system in which the Beagle and targets are tracked is discussed. In Section 4-3, some common tracking approaches are discussed. In the algorithm choice, the design requirements presented in Section 2-1 are taken into account, which is discussed in Section 4-5. The algorithm makes use of motion models to obtain the state estimate, which are discussed in Section 4-4. The chosen coordinate system requires a coordinate change, which is discussed in Section 4-6. Finally, the algorithm is evaluated on the generated data set. This is presented in Section 4-7. Which algorithm obtains the best performance is investigated. As can be seen in Figure 4-1, the tracking algorithm communicates only with the data association algorithm. The data association will also be discussed in this chapter.

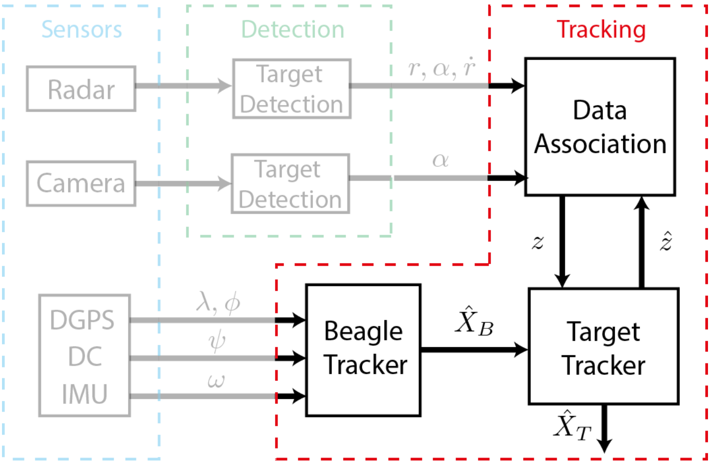


Figure 4-1: Location of tracking algorithm in system block scheme

4-1 Data Association

The data association is responsible for the correct interpretation of the collected observations, i.e. for the creation tracks, assignment of new sensor observations to the existing tracks and deletion of irrelevant tracks. The data association algorithm makes use of a gating step first, followed by the actual association step.

Gating

The data association algorithm makes use of a gating step first, followed by the actual association step. The match of a track and a measurement is not necessarily feasible. A too distant, and therefore improbable measurement can be excluded from consideration right away. This can be done using the gating procedure [27]. We define a gating region $\Gamma_k^x \in \mathbb{R}^n$, for track x at time step k. The dimension n is equal to the dimension of the measurement z_k. The validation gates are defined as hyper-ellipsoidal regions around $\hat{z}_k^x$, such that the gating probability P_g is constant for all tracks. This is done by defining the gating region Γ_k^x as

$$\Gamma_k^x(\gamma) = \{z_k \in \mathbb{R}^n \mid (z_k - \hat{z}_k^x)^\top (P_{Z_k Z_k}^x)^{-1} (z_k - \hat{z}_k^x) \leq \gamma\}, \tag{4-1}$$

where γ represents the gating threshold. The term $(z_k - \hat{z}_k^x)^\top (P_{Z_k Z_k}^x)^{-1} (z_k - \hat{z}_k^x)$ is known as the squared Mahalanobis distance [28], where $P_{Z_k Z_k}^x$ is the innovation covariance of track x. The measurements which do not fall within the gating threshold are excluded from consideration by the data association algorithm for that particular track.

Nearest Neighbor Data Association

The set of m_k^x measurements falling in the gating region is defined as:

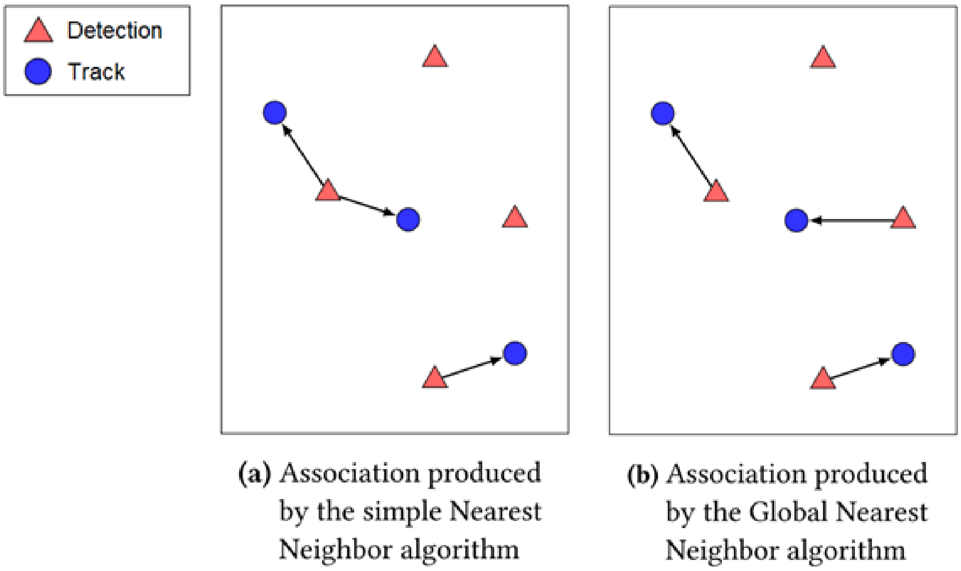
$$\mathcal{Z}_k^x := \{z_{k,1}, ... z_{k,m_k}\}. \tag{4-2}$$

It might occur that an observation falls into the gate of multiple tracks, or that multiple observations fall inside the gate of one track. The data association tries to match the correct detection with the correct track.

Nearest neighbor data association performs this pairing by matching one of the detections to a track. In the simplest form of the algorithm, the closest detection is matched to the closest track. As can be seen in Figure 4-2(a), this creates the possibility of assigning a detection to two different tracks. Therefore, there is the need for a more optimal approach.

The global nearest neighbor approach minimizes the total distance function, which is the sum of the distances for all the individual assignments [29]. This sum is computed with the use of the cost matrix C, which is defined as

$$C_{ij} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1m} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n1} & c_{n1} & \cdots & c_{nm} \end{bmatrix}. \tag{4-3}$$



(a) Association produced by the simple Nearest Neighbor algorithm

(b) Association produced by the Global Nearest Neighbor algorithm

Figure 4-2: Simple nearest neighbor and global nearest neighbor

The elements of the cost matrix c_{ij} have the following values:

$$c_{ij} = \begin{cases} 10000 & \text{if measurement j is not in the gate of track i} \\ d_{ij} & \text{if measurement j is in the gate of track i,} \end{cases} \tag{4-4}$$

where d_{ij} is the Mahalanobis distance between track i and measurement j. Since we are dealing with a relatively low amount of tracks and detections, the optimum can be found by simple enumeration. The cost function computation has to satisfy the constraint that a detection cannot be assigned to multiple tracks. When a camera and radar detection are obtained in the same time step, the radar detection has priority because of the sample rate difference. The radar detection is less accurate than the camera bearing, but it does not suffer from clutter as much as the camera. Also, the radar detection is the only sensor providing a range measurement. Therefore, it is important that this gets updated. If we match the camera bearing with the radar range detection, persistent clutter might make the track prediction drift off the target.

Camera Association

In case of multiple targets, it can occur that targets are visually crossing in front of the Beagle. If this happens, the more distant target will disappear behind the closer one. Therefore, only one camera detection can be obtained. Small delays, noise in the measurement, or error in the prediction of both vessels cause the wrong measurement assignment to the wrong track. To overcome this, it is proposed to update only the closer vessel when the distant vessel is within half of its camera gating threshold. When the Mahalanobis distance between the vessels is larger than half of the camera gating threshold, both vessels are updated by the global nearest neighbor algorithm.

Track Initiation and Termination

False detections could occur with any kind of sensor and corresponding algorithm. Ocean waves, or even simply sensor noise, could be mistaken as a potential threat. However, these false alarms are usually not persistent over time. To deal with these false detections, tracks can be initiated and terminated. The camera detection algorithm suffers from clutter more than the radar detection algorithm. Also, it only provides the bearing of a target, which is not sufficient for a complete position determination. Therefore, tracks are only initiated by a radar detection. After initiation, the track can be updated by either the radar or the camera. If no radar detection is matched to the track for too long, the track is terminated.

If a track is initiated on a radar clutter detection, the camera detections which are matched to the track are also likely to be clutter detections. Due to the significantly lower update frequency of the radar, about 90 camera frames have been processed in the time of only three radar updates. The chance of a camera clutter detection falling inside the gating region within this time is significant. This increases the possibility of maintaining clutter induced tracks. Therefore, camera detections are not included in the termination threshold.

4-2 Coordinate System

The Beagle employs sensors to obtain a perception of its surroundings. Furthermore, the movement of the vessel itself is tracked, using different sensors. These sensors provide the measurements in a certain coordinate frame, although this might not be the most ideal coordinate frame for the application. The coordinate system in which the kinematic states of both the Beagle and the target are described, is discussed in this section.

Beagle Coordinate System

The Beagle makes use of DGPS, DC and IMU data, which provide the location, heading and yaw rate respectively. The DGPS data is supplied in latitude-longitude format. The latitude and longitude describe the location on the spherical Earth, which is shown in Figure 4-3. However, since we are interested in relatively small displacements, the curvature of the Earth can be neglected. Therefore, we can convert these coordinates to Cartesian coordinates on a plane, called an equirectangular projection [30]. The latitude and longitude are structured as degrees and minutes. Each degree can be divided into 60 minutes. To convert minutes to decimal degrees, we divide the minutes by 60. The Cartesian coordinates x and y are now computed as:

$$\begin{aligned} x &= R\lambda\cos(\phi_0), \\ y &= R\phi, \end{aligned} \tag{4-5}$$

where R is the radius of the Earth at the location of interest, ϕ is the latitude, and λ is the longitude, both in decimal degrees. The ratio between latitude and longitude is not 1:1. Therefore, the aspect ratio $\cos(\phi_0)$ is used, in which ϕ_0 is a latitude value close to the center of the 2D map. The initial position of the Beagle is taken as the origin of the coordinate system. The heading ψ is measured clockwise, starting from the y-axis. The yaw rate ω is also measured clockwise.

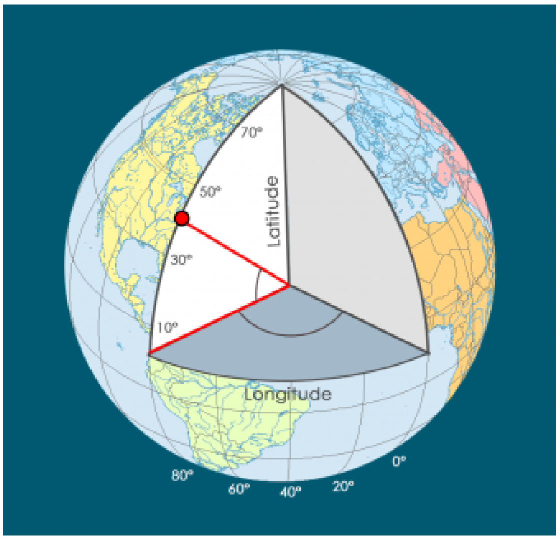


Figure 4-3: Longitude and latitude

Target Coordinate System

The sensors providing information about the environment are a FMCW radar system and a visible-light camera. The radar system obtains potential target detections in polar coordinates. This gives the measurement vector $z = \begin{bmatrix} r, & \alpha, & \dot{r} \end{bmatrix}$, where r is the range, α the bearing relative to the Beagle heading ψ_B, and $\dot{r}$ is the radial range rate. The camera system obtains potential target detections as α, the bearing relative to the Beagle heading. The coordinate frame in which the surroundings of the Beagle are described is discussed in the following paragraphs.

Beagle Coordinate System - It is possible to describe the target motion in the same coordinate frame as the measurements, being the polar coordinates relative to the beagle position and heading. The origin is taken as the center of the Beagle. However, the beagle position and heading are also varying. This makes the dynamics of states described in this way highly nonlinear. In automated driving, it is general practice to track targets relative to the own vehicle position. Under the assumption that the observed targets are straight ahead and moving nearly in the same direction, these systems work well. When movements of targets have to be tracked in all possible directions, and all around the Beagle, this is insufficient.In [31], the authors propose the use of a global motion tracking. It is shown that the proposed approach outperforms the relative motion model in the case of targets moving in every direction relative to the ego vehicle. Motivated by this, in this thesis we will also employ global target tracking.

Global Polar Coordinates - The second option would be to convert the detections to polar coordinates relative to a stationary reference point, as illustrated in Figure 4-5. This decreases the nonlinearity of the system, compared to the previous paragraph. However, intuitive interpretation of the states is still difficult. The reference point can be any point in the map. However, it is convenient to use the initial position of the Beagle. As discussed in section

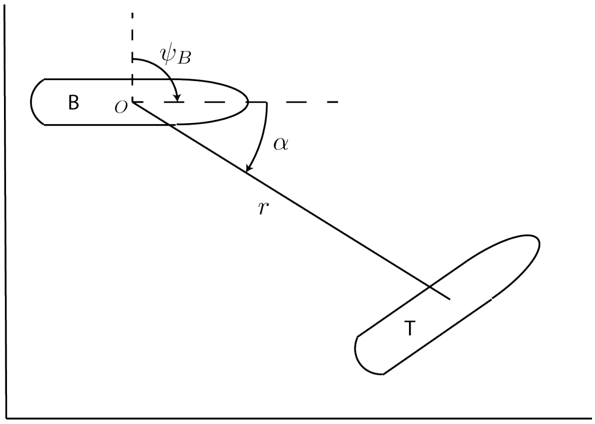


Figure 4-4: Tracking in the Beagle coordinate system

4-2, the initial position of the Beagle is the origin. Therefore, we simply have to add the current position of the Beagle in the computation of the new coordinates. To obtain the polar coordinates with respect to the reference point, both polar coordinates provided by the sensors have to be used, i.e. the global range of the converted coordinates depend on both the relative range and relative bearing, and the same for the global angle.

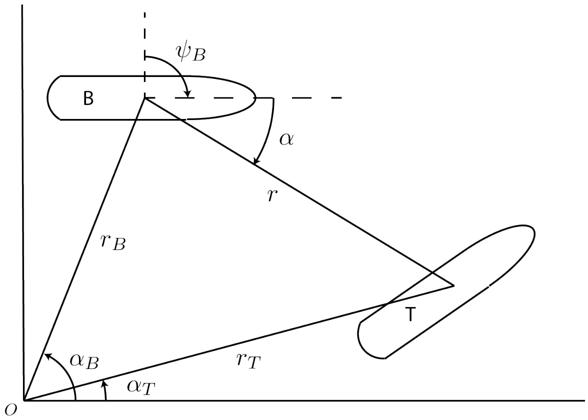


Figure 4-5: Tracking in global polar coordinates

Global Cartesian Coordinates - A third option is to convert the relative polar coordinates to global Cartesian coordinates, which is shown in Figure 4-6. The dynamic model has similar complexity as the polar coordinates, but the states are easier to interpret. If the initial position of the Beagle is used as the origin of the map, we simply have to add the current Beagle position (x_B, y_B) to the relative position to obtain the target position in Cartesian coordinates (x, y).

The polar coordinates with static reference point, and the Cartesian coordinate approach both require conversion to obtain the new coordinates. The complexity of the resulting dynamic model is comparable. Since the Cartesian coordinates are more intuitive to interpret directly, this is the one that will be used in the tracking system.

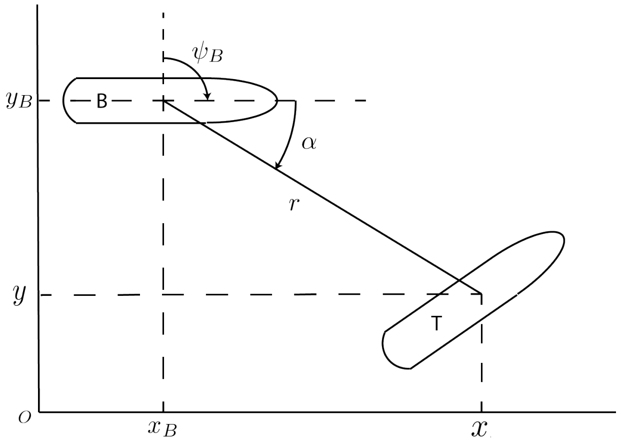


Figure 4-6: Tracking in global Cartesian coordinates

4-3 Dynamic State Estimation

Every sensor system contains a certain factor of noise. Therefore, the obtained measurements may differ from the true or expected values. The aim of a dynamic state estimator is the estimation of the true value of the state of the system, based on the provided measurements. The filters performing the state estimation aim to minimize the noise effects. Some commonly used methods for dynamic state estimation will be discussed in this section.

4-3-1 Basic Tracking

The easiest form of visually tracking a target is by updating the bounding box of detected targets within a reasonable motion threshold. An adaptive bounding box can be used, where the new bounding box is an adaptation of the bounding box of the previous frame, within specified ranges of adaptivity [13]. In the case of the open ocean, this range can only be referenced according to the target and its approximated velocity range, which is difficult due to possible appearance changes of the target. A possibility might be to use the radial velocity and distance of an object, obtained by a radar image and use this to estimate the angular velocity by fusing the radar image and a camera image. This could decrease the error due to measurement noise. In [32], a track-before-detect approach based on dynamic programming is proposed. Track-before-detect methods use multiple frames of data to obtain a detection and estimated track at the same time. This is especially useful if the Signal to Noise Ratio (SNR) is low, which is the case when the target is at the edge of the maximum detection distance. The dynamic programming approach performs an exhaustive search of all possible state sequences in an efficient manner. In [33], use is made of spatio-temporal integrated blob strength, exploitation of stable image regions [34], or a salient image point tracking based algorithm [35]. However, most of these approaches do not allow for straightforward fusion with other sensors. As discussed in Chapter 2, a sensor fusion between radar and camera is implemented. Also, it might be difficult to accurately estimate target states. Therefore, these basic tracking approaches are not considered applicable for this thesis.

4-3-2 Kalman Filter

One of the most used tracking approaches is the Kalman filter, which is a state estimator for dynamical systems. The most standard form of the Kalman filter is the linear Kalman filter. As the name suspects, the linear Kalman filter uses a linear system description. Over the years, many variations on the Kalman filter have been proposed. A well known variation is the Extended Kalman Filter (EKF), along many other variations. However, only these two are discussed in this section. In the next two subsections, the linear Kalman filter and EKF are briefly explained.

Linear Kalman Filter

The linear Kalman filter uses a system model and measurement model which are given as the linear equations

$$x_{k+1} = Ax_k + w_k, \tag{4-6}$$

and

$$z_k = Cx_k + v_k, \tag{4-7}$$

respectively, where A and C are the system matrices, and w_k and v_k the noise components. The covariances of the noise components are assumed stationary over time, and given by

$$\begin{aligned} Q &= E[w_k w_k^T], \\ R &= E[v_k v_k^T]. \end{aligned} \tag{4-8}$$

The *a posteriori* error covariance matrix P_k is defined as

$$P_k = E[e_k e_k^T]. \tag{4-9}$$

The state x_k and error covariance P_k are predicted using

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k}, \\ P_{k+1|k} &= AP_kA^T + Q. \end{aligned} \tag{4-10}$$

The filter is updated using the measurement z_k. This is done in the update step of the Kalman filter. The Kalman gain K_k for time step k is computed, which is used to update the state and error covariance.

$$\begin{aligned} K_k &= P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}, \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}), \\ P_k &= (I - K_kC)P_{k|k-1}. \end{aligned} \tag{4-11}$$

Extended Kalman Filter

When the system is nonlinear, the linear Kalman filter could be insufficient. The filter uses a single linear approximation of the system, which could result in a large estimation error in case of a nonlinear system. The EKF uses an approximation of the linearization of the system

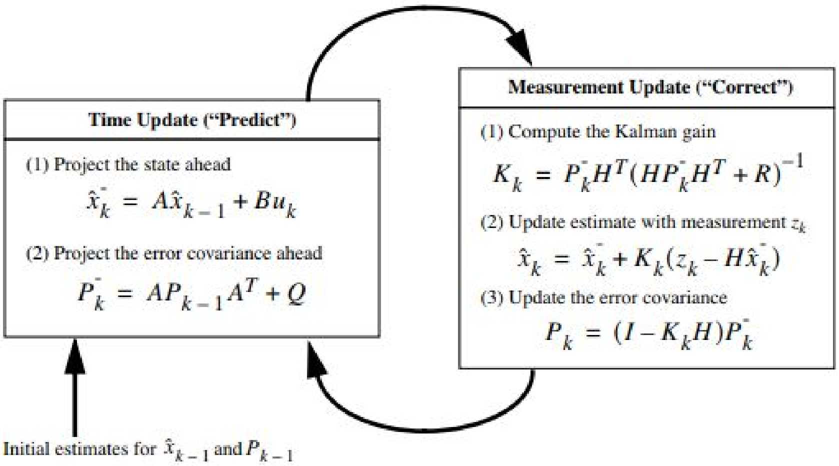

Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
(2) Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Figure 4-7: Linear Kalman filter operation [3]

in the form of a Taylor series expansion around the latest state estimate. This is followed by application of the Kalman linear recursive algorithm. The difference equations of the system model are now described as

$$x_{k+1} = f(x_k, w_k), \tag{4-12}$$

and the measurement model

$$z_k = h(x_k, v_k). \tag{4-13}$$

The nonlinear function $f(x_k, w_k)$ relates the state in the current time step to the state in the next time step. The nonlinear function $h(x_k, v_k)$ relates the state x_k to the measurement z_k. To use the same framework as the linear Kalman filter, the dynamics are linearized around the current state estimate. This linearization is done by taking the Taylor series expansion of $f(x_k, w_k)$ around the value $\hat{x}_k$, and drop all the higher order terms. This leaves us with constant and linear terms. This can be defined by the Jacobian J_F of $f(x_k, w_k)$.

$$J_{F,i,j} = \frac{\delta f_i}{\delta x_j}(\hat{x}_{k|k-1}), \tag{4-14}$$

and the Jacobian J_H for the measurement model $h(x_k, v_k)$

$$J_{H,i,j} = \frac{\delta h_i}{\delta x_j}(\hat{x}_{k|k-1}). \tag{4-15}$$

The EKF update equations are now defined as

$$\begin{aligned} \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, w_k) \\ P_{k+1|k} &= J_F P_{k|k} J_F^T + Q \end{aligned} \tag{4-16}$$

As with the linear Kalman filter, the update equations in the next time step project the state and covariance estimates from the current time step to the next time step. The Jacobians J_F and J_H are computed at every time step. The measurement update is done by computing

$$\begin{aligned} K_k &= P_{k|k-1} J_H^T (J_H P_{k|k-1} J_H^T + R_k)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1})) \\ P_{k|k} &= (I - K_k J_H) P_{k|k-1} \end{aligned} \tag{4-17}$$

A drawback of this algorithm is that it is more computationally intensive than the linear Kalman filter, because the Jacobian has to be computed in every step. However, compared to the camera detection algorithm, the additional computational complexity is likely to be negligible. Of course, we are still working with a linearization around a working point. Therefore, when the system is highly nonlinear, this approximation can be inaccurate.

4-3-3 Particle Filter

Kalman filters are optimal when estimating the states of a linear system and measurement models with zero mean Gaussian noise. However, when the model has highly nonlinear, non-Gaussian or multi-model characteristics, the Kalman approximation can be inaccurate. Non-linear or non-Gaussian filtering can be done using a deterministic method, or a stochastic method, for instance using a Kalman filter. Another approach to perform this filtering is the particle filter method [36].

Consider the state of the system x_k at time instant k. The *a posteriori* Probability Density Function (PDF) of x_k is denoted by $f(x_k|y_{(1:k)})$. This is approximated by a discrete, random measure, which consists of M so-called particles and weights:

$$\mathcal{X}_k = \{x_k^{(m)}, w_k^{(m)}\}_{m=1}^{M}, \tag{4-18}$$

where $x_k^{(m)}$ and $w_k^{(m)}$ represent the m-th particle and weight respectively. These particles are Monte Carlo samples of the states. The weights are non-negative values which sum up to one. They can be interpreted as how likely a certain particle is. The PDF can be estimated using these weights:

$$f(x_k|y_{(1:k)}) \approx \sum_{m=1}^{M} w_k^{(m)} \delta(x_k - x_k^{(m)}), \tag{4-19}$$

where $\delta(\cdot)$ denotes the Dirac delta function. This approximation can be used to compute the expectation $E[h(x_k)]$ of the random process x_k

$$\begin{aligned} E[h(x_k)] &= \int h(x_k) f(x_k|y_{(1:k)}) dx_k \\ &\approx \sum_{m=1}^{M} w^{(m)} h(x_k^{(m)}), \end{aligned} \tag{4-20}$$

where $h(x_k)$ is a function of x_k. If a large number of samples are taken from $f(x_k|y_{(1:k)})$, $E(h(x_k))$ can be estimated with arbitrary accuracy. An issue that arises in practice is that this PDF is not available for sampling directly. A solution to this problem is importance sampling. It is based on the idea of using another function to sample the particles. This function is called the proposal distribution, represented by $\pi(x_k)$. A possible issue using the large amount of particles, is that it is computationally expensive. On the other hand, the method should be able to handle highly nonlinear, non-Gaussian problems.

The downside of this tracking method is that it only processes in real-time on relatively low resolution images, around the image resolution (300 x 500) [37][38]. If the image resolution is decreased to this resolution, relatively distant targets cannot be discerned as foreground anymore, which is checked by observation. This makes the method not very attractive for our application.

4-3-4 Interacting Multiple Model

The interacting multiple model algorithm is a method which allows to combine state hypotheses from multiple filters, with the aim of obtaining better state estimates of targets with changing dynamics [39]. The block scheme for the framework is shown in Figure 4-8. The state estimates resulting from different models are mixed prior to state update with the use of conditional model probabilities $\tilde{\mu}^{i|j}$. These conditional model probabilities are computed using the model probabilities from the previous update. Also, a state switching matrix is used, which is generally selected *a priori.* A likelihood Λ^i is computed for each filter after the state update. This likelihood, model probabilities and prior state switching matrix are used to update the conditional model probabilities.

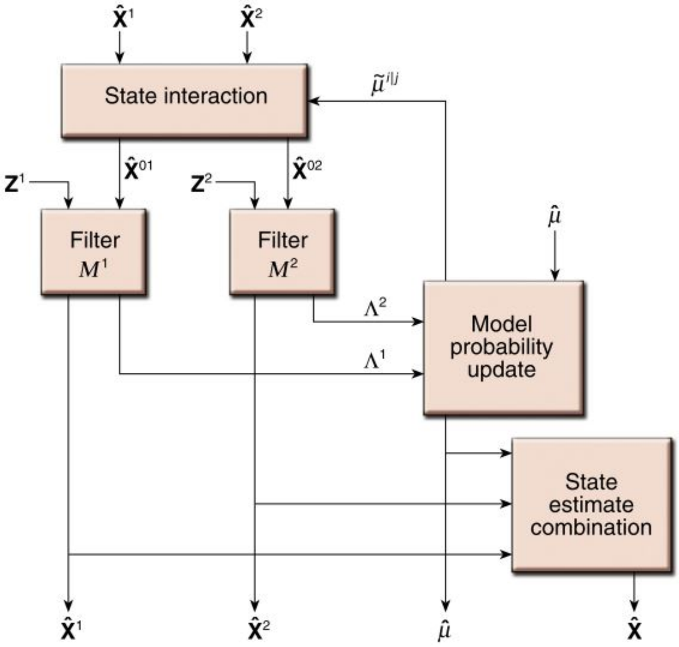


Figure 4-8: IMM framework

The state estimates from the different models are mixed using the conditional model probabilities. The mixed state $\hat{X}_k^{0j}$ and mixed covariance $\hat{P}_k^{0j}$ for filter j at time step k is defined as

$$\hat{X}_k^{0j} = \sum_{i=1}^{N} \hat{X}_k^i \tilde{\mu}_k^{i|j}, \tag{4-21}$$

$$\hat{P}_k^{0j} = \sum_{i=1}^{N} \tilde{\mu}_k^{i|j} \Big[\hat{P}_k^i + \Big(\hat{X}_k^i - \hat{X}_k^{0j}\Big)\Big(\hat{X}_k^i - \hat{X}_k^{0j}\Big)^T\Big], \tag{4-22}$$

where $\hat{X}_k^i$ is the state vector and $\hat{P}_k^i$ the error covariance of model i at time step k. The conditional model probability $\tilde{\mu}^{i|j}$ is computed as

$$\tilde{\mu}^{i|j} = \frac{1}{\bar{\Psi}_k^j} p^{ij} \hat{\mu}_{k-1}^i, \tag{4-23}$$

with the normalization factor $\bar{\Psi}^j$

$$\bar{\Psi}^j_k = \sum_{i=1}^{N} p^{ij} \hat{\mu}^i_{k-1}, \tag{4-24}$$

where p^{ij} is the (i, j) element of the state switching matrix. This defines the *a priori* probability for switching from model i to j. The likelihood Λ^j_k of model j is given by

$$\Lambda^j_k = \frac{1}{\sqrt{|2\pi \tilde{S}^j|}} \exp\Big[-0.5(Z^j)^T(\tilde{S}^j)^{-1}(Z^j)\Big], \tag{4-25}$$

with innovation Z^j and innovation covariance $\tilde{S}^j$, which are defined as

$$\begin{aligned} Z^j &= z_k - h(X^j_k) \\ \tilde{S}^j &= C^j \tilde{P}^{0j}_k (C^j)^T + R. \end{aligned} \tag{4-26}$$

In case of an EKF, the innovation covariance is computed using the measurement model Jacobian J^j_H, instead of the observation matrix C^j. Finally, the model probabilities $\hat{\mu}^j_k$ are updated after all filter models have been updated, which is done by computing

$$\begin{aligned} \hat{\mu}^j_k &= \frac{1}{c} \Lambda^j_k \bar{\Psi}^j_k, \\ c &= \sum_{i=1}^{N} \Lambda^i_k \bar{\Psi}^j_k. \end{aligned} \tag{4-27}$$

Again, the term c is a normalization constant. Finally, the target state estimate $\mathbf{X}_{T,k}$ is computed as

$$\mathbf{X}_{T,k} = \sum_{i=1}^{N} \hat{X}^i_k \hat{\mu}^i_k \tag{4-28}$$

4-4 Motion Models

Multiple dynamic models can be defined to describe the target motion. In [40], the authors discuss commonly used motion models. Some of these models are discussed in Subsection 4-4-1. In Subsection 4-4-2, a motion model is presented based on ship dynamics.

4-4-1 Standard Motion Models

Constant Velocity Model - As the name suggests, the velocity and heading are assumed constant. This is the model which will yield the most accurate estimates for non-manoeuvring and non-accelerating targets. The state vector is defined as

$$\mathbf{X} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T. \tag{4-29}$$

The corresponding state update equations are defined as

$$\mathbf{X}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X}_k + \eta_k, \tag{4-30}$$

where η_k is the process noise and Δt represents the time in between updates. The presented dynamic model can be operated by a linear Kalman filter. This motion model uses Cartesian velocity. Essentially, the two velocity states $\dot{x}$ and $\dot{y}$ represent both the velocity and the heading. In case of manoeuvring targets, this is difficult to incorporate in the process noise covariance. It is likely that the total target velocity will not fluctuate a lot, assuming the target will try to move near their top speed in case of a pursuit. However, when the target starts turning, the two velocity states will change significantly, while the total target velocity might be maintained. In [41], the author proposes the use of mixed Cartesian position and polar velocity models, which outperforms the use of the Cartesian velocity models when applied in an Interacting Multiple Model (IMM). The polar velocity decouples the velocity and the heading, which gives the following state vector:

$$\mathbf{X} = \begin{bmatrix} x & y & \psi & v \end{bmatrix}^T, \tag{4-31}$$

where ψ represents the target heading, and v is the total velocity. This gives the state update equations

$$\mathbf{X}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & \Delta t \sin(\psi) \\ 0 & 1 & 0 & \Delta t \cos(\psi) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X}_k + \eta_k. \tag{4-32}$$

The state update of Equations 4-30 and 4-32 essentially perform the same transition. However, the latter equation has decoupled the heading and the velocity, making it more suitable for our application. The second model does have a drawback. The state transition of the position states are nonlinear because of the sine and cosine functions. Therefore, this model cannot be used by a linear Kalman filter. However, it is possible to use an EKF.

Coordinated Turn Model - The coordinated turn model makes the assumption of a constant yaw rate ω. This means the vessel is moving along a circular path. The state transition matrices of the coordinated turn model can also be defined using Cartesian velocity or polar velocity. Again, the state vector for Cartesian velocity is defined as

$$\mathbf{X} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T. \tag{4-33}$$

The state transition matrix is defined with the assumption that a constant velocity turn manoeuvre is performed. This results in the state update equations

$$\mathbf{X}_{k+1} = \begin{bmatrix} 1 & \omega^{-1}\sin(\omega\Delta t) & 0 & -\omega^{-1}(1-\cos(\omega\Delta t)) \\ 0 & \cos(\omega\Delta t) & 0 & -\sin(\omega\Delta t) \\ 0 & \omega^{-1}(1-\cos(\omega\Delta t)) & 1 & \omega^{-1}\sin(\omega\Delta t) \\ 0 & \sin(\omega\Delta t) & 0 & \cos(\omega\Delta t) \end{bmatrix} \mathbf{X}_k + \eta_k, \tag{4-34}$$

where ω is the turning rate or yaw rate. Another option of writing the same equation by using the mixed Cartesian position and polar velocity, with the corresponding state vector

$$\mathbf{X} = \begin{bmatrix} x & y & \psi & v \end{bmatrix}^T. \tag{4-35}$$

This gives the state update equations

$$\mathbf{X}_{k+1} = \begin{bmatrix} x_k + \frac{v_k}{\omega_k}(-\cos(\omega_k \Delta t + \psi_k) + \cos(\psi_k)) \\ y_k + \frac{v_k}{\omega_k}(\sin(\omega_k \Delta t + \psi_k) - \sin(\psi_k)) \\ \psi_k + \omega_k \Delta t \\ v_k \end{bmatrix} + \eta_k, \tag{4-36}$$

where η_k is the process noise. The notation of the state transition is different from Equation (4-34), because of the extra influence of $\omega_k \Delta t$ in the computation for ψ_k. Therefore, the equations cannot be written in the same linear format. However, the two equations still perform the same action. Again, the mixed Cartesian position and polar velocity model should be employed using, for instance, an EKF.

Constant Turn Rate - Constant Velocity - In case the turn rate is unknown, we can incorporate it as a state in the dynamic model. However, this inherently causes the system to be nonlinear, which makes the use of the linear Kalman filter impractical. Therefore, only the mixed Cartesian position and polar velocity model is considered, since it is expected to perform better in view of the results of [41]. This gives the state vector

$$\mathbf{X} = \begin{bmatrix} x & y & \psi & v & \omega \end{bmatrix}, \tag{4-37}$$

and the state transition equations

$$\mathbf{X}_{k+1} = \begin{bmatrix} x_k + \frac{v_k}{\omega_k}(-\cos(\omega_k \Delta t + \psi_k) + \cos(\psi_k)) \\ y_k + \frac{v_k}{\omega_k}(\sin(\omega_k \Delta t + \psi_k) - \sin(\psi_k)) \\ \psi_k + \omega_k \Delta t \\ v_k \\ \omega_k \end{bmatrix} + \eta_k. \tag{4-38}$$

Note that in the position update equations, we divide by ω_k. Therefore, when this value approaches zero, we run into numerical issues. To overcome this, the state update equation for very small yaw rates is defined as

$$\mathbf{X}_{k+1} = \begin{bmatrix} x_k + v_k \sin(\psi_k) \\ x_k + v_k \cos(\psi_k) \\ \psi_k + \omega_k \Delta t \\ v_k \\ \omega_k \end{bmatrix} + \eta_k, \tag{4-39}$$

which is very similar to Equation (4-32), but with consideration of the yaw rate. All of the discussed models can be incorporated in the different filter models of the IMM.

Equation (4-38) is similar to the one in [42], but slightly different due to a difference in the setup of the measurements. The heading of the vessel is measured clockwise, with northern direction zero. This is depicted in Figure 4-9. For a certain point in time, the curved trajectory of the vessel can be described as a circle with a radius

$$r = \left| \frac{v}{\omega} \right|, \tag{4-40}$$

which follows from the relationship between the translational and rotational velocities v and ω. The angle between the radius with the horizontal, β, is computed as

$$\begin{aligned} \alpha &= \frac{1}{2}\pi - \psi, \\ \beta &= \pi - \frac{1}{2}\pi - \alpha = \psi. \end{aligned} \tag{4-41}$$

The center of the circle is computed as

$$\begin{aligned} x_c &= x + \frac{v}{\omega}\cos(\psi), \\ y_c &= y - \frac{v}{\omega}\sin(\psi). \end{aligned} \tag{4-42}$$

The state update equation is now computed as

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} &= \begin{bmatrix} x_c - \frac{v_k}{\omega_k}\cos(\psi_k + \omega_k \Delta t) \\ y_c + \frac{v_k}{\omega_k}\sin(\psi_k + \omega_k \Delta t) \end{bmatrix} \\ &= \begin{bmatrix} x + \frac{v_k}{\omega_k}(\cos(\psi_k) - \cos(\psi_k + \omega_k \Delta t)) \\ y + \frac{v_k}{\omega_k}(-\sin(\psi_k) + \sin(\psi_k + \omega_k \Delta t)) \end{bmatrix} \end{aligned} \tag{4-43}$$

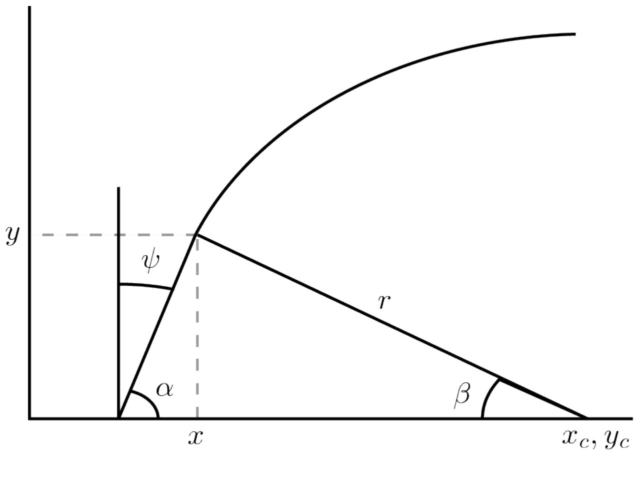


Figure 4-9: Motion along curved trajectory by vessel

4-4-2 Ship Model

The models described above make assumptions purely based on the manoeuvring actions of the target. No assumptions or observations about the physical appearance of the target have been made or included. In automotive applications, a common approach for incorporating physical information, is the bicycle model [43]. This approach incorporates the actual kinematics of a front-wheel-steered vehicle. The two front wheels are modeled as one wheel, located at the middle axis of the vehicle. This is illustrated in Figure 4-10. For the rear wheels, the same is done. In [4], the vehicle equations are defined as

$$\begin{aligned}\dot{x} &= v\cos(\psi+\beta)\\ \dot{y} &= v\sin(\psi+\beta)\\ \dot{\psi} &= \frac{v}{l_r}\sin(\beta)\\ \dot{v} &= a\\ \beta &= \arctan\Big(\frac{l_r}{l_f+l_r}\tan\delta_f\Big),\end{aligned} \tag{4-44}$$

where l_r and l_f represent the distance from the center of mass of the vehicle to the front and rear axles respectively, δ_f is the front steering angle, and β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the vehicle. In [44], the authors have incorporated intrinsic parameters of the preceding vehicle. The mass, cornering stiffness, and inertia have been added to the model. These variables were obtained using vehicle type recognition techniques.

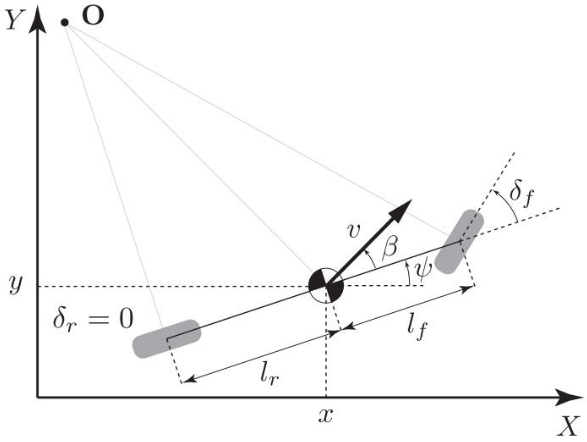


Figure 4-10: Kinematic bicycle model [4]

For the tracking of relatively small vessels, no literature was found on the use of advanced motion models. However, work was done on the development of ship manoeuvring models. It is investigated whether some of these models could be applied in this thesis. The body-fixed coordinate system of a vessel are depicted in Figure 4-11 [5]. If the assumption is made that the vessel is symmetrical about its longitudinal plane, the motion equations according to Newton's second law are defined as

$$\begin{aligned}F_X &= m(\dot{u} - v\omega - x_G\omega^2),\\ F_Y &= m(\dot{v} + u\omega + x_G\omega^2),\\ N &= I_z\dot{\omega} + m x_G(\dot{v} + u\omega),\end{aligned} \tag{4-45}$$

where F_X, F_Y, N are external forces and moment which act on the middle of the vessel in direction of the x-axis, y-axis and around z-axis respectively, u and v are the velocities of the vessel in x direction and y direction respectively, ω is the yaw rate, I_z is the inertia and m

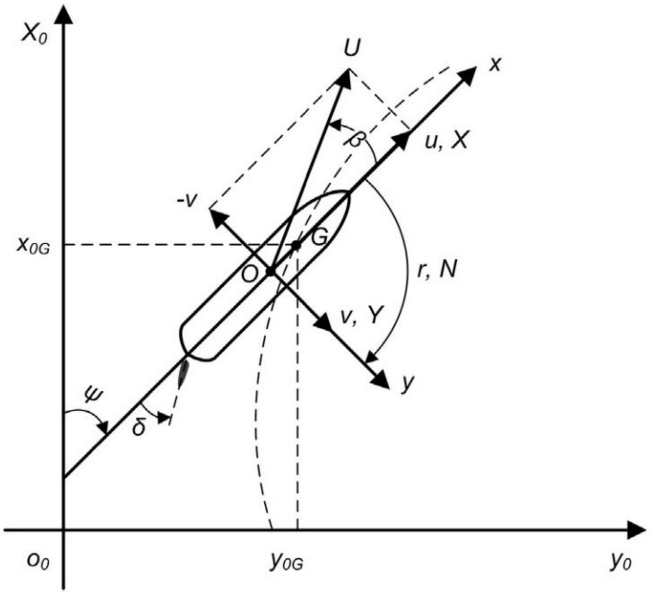


Figure 4-11: Body-fixed coordinate system of vessel [5]

is the mass of the vessel. The external forces F_X and F_Y, and the moment N are required to solve the motion equations 4-45. These forces and moment are the result of the ship hydrodynamics and environmental disturbances. For the modeling of these forces, multiple methods exist.

One of the options is the approach presented by Abkowitz [45]. The method expresses the hydrodynamic forces in terms of the kinematics and the rudder angle. The method uses a Taylor series expansion to describe an expression for these forces. Another option is the use of the so-called MMG model, which splits the forces and moment into three subgroups: the hull, propeller, and rudder. For both options, hydrodynamic coefficient estimation is required. Often, this is done using either model tests or using numerical methods [5]. Both are not possible in case of unknown target tracking.

Another simplified description of the ship motions, is an approach called Nomoto's KT model [46]. The author presented a first order transfer function

$$T\dot{\omega} + \omega = K\delta, \tag{4-46}$$

where T represents the course stability coefficient, and K represents the turning ability coefficient. This method simplifies the modeling of the forces. However, these coefficients still have to be estimated, for instance using model tests.

To avoid these difficulties, it is desired to use an approximation based on terms which can be estimated by observation. Such an expression was proposed by Lewandowski in [47]. Partly based on the full-scale data presented in [48], the author derived an approximation for the Steady Turning Diameter (STD)

$$\frac{\text{STD}}{L} \approx \left[1.7 + 0.0222 F_{\nabla} \left(\tfrac{L}{\nabla^{1/3}}\right)^{2.85}\right] \left(\frac{30}{\delta}\right), \tag{4-47}$$

for

$$\begin{aligned} 0.3 < &F_\nabla < 4, \\ 4.5 \leq L/&\nabla^{1/3} \leq 7, \end{aligned} \tag{4-48}$$

with the volumetric Froude number

$$F_\nabla = \frac{v}{\sqrt{g\nabla^{1/3}}}. \tag{4-49}$$

Here, ∇ represents the under water volume, L is the length of the vessel on the waterline, and δ is the rudder angle. In this thesis, it is proposed to incorporate the expression proposed by Lewandowski into a motion model. The STD is equal to two times the turning radius R_k, for time step k. As stated previously in Equation (4-40), the turn radius can be described by the fraction of the velocity and the yaw rate, based on the relationship between the translational and rotational velocities. If we substitute Equation (4-40) in Equation (4-47), and rewrite the equation, we obtain

$$R_k = \frac{v_k}{\omega_k} \approx \frac{1}{2}L\left[1.7 + 0.0222F_\nabla\left(\tfrac{L}{\nabla^{1/3}}\right)^{2.85}\right]\left(\tfrac{30}{\delta}\right). \tag{4-50}$$

Rewriting the equation, we obtain an expression for the yaw rate ω, based on the velocity v, the rudder angle δ, and the physical quantities L and ∇.

$$\omega_k \approx v_k\left(\frac{1}{2}L\left[1.7 + 0.0222F_\nabla\left(\tfrac{L}{\nabla^{1/3}}\right)^{2.85}\right]\left(\tfrac{30}{\delta}\right)\right)^{-1}. \tag{4-51}$$

This expression can be substituted in the coordinated turn motion model

$$\mathbf{X}_{k+1} = \begin{bmatrix} x_k + \frac{v_k}{\omega_k}(-\cos(\omega_k\Delta t + \psi_k) + \cos(\psi_k)) \\ y_k + \frac{v_k}{\omega_k}(\sin(\omega_k\Delta t + \psi_k) - \sin(\psi_k)) \\ \psi_k + \omega_k\Delta t \\ v_k \end{bmatrix} + \eta_k, \tag{4-52}$$

for a certain constant rudder angle δ. Because of the relation between the yaw rate, velocity and rudder angle, we now have a natural and dynamic limit on the turn radius, instead of a maximum yaw rate value which has to be chosen *a priori* and remains constant. Based on [49], the maximum rudder deflection angle is taken as 35°. The physical quantities L and ∇ should be estimated. The length on waterline of the vessel L could be approximated using extended object tracking [50]. In extended object tracking, shape parameters are incorporated into the state vector. This can also be done for the width of the vessel. However, we need the under water volume of the target. The draft of the vessel is not visible by the camera. Therefore, this will have to be approximated based on the quantities that are observable, or estimated *a priori*. For this thesis, the assumption is made that the length and volume are known.

4-5 Implementation

In the previous section, some common approaches for tracking have been discussed. As discussed in Section 4-2, the movements of both the Beagle and the target or targets are tracked. The estimated state of the Beagle can be used to determine the global states of the target. This section discusses the chosen tracking approach, both for the Beagle and the target.

4-5-1 Target Tracking

Multiple tracking methods have been discussed. In the suggested literature, very basic tracking approaches were employed. The advantage of these approaches is their low complexity. Another commonly used approach is the well known Kalman filter, of which the linear form has the lowest complexity. As the name suggests, it works with linear systems or linear approximations of systems. If a single linear approximation is not sufficient to describe the entire target motion, a local approximation can be employed. The EKF is a commonly used approach for nonlinear systems. Since it uses a Taylor series approximation as a linearization of the system around a working point, it is less computationally efficient than the linear Kalman filter. However, as discussed in Section 4-4, it can be beneficial to use an EKF over a linear Kalman filter, even if the system can be described as a linear system. The mathematical operations required to process the Kalman filter equations are very computationally cheap, compared to the image processing operations in the detection algorithm. Therefore, it is expected that the computation time increase related to the linearization is insignificant.

The other tracking approach which was discussed, is the particle filter. An advantage of this approach is that it should be able to handle highly nonlinear systems. The major drawback is the computational complexity of the approach. As discussed in Subsection 4-3-3, real-time operation could only be obtained for an image resolution of about (500x300). Based on the relatively large target range, it is expected that a higher resolution is required to obtain good performance. This would violate the real-time operability requirement.

The last tracking approach which was discussed is the IMM. The IMM framework uses multiple different filter models, and combines these into a final state estimate. The approach aims to combine the advantages of different filter models. Since the system evaluates the model performance in every time step, the advantage of a filter model performing well in a certain situation is emphasized. The combination of multiple filters, adds computational complexity. However, for the same reason as stated above, it is expected that this does not significantly increase the computation time of the system.

The particle filter struggles to process high resolution images in real-time. The distance of the potential targets requires that the resolution is increased compared to the resolution described in literature, to be able to discern the target from the background. Of course, more computing power would overcome this, but considering there are other suitable options that are likely to meet the real-time requirement, the particle filter is not considered as a suitable option for this application. One of the other design requirements is that the system should be able to work under manoeuvring of both vessels. The IMM is specifically designed to do this. Motivated by this, it is decided to use the IMM framework to track the targets. Multiple

Kalman filters are incorporated in the IMM framework. Because the (extended) Kalman filter is employed in the IMM, these filters will also be evaluated separately in the results.

To cover the manoeuvres of the target, the IMM makes use of the Constant-Velocity (CV) and Coordinated-Turn (CT) motion models, described in Equations (4-32) and (4-36) respectively. In order to compare the performance of the algorithm, also the linear Kalman filter and EKF are implemented separately. The linear Kalman filter employs the CV model defined in Equation (4-30) and the EKF employs the Coordinated-Turn, Constant-Velocity (CTCV) model defined in Equations (4-38) and (4-39). Next to the standard motion models, results are computed for an IMM utilizing the CV model and the Constant-Rudder (CR) model. The latter is the ship model defined in Equations (4-50), (4-51) and (4-52). The target state vector $\mathbf{X}_T$ is dependent on the motion model. For the IMM, the *a priori* transition probability matrix p is taken as

$$p = \begin{bmatrix} 0.96 & 0.02 & 0.02 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}, \tag{4-53}$$

which is based on the presented literature and test simulations. Slight changes in these values did not significantly change the results.

4-5-2 Beagle Tracking

To obtain the global states of the target, it is desired to know the state information of the Beagle for every time step. The prediction obtained by the Beagle tracking filter is used as input to the target tracking filter. The states of the Beagle are also required for decision-making and trajectory planning to the targets. The state vector $\mathbf{X}_B$ of the Beagle is given by

$$\mathbf{X}_B = \begin{bmatrix} x_B & y_B & \psi_B & v_B & \omega_B \end{bmatrix}, \tag{4-54}$$

where x_B and y_B represent the positions in the northern and eastern direction respectively, ψ_B the yaw or heading of the vessel, v_B the velocity and ω_B the yaw rate. As discussed previously, the Beagle is tracked in the global coordinate frame.

Using DGPS, the location of the Beagle can be accurately measured. The heading is measured by a Digital Compass (DC). Finally, the yaw rate is measured using the IMU in the vessel. This means we can update all states of the Beagle, except for the velocity of the vessel. The estimated states of the Beagle are used by the target tracking filter to obtain the global measurements. Therefore, the accuracy of the Beagle state estimation reflects back to the target position estimation. The position, heading, and velocity are used to reconstruct the target measurements, with respect to a reference point. The motion model which is used to predict the states of the Beagle is the CTCV model, described in Equations (4-38) and (4-39.

To deal with the nonlinearities in the proposed motion model, an EKF is used. A lot of state information can be measured at a high sample rate. The measurement vector for the Beagle is $z = \begin{bmatrix} x & y & \psi & \omega \end{bmatrix}$. Only the velocity is not measured. Therefore, it is expected that a single filter, using one dynamic model, will suffice to properly estimate the states.

The final product is a fully autonomous vessel, also capable of decision making, trajectory planning and trajectory control. Therefore, the Beagle also requires a control loop, responsible for the trajectory following of the vessel. Such a control loop will also need some form of state estimation. Therefore, the Situational Awareness System (SAS) can make use of the filter which is used in the control loop of the Beagle.

It could be advantageous for the SAS to incorporate the controller responsible for the Beagle movement into the estimation, since this would additionally adjust the states considering the true input, instead of only the resulting physical effects. However, the controller design is not in the scope of this thesis. Therefore, the control input will be disregarded.

4-6 Coordinate Conversion

In a synchronous sensor system, all sensors take every measurement at the same time instant. Asynchronous sensors operate independently and at different sample rates. For this application, the radar and camera measurements are obtained at highly different sample rates. The radar is sampled at 0.5Hz, while the camera samples at 15Hz. To prevent sensor information loss, we make use of asynchronous sensor updates.

The detections are provided in relative polar coordinates. However, the motion equations are described in Cartesian coordinates. Therefore, the detection has to be converted to the global coordinate frame. In the previous section, it was decided to track the targets with a Kalman filter approach. However, multiple options are still available to convert the measurements to the correct coordinate system. The mixed-coordinate EKF, and the Converted Measurement Kalman Filter (CMKF), of which the relative measurement models are presented in [51], are discussed in the following subsections.

Mixed-Coordinate EKF

The mixed-coordinate EKF incorporates the coordinate transformation into the filter. In the filter, the polar coordinates are implicitly transformed to Cartesian coordinates. The radar provides the measurements $z = \begin{bmatrix} r, & \alpha, & \dot{r} \end{bmatrix}$, where r is the range, α the bearing, and $\dot{r}$ is the radial range rate. The estimates of these predictions are obtained by the nonlinear global measurement model $H(X_T, X_B)$, which is a function of the target state X_T and the Beagle state X_B. The measurement model is defined as

$$H(X_T, X_B) = \begin{bmatrix} \hat{r} \\ \hat{\alpha} \\ \hat{\dot{r}} \end{bmatrix} = \begin{bmatrix} \sqrt{(\hat{x}_T - \hat{x}_B)^2 + (\hat{y}_T - \hat{y}_B)^2} \\ \arctan\left(\frac{\hat{x}_T - \hat{x}_B}{\hat{y}_T - \hat{y}_B}\right) - \hat{\psi}_B \\ -\hat{v}_T \cos(\hat{\alpha} + \hat{\psi}_B - \hat{\psi}_T) - \hat{v}_B \cos(\hat{\alpha}) \end{bmatrix}, \tag{4-55}$$

where the subscripts T and B represent the target and Beagle states respectively. The Beagle states are updated using another filter. Therefore, they are incorporated in the measurement model as constants. The Jacobian of the measurement model J_H is computed to determine the Kalman gain.

$$J_H(x) = \begin{bmatrix} \frac{\hat{x}_T - \hat{x}_B}{\sqrt{(\hat{x}_T - \hat{x}_B)^2 + (\hat{y}_T - \hat{y}_B)^2}} & \frac{\hat{y}_T - \hat{y}_B}{\sqrt{(\hat{x}_T - \hat{x}_B)^2 + (\hat{y}_T - \hat{y}_B)^2}} & 0 & 0 & 0 \\ \frac{\hat{y}_T - \hat{y}_B}{\hat{x}_B^2 - 2\hat{x}_B\hat{x}_T + \hat{y}_B^2 - 2\hat{y}_B\hat{y}_T + \hat{x}_T^2 + \hat{y}_T^2} & \frac{\hat{x}_B - \hat{x}_T}{\hat{x}_B^2 - 2\hat{x}_B\hat{x}_T + \hat{y}_B^2 - 2\hat{y}_B\hat{y}_T + \hat{x}_T^2 + \hat{y}_T^2} & 0 & 0 & 0 \\ \frac{\delta\alpha}{\delta x_T}(\hat{v}_T \sin(\hat{\gamma}) - \hat{v}_B \sin(\hat{\alpha})) & \frac{\delta\alpha}{\delta y_T}(\hat{v}_T \sin(\hat{\gamma}) - \hat{v}_B \sin(\hat{\alpha})) & \hat{v}_T \sin(\hat{\gamma}) & -\cos(\hat{\gamma}) & 0 \end{bmatrix}, \tag{4-56}$$

where $\hat{\gamma} = \hat{\alpha} + \hat{\psi}_B - \hat{\psi}_T$, for better readability. The implicit coordinate change is done by multiplying the resulting Kalman gain with the measurement error. The camera update is done using only the expression for α in Equation (4-55) and its corresponding Jacobian.

An EKF works well if the linearization errors are small and the system operates within a restricted range. Otherwise, errors might be introduced. As can be seen in Equation (4-55), the range rate equation is highly nonlinear. Therefore, some target tracking filters ignore the range rate measurements. In [52], the author discusses this issue, and proposes an alternative linearization. The alternative linearization is based on the idea that the expected value of the first two terms of the range rate Jacobian sum to zero, $E\left[\frac{\delta \dot{r}}{\delta x}\hat{x} \quad \frac{\delta \dot{r}}{\delta y}\hat{y}\right] = 0$. Following this approach, the complete Jacobian J_H is defined as

$$J_H(x) = \begin{bmatrix} \frac{\hat{x}_T - \hat{x}_B}{\sqrt{(\hat{x}_T - \hat{x}_B)^2 + (\hat{y}_T - \hat{y}_B)^2}} & \frac{\hat{y}_T - \hat{y}_B}{\sqrt{(\hat{x}_T - \hat{x}_B)^2 + (\hat{y}_T - \hat{y}_B)^2}} & 0 & 0 & 0 \\ \frac{\hat{y}_T - \hat{y}_B}{\hat{x}_B^2 - 2\hat{x}_B\hat{x}_T + \hat{y}_B^2 - 2\hat{y}_B\hat{y}_T + \hat{x}_T^2 + \hat{y}_T^2} & \frac{\hat{x}_B - \hat{x}_T}{\hat{x}_B^2 - 2\hat{x}_B\hat{x}_T + \hat{y}_B^2 - 2\hat{y}_B\hat{y}_T + \hat{x}_T^2 + \hat{y}_T^2} & 0 & 0 & 0 \\ 0 & 0 & \hat{v}_T\sin(\hat{\gamma}) & -\cos(\hat{\gamma}) & 0 \end{bmatrix}. \tag{4-57}$$

Converted Measurement Kalman Filter

The conversion process used in the CMKF involves transforming the raw measurement into a converted measurement and estimating the converted measurement noise covariance matrix.

Most literature describes the CMKF only for the position estimates [51][53]. Therefore, this is discussed first. As discussed in the previous subsection the radar provides the measurements $z = \begin{bmatrix} r, & \alpha, & \dot{r} \end{bmatrix}$. For the position conversion, we consider the first two observations, range and bearing. The literature described above assumes a static observer. To incorporate the moving Beagle, the relative position is translated to the global coordinate system by adding the Beagle position. The Beagle heading is used to rotate the observation to the global coordinate system, which is the navigation frame. The nonlinear radar measurement conversion from relative polar coordinates to global Cartesian coordinates is computed as

$$\begin{aligned} x_T &= r\sin(\alpha + \hat{\psi}_B) + \hat{x}_B, \\ y_T &= r\cos(\alpha + \hat{\psi}_B) + \hat{y}_B. \end{aligned} \tag{4-58}$$

The resulting bearing relative to the navigation frame will be denoted as $\alpha_N = \alpha + \hat{\psi}_B$. The Beagle measurement update is done in a separate filter, which is done before the update of the tracks. In the Beagle filter, the measurement noise is already suppressed. Therefore, there is no need to consider this noise again, so the Beagle estimates are considered as constants. The position measurement noise covariance matrix of the radar measurements is

$$R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\alpha^2 \end{bmatrix}, \tag{4-59}$$

where σ_r and σ_α denote the standard deviations for the range and bearing respectively. However, the measurements used to update the states are in Cartesian coordinates. Therefore, the noise covariance matrix should also be converted to Cartesian coordinates. The conversion of the position measurements are given in Equation (4-58). The Jacobian J_T of these equations is used as the transformation matrix

$$J_T = \begin{bmatrix} \frac{\delta x_T}{\delta r} & \frac{\delta x_T}{\delta \alpha_N} \\ \frac{\delta y_T}{\delta r} & \frac{\delta y_T}{\delta \alpha_N} \end{bmatrix} = \begin{bmatrix} \sin(\alpha_N) & r\cos(\alpha_N) \\ \cos(\alpha_N) & -r\sin(\alpha_N) \end{bmatrix}. \tag{4-60}$$

This matrix is used to convert the measurement noise covariance matrix of Equation (4-59) to the measurement noise covariance matrix in Cartesian coordinates [54].

$$R_{xy} = J_T R J_T^T \tag{4-61}$$

$$= \begin{bmatrix} \sin^2(\alpha_N)\sigma_r^2 + r^2\cos^2(\alpha_N)\sigma_\alpha^2 & \sin(\alpha_N)\cos(\alpha_N)(\sigma_r^2 - r^2\sigma_\alpha^2) \\ \sin(\alpha_N)\cos(\alpha_N)(\sigma_r^2 - r^2\sigma_\alpha^2) & \cos^2(\alpha_N)\sigma_r^2 + r^2\sin^2(\alpha_N)\sigma_\alpha^2 \end{bmatrix}. \tag{4-62}$$

This matrix is computed in every time step. In [51], the authors present a potential issue in this approach. The expected value of the conversion has a bias in the mean of the converted measurement. This bias is found by taking the expectation of the converted range and bearing measurements. Assuming the range measurement noise w_r, and bearing measurement noise w_α, are uncorrelated, zero mean and Gaussian with standard deviations of σ_r and σ_α respectively, the expectation is

$$E\begin{bmatrix} (r + w_r)\cos(\alpha_N + w_\alpha) \\ (r + w_r)\sin(\alpha_N + w_\alpha) \end{bmatrix} = e^{-\sigma_\alpha^2/2}\begin{bmatrix} r\cos(\alpha_N) \\ r\sin(\alpha_N) \end{bmatrix}. \tag{4-63}$$

As can be seen, there is a bias along the true bearing to the target. This bias can be eliminated, by introducing a multiplicative compensation $e^{\sigma_\alpha^2/2}$. The unbiased converted measurements x and y are defined as

$$\begin{bmatrix} x_T \\ y_T \end{bmatrix} = e^{\sigma_\alpha^2/2}\begin{bmatrix} r\cos(\alpha_N) \\ r\sin(\alpha_N) \end{bmatrix} + \begin{bmatrix} \hat{x}_B \\ \hat{y}_B \end{bmatrix}. \tag{4-64}$$

The measurement covariance computation requires the true range and bearing. Therefore, it cannot be computed in practice. Multiple variations have been presented to deal with this problem [55]. One of them is to compute the measurement covariance matrix using the measurement values, in the same manner as Equation (4-61), leading to the measurement noise covariance components

$$\begin{aligned} R_{11} &= \frac{1}{2}(r^2 + \sigma_r^2)\left[1 + \cos(2\sigma_\alpha)e^{-2\sigma_\alpha^2}\right] + (e^{2\sigma_\alpha^2} - 2)r^2\cos^2(\alpha_N), \\ R_{22} &= \frac{1}{2}(r^2 + \sigma_r^2)\left[1 - \cos(2\sigma_\alpha)e^{-2\sigma_\alpha^2}\right] + (e^{2\sigma_\alpha^2} - 2)r^2\sin^2(\alpha_N), \\ R_{12} &= \frac{1}{2}(r^2 + \sigma_r^2)\sin(2\sigma_\alpha)e^{-2\sigma_\alpha^2} + (e^{2\sigma_\alpha^2} - 2)r^2\cos(\alpha_N)\sin(\alpha_N). \end{aligned} \tag{4-65}$$

In [53], the author presents a comparison between the Mixed-Coordinate EKF and the CMKF. It is shown that both the debiased and conventional conversion approaches outperform the mixed-coordinate EKF for a bearing standard deviation $\sigma_\alpha = 1.5°$. Furthermore, it was stated that the CMKF also outperformed the MCEKF for sensors with $\sigma_\alpha = 0.5°$.

The alternative linearization discussed in the previous subsection, can also be used for the CMKF approach. The authors of [52] convert the position estimates using the CMKF approach, while the range rate is used by the filter directly. In [56], the same is done more recently. This leads to the following measurement Jacobian:

$$J_H(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & v_T \sin(\gamma) & -\cos(\gamma) & 0 \end{bmatrix}. \tag{4-66}$$

It is possible to convert the range rate to Cartesian coordinates outside the filter, like position measurements of the CMKF. However, in [55], it is shown that this adds extra complexity, while no performance improvement was found for a small bearing standard deviation $\sigma_\alpha = 1°$.

Motivated by this, the updates of the filter are done using the CMKF for the position, and the alternative linearization for the range rate update. The position measurement conversion used the conventional method, with unbiasing step.

Range Variance of Camera Detection

The camera detection algorithm only provides the target bearing. However, to transform the polar coordinates to Cartesian coordinates, we require the bearing and the range. The range update frequency is significantly lower than the bearing update frequency. To overcome this, the prediction of the range is used to obtain the transformation of the camera bearing to Cartesian coordinates.

Only the radar measurement updates the (converted) range. Therefore, this predicted range becomes more uncertain for increasing time without radar measurement. Essentially, this means that the standard deviation of the range error might increase with every time step in which no radar detections are obtained. This is depicted in Figure 4-12. It can be seen that in some occasions, the range error significantly increases between radar detections. To incorporate this growing uncertainty in the filter, we propose to introduce a time dependency in the range variance, which is also shown in Figure 4-12.

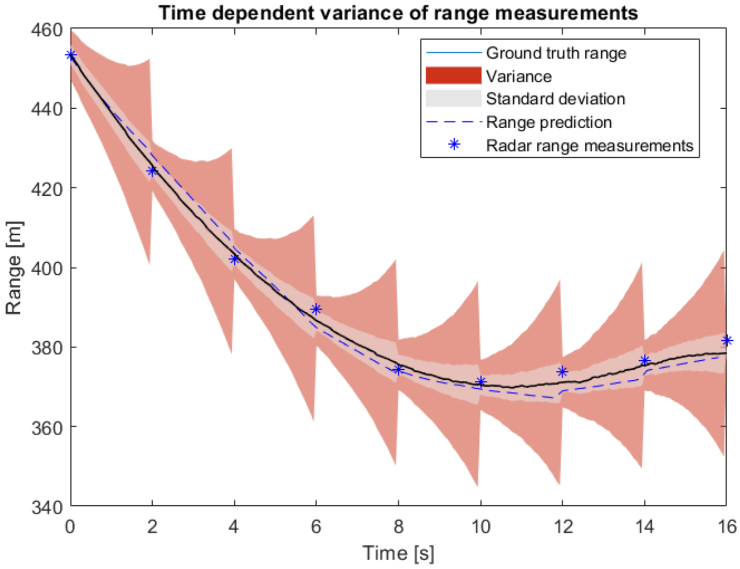


Figure 4-12: Increasing variance on range between radar updates

Manoeuvring of the target can influence the range error. To cover possible manoeuvres of the target, and thus range errors, the range variance σ_k^2 is taken dependent on time exponentially

$$\sigma_k^2 = \sigma_r^2 b^l, \tag{4-67}$$

where l denotes the amount of time steps without radar detection, b is the growth factor, and σ_r^2 is the range variance of the radar. The proposed time dependency in the range variance was evaluated for different growth factors. The result is shown in Figure 4-13. The position RMSE decreases for five of the scenarios. For both Scenario 2 data sets, a minimum is found for a growth factor $b = 1.1$. For the other three data sets which obtained an improvement, the RMSE decreases to a certain value for increasing growth factor, until it remains constant. This can be explained by looking at the influence of the range variance on the Kalman gain. If the range variance is large, this results in a low Kalman gain for the coordinate relevant for the target direction. If the target is located north of the ego vessel, i.e. they have the same x-coordinate, a range error is mainly influencing the y-coordinate. For a high range variance, the Kalman gain influencing the y-coordinate will be small. Increasing the range variance further, will just slightly decrease the already small Kalman gain for this coordinate. Therefore, the performance becomes constant at some point.

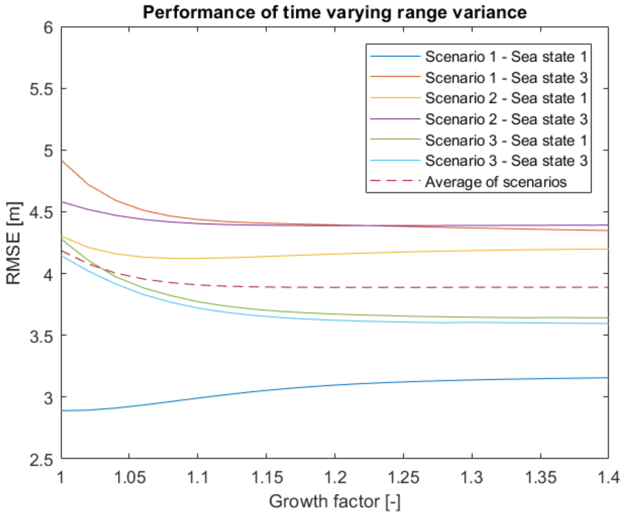


Figure 4-13: Position RMSE result of incorporating time dependent range variance

The data set "Scenario 1 - Sea state 1" did not improve for the time dependency of the range variance. The range error mainly increases when the heading is frequently estimated incorrectly. A heading error causes a range error over time, since the camera does not provide range measurements. When the heading is estimated correctly by the algorithm, the range error is also likely to remain relatively small. The increasing range error was considered as increasing measurement variance. Therefore, increasing the variance, while the range error does not significantly increase with time, has as a result that the filter performs less optimal. If we consider the average RMSE of all data sets, we observe an improvement of 7% for a growth factor $b = 1.1$, compared to the performance without time dependency. Simply increasing the range value of the camera without the time dependency also showed improvement. However, this was less than with time dependency.

4-7 Results

In this section, the results of the evaluation of the complete algorithm are discussed, with a focus on the tracking algorithm. The algorithm has been evaluated for different test cases, which is discussed in Subsection 4-7-1. After this, some flaws in the simulation software are discussed in Subsection 4-7-2. Finally, the algorithm is evaluated on multiple aspects in Subsection 4-7-3. In Chapter 3, the camera algorithm is evaluated according to performance indicators. The tracking algorithms are not evaluated separately, as this is done extensively in literature. The performance of the complete algorithm is quantified in terms of the Root Mean Square Error (RMSE), which is computed as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{z}_i - z_i)^2}, \tag{4-68}$$

where n is the total number of time steps. This metric and computation time per frame are used to reflect on the design requirements.

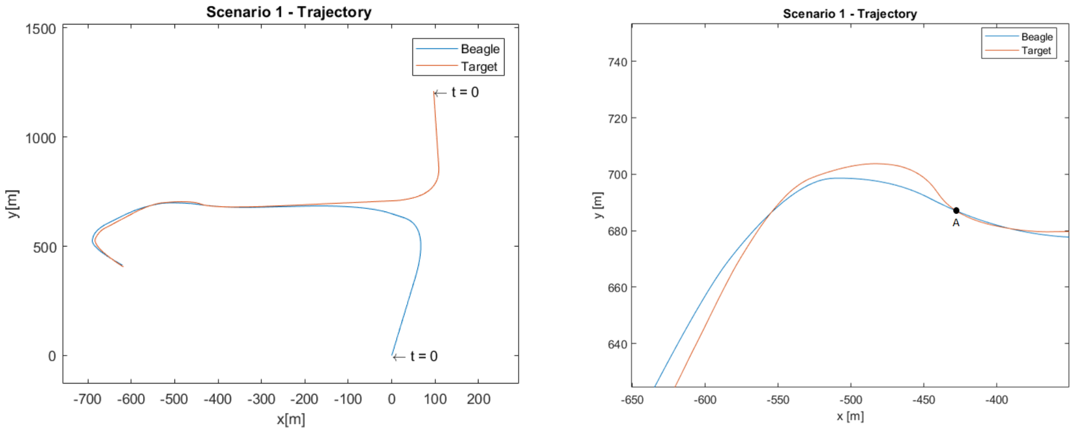
4-7-1 Test Cases

The algorithm is tested on three different scenarios, each under two different weather conditions. Thus, we have 6 data sets, with a total of 13400 frames. These data sets consider one target. Two of the scenarios are also simulated considering two targets. All three scenarios are pursuits, where the Beagle is aiming to intercept the target vessel. However, the initialization and evasive actions of the target are different.

The velocity is based on the Piracy & Armed Robbery Report 2019, which is updated almost daily by the ICC [57]. Multiple attack reports state that piracy vessels approached merchant ships at a velocity of around 23 knots, which corresponds to a velocity of 12 m/s. To simulate this while taking into account some safety margin, the vessels in the simulation are moving at a velocity of around 14 m/s. Since the trajectory planning and control algorithms are not designed yet, the trajectories of the vessels are predefined in the simulation software. The actions of the Beagle are defined such that the target remains in the camera Field Of View (FOV) for the vast majority of the time. The specifications of the scenarios is discussed below.

Scenario 1 - The first scenario simulates a piracy attack, where the hostile vessel is moving straight towards the merchant vessel. The Beagle is released, and moves towards the target. Initially, the target is located around 1200 meters from the Beagle, moving straight towards the Beagle. This decreases to 150 meters towards the end of the scenario. At some point in time, the target initiates an evasive manoeuvre to the right. The Beagle responds by following this action. Figure 4-18a depicts the trajectories of the Beagle and the target vessel. The target vessel initiates another evasive manoeuvre at point A. This manoeuvre can be seen in Figure 4-18b. After the first evasive manoeuvre, the Beagle is following at about 150 meters behind the target vessel.

To test the ability of the algorithm to track multiple targets at the same time, another target was added to scenario 1. This can be observed in Figure 4-15. The track of the first target remains the same. The track of the second target is defined such that the target is in the camera FOV most of the time, but occasionally moves out of the image frame. The velocity of target 2 is also around 14 m/s.



(a) Full trajectories

(b) Evasive manoeuvre at point A

Figure 4-14: Trajectories of Beagle and target for scenario 1

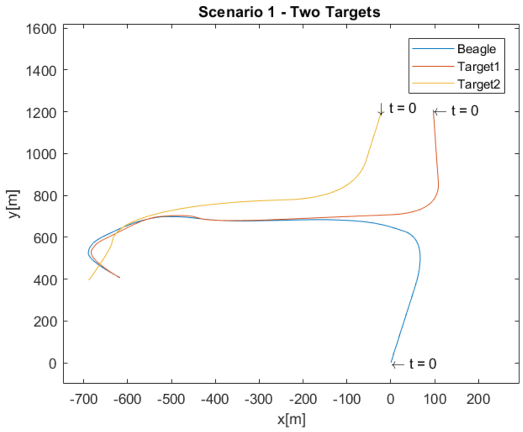
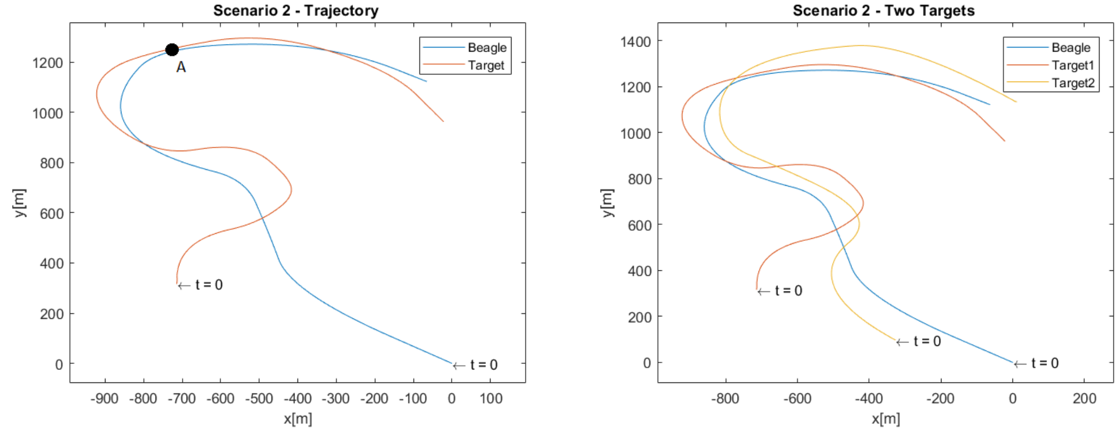


Figure 4-15: Scenario 1 with two targets

Scenario 2 - The second scenario is initialized with the target moving nearly perpendicular to the Beagle heading. This means that the initial Doppler measurement of the radar, which provides the range rate, is nearly zero. This makes it difficult to estimate the velocity at the beginning of the scenario. The trajectory can be seen in Figure 4-16a. The target vessel follows a zig-zag trajectory, crossing the Beagle trajectory multiple times. The Beagle starts at a distance of 800 meters from the target, which decreases to about 250 meters during the zig-zag up to point A in the image. After the vessel move passed point A, the range decreases to about 150 meters until the vessels reach the end of the trajectory. The scenario is also simulated with two vessels, which is depicted in Figure 4-16b. Again, the first target track remains the same. The two targets visually cross each other multiple times.



(a) Trajectories of Beagle and target for scenario 2

(b) Scenario 2 with two targets

Figure 4-16: Trajectories of Beagle and target for scenario 1

Scenario 3 - The third and last scenario starts with the Beagle already positioned right behind the target. The range is between 100 and 150 meters for the entire scenario. As can be seen in Figure 4-17, the target vessel makes some strong evasive manoeuvres in both directions. The close range and strong manoeuvres are a challenge for both the camera detection algorithm and the data association algorithm. If the camera detection algorithm only classifies part of the target correctly, the bearing error might increase, due to the short range. As the target takes up a significant part of the frame, the corresponding bearing span is significant. Furthermore, the strong manoeuvres are a challenge. If the prediction of the tracking algorithm is slightly off, the high yaw rate causes the obtained camera detection to fall outside the gating region.

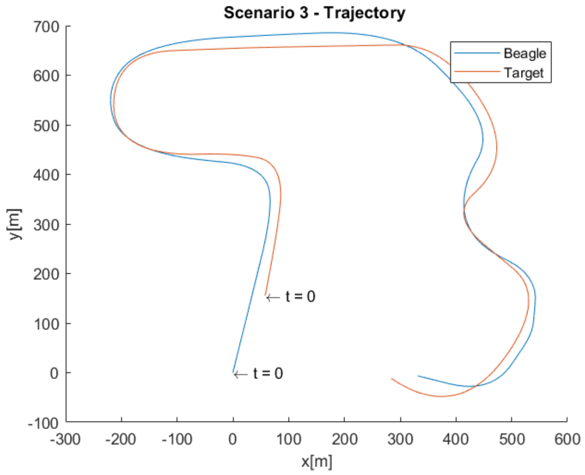


Figure 4-17: Trajectories of Beagle and target for scenario 3

4-7-2 Simulation Limits

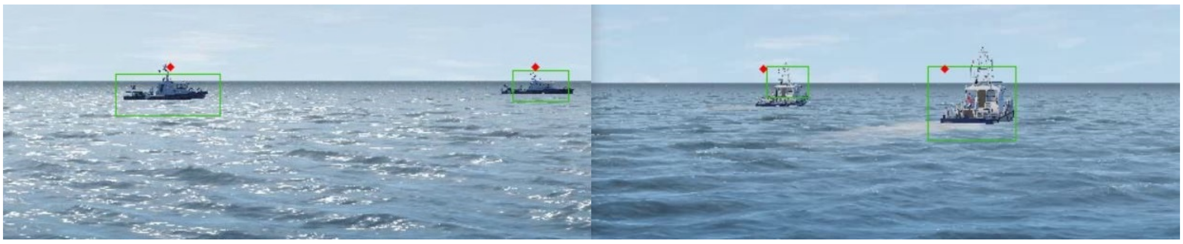
As discussed in Section 2-4, the sensor data is acquired using the Nautis software. The version which is used is a beta version of the software. Therefore, not all capabilities of the software are fully operational. For instance, the dynamic models of the vessels are very basic, resulting in acceptable, but slightly imprecise ship movements. Furthermore, some other properties of the software create inaccuracies in the resulting data. These flaws in the simulation software are discussed in this subsection.

Data export protocol - The tracks presented in the previous subsection are predefined in the simulation software. When the simulation is started, the vessels start following their track. It is desired to obtain sensor data from all vessels present in the simulation. However, the data export protocol of the software allows the sensor data export of only one vessel for each session. Therefore, a scenario has to be simulated once for each vessel. Multiple sessions have to be recorded and synchronized to obtain the complete data set. Due to the random sea motion, these simulations are never exactly the same. Furthermore, the initiation of the simulation requires a sudden computational load, which causes small delays. Therefore, it is difficult to determine the exact start of the simulation in the data.

Visual feed synchronization - Another issue is that it is not possible to export the visual feed of the simulation directly in the software. To obtain the visual feed, it has to be recorded externally, for instance using screen capture software. Because of this alternative exporting method, the visual feed and the sensor data are not synchronized. Because of the computationally intensive Nautis software, the simulation PC frequently suffers from delays in the visual feed. If the visual feed would be exported by the software, these delays can be compensated using simulation time stamps. The screen capture software simply records the screen at a certain frame rate, without time stamping the images.

These two difficulties make the synchronization inaccurate and time consuming. It is attempted to synchronize the data at the start and end of the scenario, which is slightly inaccurate due to the initialization characteristics of the software. The synchronized sensor data sets are interpolated to obtain the same amount of data points as the images in the visual feed data. Still, the limitations in the simulation cause temporal and spatial mismatches between the sensor data and the visual feed. This can be seen in Figure 4-18. The red diamond represents the expected location of the target in the image based on the sensor data. The DGPS transponder is located on the communications mast on top of the ship. In Figure 4-18 (a), we see that the predicted location on the image matches the actual location of the communications mast of the vessel in the image. This instant is at the beginning of the scenario. Figure 4-18 (b) is depicting the same scenario at half of the total time. There exists a mismatch between the actual location of the mast and the location predicted by the sensors. This offset is due to the inaccuracy in the synchronization of the sensor data sets and the visual feed data. This offset does not steadily grow over time, but oscillates around zero.

Of course, nearly every sensor system suffers from delays, which are often unpredictable. However, in case of one target, there are three data streams which should be synchronized: the visual feed, the Beagle sensor data and the target sensor data. When the sensor data of the target and the Beagle are not exactly synchronized, the radar data generated from



(a) Match between sensor data and visual feed **(b)** Offset between sensor data and visual feed

Figure 4-18: Visual feed with detected bounding box (green), and sensor data position (red diamond)

these two data sets becomes biased. A temporal shift in the two data sets, causes the relative position of the vessels to be different. The spatial error which is incorporated in the radar detection is very difficult to correct when the exact delay is not known. In a real life system, this issue does not occur, because the radar detection is based on the actual reflections of its signals. Therefore, the detection only suffers from general measurement noise, and possibly time delay.

4-7-3 Evaluation

The algorithm has been tested on the data sets presented in subsection 4-7-1. The noise covariance matrices of the filters have been determined based on a different data set. The evaluation considers the camera detection parameters, the algorithm design and the proposed modifications.

Image Resolution

In Chapter 3, the performance differences between image resolutions and detection thresholds were presented. We observed that for low sea states, the small image resolution of $(800x225)$ performed better than the larger image resolutions. A question that remained after this chapter was how to decide on the precision-recall trade-off. Should we aim for high recall, or high precision? This can be answered by investigating the performance of the tracker, evaluated over the three different test scenarios, considering a span of thresholds.

As we have seen in Subsection 3-7-3, a high threshold generally results in high precision, but low recall. A lower threshold generally results in high recall, but with a lower precision. However, this evaluation did not take the data association and tracking into account. It evaluates the performance of just the detection algorithm. One of the advantages of the camera is the higher update frequency. To use this advantage, it makes sense to aim for a high recall. However, when the detection precision becomes too low, the tracking algorithm will start to suffer from more false detections. The performance of different image resolutions and detection thresholds can be observed in Figure 4-19.

The figure shows the average performance on the different scenarios, with the maxima and minima as error bars. As can be seen, the image resolution does not significantly influence the performance of the tracker. This is different from what we observed in Section 3-7. Using

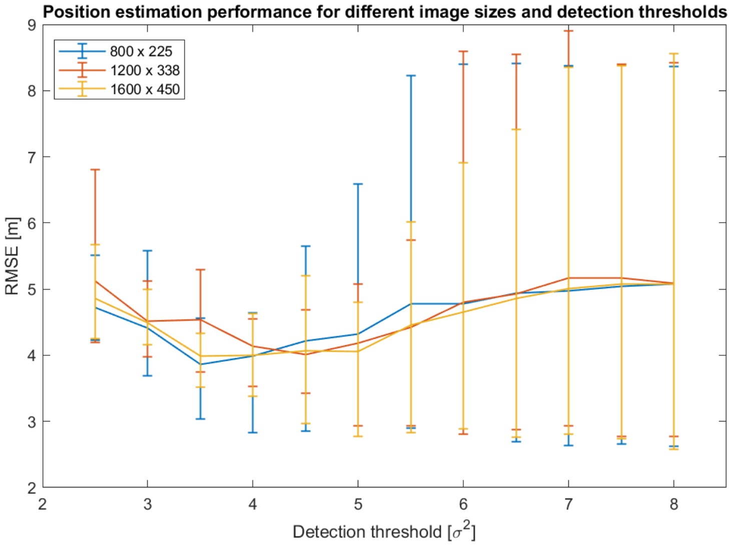


Figure 4-19: RMSE of the position estimate

a larger image resolution, the camera detection range increases in a difficult environment, e.g. high sea state. Therefore, the camera detection algorithm obtained a higher precision for a high recall value. In the tracking algorithm, this difference is less visible due to the addition of the radar measurements. The radar measurements allow that the target is tracked despite the absence of camera detections. In the detection results, we observed a strong increase in the computation time for the larger image resolutions. Therefore, the image resolution ($800x225$) is used to meet the real-time requirements of the system. It can be seen that an optimum exists. For the image resolution ($800x225$), the best performance is obtained for a threshold of 3.5 times the standard deviation. Using this threshold, a recall of about 0.8 was obtained.

If the threshold is decreased, the precision of the detection algorithm decreases. On the other hand, the recall increases. This can be observed in Figure 4-19 for thresholds below 3.5 times the standard deviation. The RMSE value increases due to false detections. If we increase the detection threshold, we also observe an increasing average error. This is due to the decreasing recall. However, for one of the scenarios, the position RMSE becomes lower for an increasing threshold. This is scenario 3, in which the Beagle is in close proximity of the target. For lower thresholds, the algorithm tends to generate a bounding box which is slightly oversized. Because of the close proximity, small bearing estimation errors occur. The bounding box is larger when the target is close. If the bounding box becomes slightly too wide on one side of the target, the middle of the bounding box shifts, and thus also the detected bearing. With a higher threshold, the algorithm suffers less from this effect, which results in a lower error in close range.

It is possible to implement different detection thresholds for different scenarios, which can be chosen when the Beagle is released. However, for the remainder of this thesis, only one value is used for simplicity. The threshold which is used is 3.5 times the standard deviation, with an image resolution of ($800x225$).

Tracking Filter

The tracking filters make use of motion models to predict the states.n In Section 4-4, multiple dynamic models have been discussed. On the models having linear state update equations, a linear Kalman filter can be applied. Others models have nonlinear state update equations, which can be handled by an EKF. The IMM framework can apply both filter methods, and the corresponding dynamic models. The performance of different filter models is compared. Furthermore, the choice was made to fuse radar with visible-light camera. Therefore, the performance difference between radar, and the radar-camera fusion is presented. The complete results table is added in Appendix C.

Position - The position RMSE for the filter methods is shown in Figure 4-20. The filter approaches are structured from left to right with increasing complexity. The Constant Velocity CV approach uses the assumption of a non-manoeuvring target and is employed by a linear Kalman filter. The Coordinated Turn - Constant Velocity CTCV approach includes the yaw rate in the system as a state. Since this makes the state transition equations nonlinear, an EKF is used to employ this model. The third filtering approach is the interacting multiple model, which makes use of the constant velocity model and coordinated turn model (CTCV). Note that this is a combination of a constant velocity model, and a coordinated turn model, instead of the incorporation into one model. The final approach is again the IMM, but making use of the proposed ship model. The results are discussed separately for the position, heading and velocity estimation.

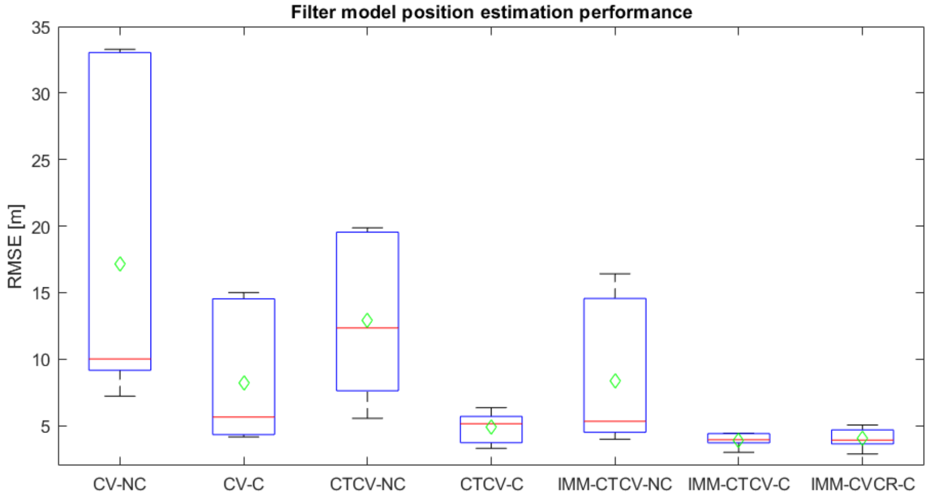


Figure 4-20: Position RMSE for different filter and motion model approaches, with the green diamond representing the average RMSE. From left to right the filter models: CV - without and with camera resp. (KF), CTCV - without and with camera resp. (EKF), IMM - CT and CV - without and with camera resp., IMM - CV and Constant Rudder (Ship Model)

It can be seen that the "Constant-Velocity model - no camera" performs worst in terms of position RMSE. Adding the camera detections improves the average RMSE significantly. When the EKF only utilizes the radar measurements, some improvement is observed in the average RMSE compared to the linear Kalman filter without camera. However, significant performance improvement is observed when the camera detections are added, as can be seen in the RMSE for CTCV-C (EKF with camera). The CTCV-C obtained the best position RMSE on the Scenario 3 data sets. Still, the most robust performance was obtained by the IMM-CTCV, which obtained the best performance in terms of average, minimum, maximum and median values.

It must be noted that this simulation data can be considered as experimental data. Therefore, the differences which are very small cannot be considered to be conclusive. However, since the data set is made specifically focused on this application, it does show the robustness of the system, for this application. The camera fusion filter models clearly perform better than the radar only filter models. However, this relatively large advantage is mainly obtained by a better performance of the fusion algorithms on scenario 2. On the other two scenarios, the position estimation performance was still better with camera, but the difference was less significant. Scenario 2 is initialized quite poorly, since the range rate of the vessel is nearly zero. Therefore, it is difficult to estimate the target velocity from initialization. Due to the high update rate of the camera, this inaccurate initial estimate is quickly overcome. Of course, it is possible to improve the initial estimate, such that the radar-based approach performs better. However, it shows the flexibility of the algorithm in an unexpected situation.

If we observe the IMM employing the ship model, it can be seen that it performs very similar to the IMM using the standard motion models in terms of RMSE average, median, minimum and maximum. Based on this experimental data, we cannot conclude on which motion model performs better for the position estimation.

Heading - Besides the position, the algorithm also estimates the heading, velocity and yaw rate. The heading estimation performance in terms of RMSE can be seen in Figure 4-21. The filter approaches are structured in the same order as in Figure 4-20. Interestingly, the EKF has obtained the worst performance of all the filters on some of the data sets. For the CTCV-NC, this is due to the low update frequency of the radar, while five states have to be estimated. A significant error in the estimation of the yaw rate, results in a heading error over time. This is only corrected in the next measurement update. For the CTCV-C, the temporal and spatial mismatch between the data sets cause oscillations in the yaw rate estimation, which result in errors in the heading estimation. The IMM shows a significant improvement compared to the single EKF approaches.

It can be observed that the IMM-CVCR-C (ship model) obtained a performance improvement compared to the IMM-CTCV-C (standard motion models). The average heading RMSE is 12% lower using the ship model. On the first four data sets, the ship model significantly improved the performance. However, on the last two data sets (scenario 3), the standard motion models showed better performance, although the differences are relatively small. In scenario 3, the target remains relatively close to the Beagle, which is located right behind the target. Therefore, the recall of the camera algorithm is high, so the states are updated frequently. Because of this high update rate, the influence of the motion model decreases, which explains the small differences between the two motion models.

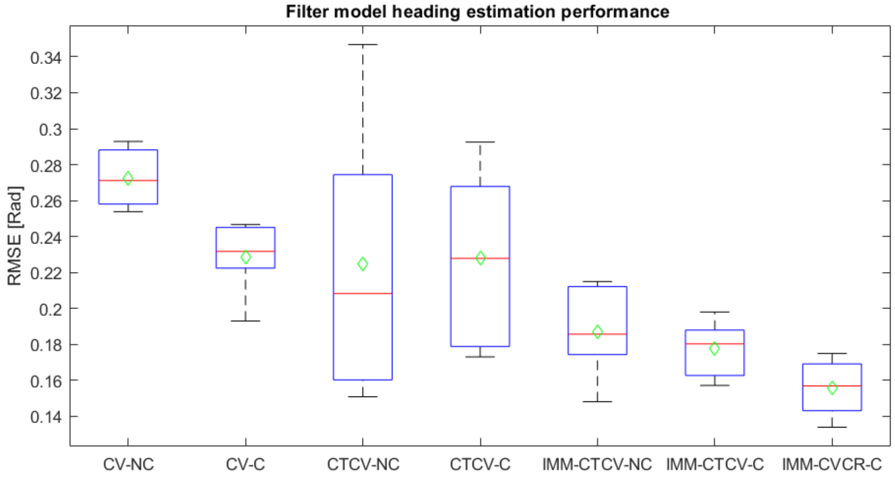


Figure 4-21: Heading RMSE for different filter approaches, with the green diamond representing the average RMSE. From left to right the filter models: CV - without and with camera resp. (KF), CTCV - without and with camera resp. (EKF), IMM - CT and CV - without and with camera resp., IMM - CV and Constant Rudder (Ship Model)

Velocity - The velocity estimation performance is depicted in Figure 4-22. On four of the data sets, the IMM-CTCV-NC, which uses only radar measurements, outperforms the IMM-CTCV-C. The two data sets for which it performs worse are for scenario 2, which is due to the poor initialization discussed previously.

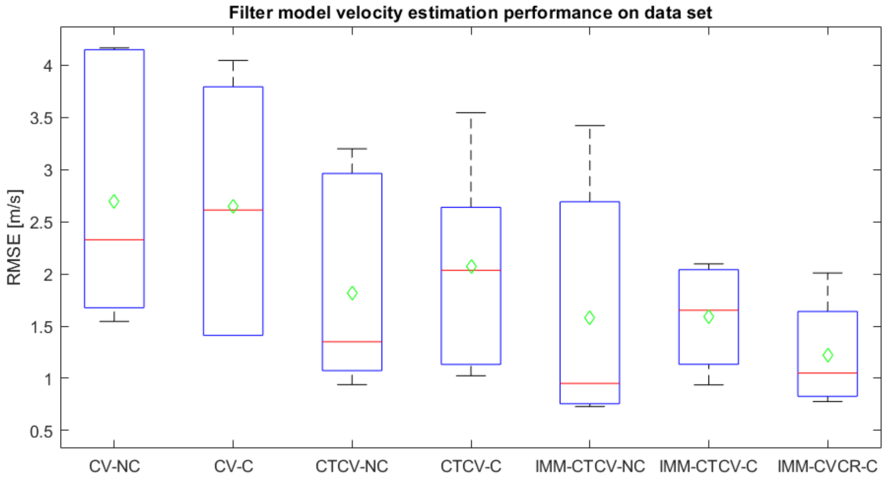


Figure 4-22: Velocity RMSE for different filter approaches,with the green diamond representing the average RMSE. From left to right the filter models: CV - without and with camera resp. (KF), CTCV - without and with camera resp. (EKF), IMM - CT and CV - without and with camera resp., IMM - CV and Constant Rudder (Ship Model)

The camera does not provide range measurements. When the target vessel is moving perpendicular to the Beagle, the target could also be moving towards or away from us. Using bearing only measurements, this cannot be observed. However, this is only possible if there exists an error in the velocity estimation, since the target would have to travel an either longer or shorter path. This is shown in Figure 4-23. For the same bearing α, the target can be located at multiple ranges, and thus also the heading can have multiple values. Because of this, heading errors can partly cause velocity errors and vice versa. This effect is more significant for the filter approaches using the camera, since the bearing is updated more frequently. If we compare the IMM-CTCV-C (standard motion model) and IMM-CVCR-C (ship model), the ship model obtained an average RMSE which was 0.35 m/s lower. This improvement was mainly on scenario 1, on which the improvement on the heading was also most significant.

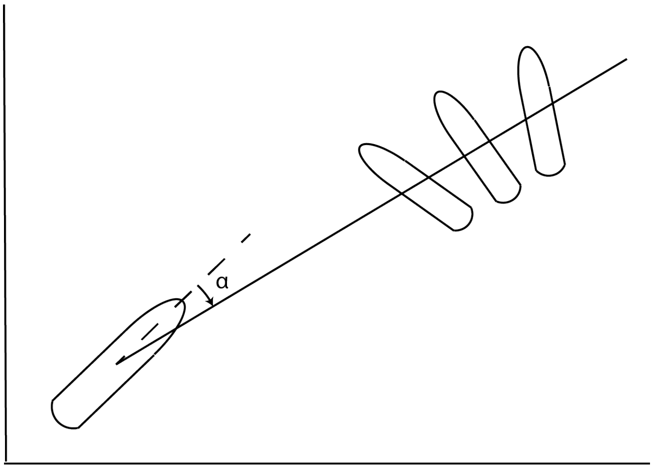


Figure 4-23: Heading errors cannot always be observed using only the bearing

Yaw rate - Direct state estimation of the yaw rate is only done using the CTCV model, employed by the EKF. The CV approach assumes the yaw rate is zero and the IMM-CTCV combines the fixed values for the yaw rate of the different motion models into one state estimation using the model probabilities. The yaw rate tends to fluctuate a lot due to the manoeuvring of the target. However, it takes some time to converge to the true values. Figure 4-24 shows the yaw rate estimation of the CTCV-C model for scenario 3 - sea state 1, which makes use of camera and radar information. It can be seen that the estimated value oscillates around the true value when it is stationary for some time. This happens mainly because of the spatial offset between the camera and the radar, which is discussed in Subsection 4-7-2. The offset causes the position to be corrected significantly in the update step of the filter. Therefore, also the yaw rate is updated, causing the oscillations. Due to the heavy oscillations and the slow convergence, the differences with the ground truth can become relatively large. Therefore, the RMSE does not show much about the performance, since a small time delay in the filter estimation can cause a large relative error. For this reason, we will not consider it. In terms of the yaw rate RMSE, the best performing filter model would be the constant velocity filter, which assumes the yaw rate to be zero. What we can conclude, is that the yaw rate estimation requires improvement. However, elimination of the spatial and temporal offset is expected to make a significant improvement. Because of the offset, the yaw rate update by the Kalman filter is significant. Elimination of this offset will decrease difference between the radar and camera bearing, and thus also the prediction error. Therefore, the yaw rate will oscillate less.

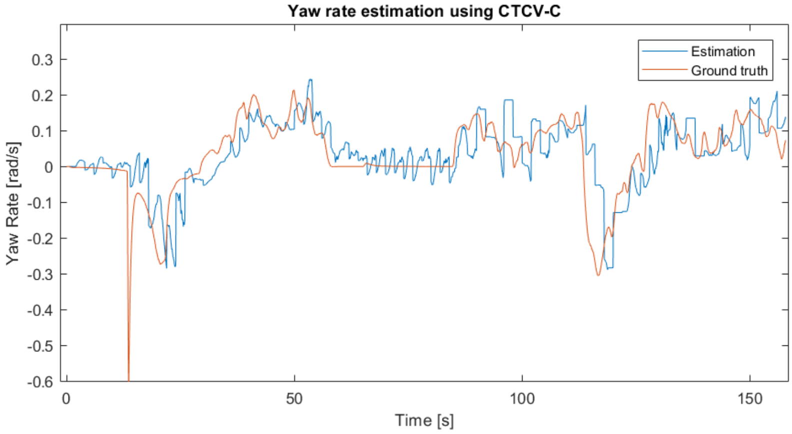


Figure 4-24: Yaw rate estimation using the CTCV-C model

Missing Radar Detections

In literature, it was stated that in higher sea states, small vessels might be hard to detect because of waves blocking the radar signal [11]. Also, the material of the skiff hull is a highly non-ideal radar reflector [58]. Radar is the main far-field obstacle detection option at the moment. However, normal navigation radar has difficulty in reliably detecting small open boats without radar reflectors. The detection can be rapidly fluctuating and the target might be hidden behind waves up to 50% of the time. This results in the target often being classified as clutter. Therefore, a design requirement was set: *The system should be robust to fluctuating target appearance due to waves, i.e. it should be able to maintain tracks even if the target does not appear in the sensor data for a period of time.* According to a certain ratio, called the drop-out rate, the radar detections have been randomly omitted. The simulation is repeated multiple times for different radar drop-out rates. For simplicity, the drop-out is considered as a random variable with a Bernoulli distribution. The simulation was repeated 10 times for each rate value. Figure 4-25 shows the position RMSE for different drop-out rates. As expected, the RMSE increases for increasing radar drop-out rate. However, the algorithm is able to maintain the target track, although the performance decreases significantly.

Multiple Targets

The algorithm is able to track multiple targets at the same time. The data association algorithm decides which detection is linked to which track. After this, the tracks are updated separately. The track updates are computed in parallel. Therefore, the extra track does not significantly increase the computation time. Figure 4-26 shows the two scenarios which involve two targets. The rectangles in the upper images are enlarged in the lower images. It can be seen that the predicted track remains close to the ground truth.

Point A, in the lower right figure, shows the drawback of the bearing only measurement of the camera. The heading of the target is difficult to determine based on the camera bearing. For the same detection bearing, the target could be located at multiple ranges. Therefore, an error in the heading estimation could occur, especially when the target vessel moves perpendicular to the Beagle. This causes an error in the velocity estimation.

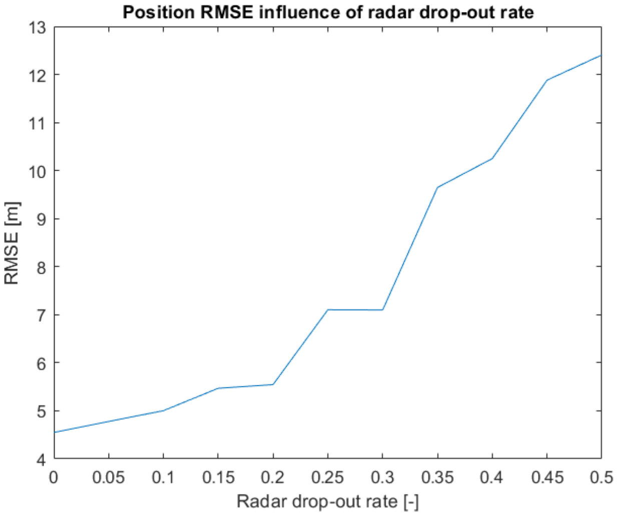


Figure 4-25: Average algorithm performance for certain radar drop-out rate

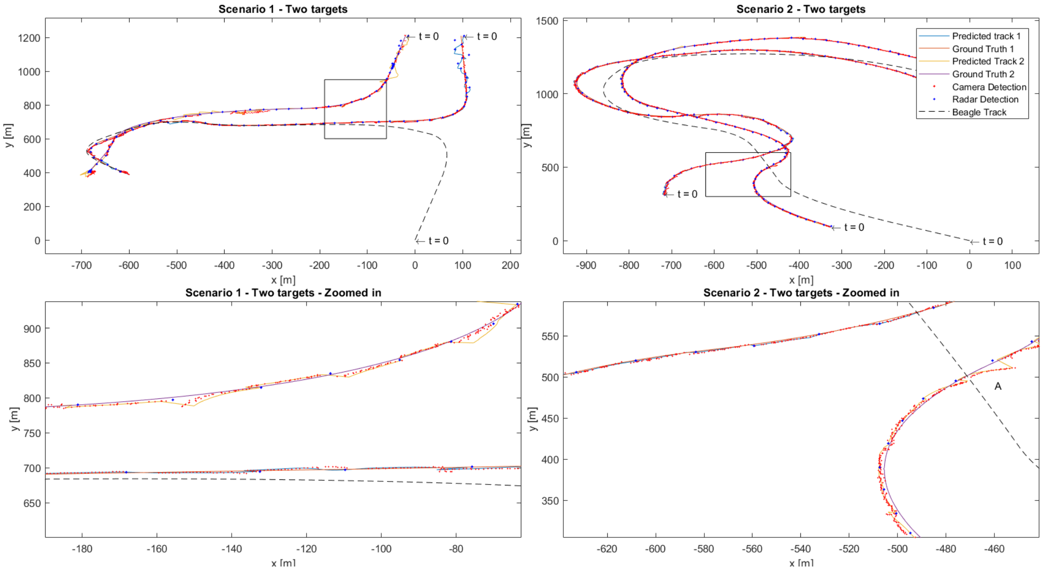


Figure 4-26: Multiple target tracking, the zoomed-in figures correspond to the black rectangles in the upper images.

The RMSE values for both scenarios are shown in Table 4-1. The performance is similar to the performance for single targets. This was expected, since the main difference is the crossing of the targets. If desired, more than two targets could also be tracked.

Table 4-1: RMSE while tracking multiple targets, track 1 and 2 respectively

Scenario\RMSE	Position [m]	Heading [rad]	Velocity [m/s]
Scenario 1	3.96, 5.88	0.21, 0.23	1.09, 1.97
Scenario 2	4.49, 2.36	0.14, 0.12	2.14, 1.25

Computation time

The design requirements were proposed in section 2-1. One of these requirements was that the system should be real-time operable. In the Chapter 3, it was shown that the chosen saliency-based detection approach surpassed real-time operability for larger image resolutions, measured on the described test hardware. As discussed in Subsection 4-7-3, image resolution ($800x225$) is used to meet the real-time operability design requirement. This creates a significant computational advantage over larger image resolutions, as we observed in Subsection 3-7-2. The complete algorithm should also be real-time operable. The update frequency in the tests was 15 Hz. The computation time per time step was measured to compare the different filter models presented previously, which is shown in Table 4-2. In view of the complete algorithm computation time, the computation time of the linear and extended Kalman filter is insignificant, since the computation time added by the filters is less than 1% of the total. Even the significance of the IMM is small, with a contribution under 5% of the total computation time per time step.

Table 4-2: Computation time per frame for different filter approaches

Filter	Computation time per frame [s]
Linear Kalman filter	6.51e-6
Extended Kalman filter	7.66e-4
Interacting multiple model (3x EKF)	3.85e-3
Complete algorithm	8.40e-2

The overall computation time per sample step, using the IMM, was 0.084 seconds, averaged over 12000 sample steps. This is slightly slower than real-time processing, which would be a computation time of 0.067 seconds per sample step. The obtained computation time corresponds to a frame rate of about 12 Hz. Intuitively, it is not expected that decreasing the frame rate to 12Hz would influence the performance of the algorithm significantly. However, to meet the current real-time requirement, one option is to simply increase the computation power. This will improve the processing speed to meet the desired frame rate. Decreasing the FOV of the camera is not expected to have an effect. As we observed in Subsection 3-7-2, the computation time of the algorithm did not decrease further for a lower image resolution. Something that could be done is the implementation of parallel processing of the camera detection algorithm and the rest of the algorithm.

Coordinate Conversion

In section 4-6, it was discussed what approach should be used to convert the measurements from polar coordinates to Cartesian coordinates. It was decided to use the CMKF approach in the algorithm. Furthermore, it was proposed to incorporate a time dependency on the camera range variance. It was shown that this improved the average position RMSE. To compare the algorithm with the mixed-coordinate EKF, both approaches have been implemented. Both algorithms use the same noise covariance matrices. The growth factor is taken as 1.1, since this obtained the best performance on the data set.

The results can be seen in Table 4-3. The time dependent CMKF performs better on the first four scenarios. The mixed-coordinate EKF performs better on the last two scenarios.

This is the scenario in which the target remains relatively close to the Beagle. Therefore, the bias induced by the bearing noise is quite small. Therefore, it is expected that this is the reason the mixed-coordinate EKF outperforms the time-dependent CMKF on these data sets. However, the differences are quite small. To fully conclude on the performance of the algorithms, more data is required. The scenarios should be repeated many times, creating an evaluation similar to Monte Carlo simulations. This could be done by implementing the SAS algorithm into or in cooperation with the simulation software. However, this is not possible due to the export protocol of the software, which is discussed in Subsection 4-7-2. It is not expected that the result of such an evaluation will be highly different, as the sensor noise characteristics are small. Therefore, we can conclude that for this application, either of the two algorithms can be used due to the relatively small differences.

Table 4-3: Position RMSE [m] comparison of time-dependent CMKF and mixed-coordinate EKF

Scenario\Filter	TDCMKF	MCEKF
Scenario 1 - Sea state 1	2.99	3.35
Scenario 1 - Sea state 3	4.43	4.53
Scenario 2 - Sea state 1	4.12	4.45
Scenario 2 - Sea state 3	4.40	4.52
Scenario 3 - Sea state 1	3.74	3.05
Scenario 3 - Sea state 3	3.71	3.17

The synchronization mismatch between the data streams makes it difficult to improve the result much more. It is expected that when the bearing measurement noise of the sensors is larger, the differences would be larger in favor of the time-dependent CMKF, since the bias induced by the bearing noise becomes more significant in this case.

Chapter 5

Conclusion and Recommendations

5-1 Conclusion

Piracy is an increasing problem in certain areas of the world. An alternative is sought for the current protection strategy, which consists of flown-in mercenaries. The alternative proposed by Seastate5, is to equip merchant vessels with an autonomous, unmanned vessel which is capable of non-lethal interception of the pirates. This alternative is called the Beagle. A situational awareness system for such a vessel was designed in this Master's thesis. Multiple methods and approaches have been explored to fulfill the main design goal:

- Design a detection and tracking system for an unmanned surface vessel using radar and vision systems

Some design requirements have been defined, which can be used to reflect on the behaviour and performance of the algorithm. The reflection of the three design requirements is discussed in the following paragraphs.

Adequate detection and tracking - The complete requirement is defined as: *The target vessel should be adequately detected and tracked, even when the target vessel and the Beagle are manoeuvring.* The chosen camera detection algorithm was shown to perform better than the radar in terms of bearing accuracy. Three test scenarios, under two different weather conditions, were generated using simulation software. The scenarios involved actions and target movements expected in a real pursuit, such as: mid-range and short-range targets, up to 180° heading differences of target and Beagle, single or multiple targets, evasive manoeuvring by the target and self manoeuvring. A variety of filter methods was implemented, of which the Interacting Multiple Model (IMM) showed the best performance on the test set. The algorithm obtained a total position RMSE of 5.5m, while also estimating the other states. For a target vessel of 10 meters in length, and beam of around 2 meters, this performance is considered adequate. It is expected that the performance will improve when the data synchronization is done better.

Fluctuating target appearance - With the full requirement: *The system should be robust to fluctuating target appearance due to waves, i.e. it should be able to maintain tracks even if the target does not appear in the sensor data for a period of time.* Due to waves, the target does not always appear in the radar reflection. This is an issue, since the radar is the only sensor providing the range. It was shown that the tracks were maintained, although there was some performance loss. Even for a radar drop-out rate of 50%, the tracks could be maintained. Therefore, the algorithm is considered sufficiently robust against fluctuating target appearance.

Real-time operation - The system should be able to operate in the real world. Therefore, real-time operation is essential. The obtained algorithm update frequency is 12Hz per time step. This is lower than the current update frequency of the camera, which is 15Hz. However, the computation time was measured on a conventional laptop. The easiest approach to meet the requirement is to improve the computational resources. Therefore, real-time operation is considered easily realizable.

The algorithm has been validated on a variety of situations that could take place in case of a piracy threat. Under these situations, all three design requirements are met within acceptable margin, or are easily realizable with the suitable hardware. Therefore, we conclude that the main design goal is fulfilled.

Modifications and Improvements

Next to the design of the algorithm, some modifications and improvements have been made on some of the algorithms and approaches. These are briefly summarized in this subsection.

Data generation - With the developments in automation of ships, proper test and training data seems to be lacking. Field tests for the generation of data are time consuming and costly. Data generation using software can solve these issues. In this thesis, the data is generated using the simulation software Nautis. The situation-specific scenarios provide confidence that the Situational Awareness System (SAS) can also be applied in real-life.

Saliency-based detection - For this application, it is important that the false detection rate does not go up when no target is present. A modification to the BMS method was proposed in this thesis, which decreases false detections in case of target absence. The modification shows a significant improvement in terms of detection precision, especially when the target completely disappears from the image.

Ship motion model - A motion model based on ship dynamics is proposed. Next to the kinematic states, the model is relying on the length and under water volume of the target, which should be observed or estimated from the camera image. Since this physical information was assumed known in this thesis, estimating this is still an open challenge. Also, the influence of the uncertainty on this estimation should be investigated. The ship motion model performed similar to the standard motion models on the position estimation. However, it showed improvement on the average heading and velocity estimation performance. More work and validating should be done, as will be discussed in section 5-2.

Asynchronous measurement update using CMKF - It was investigated whether explicit conversion of the polar coordinates provided by the sensors, would outperform the implicit conversion done in a mixed-coordinate Extended Kalman Filter (EKF). A time dependent range variance was proposed to deal with growing uncertainty in the predicted range in between radar measurements. This time dependency showed an improvement on the average Root Mean Square Error (RMSE) of the 6 data sets. Comparison with the mixed-coordinate EKF showed similar performance. However, the differences are too small to be conclusive. More work should be done, which is discussed in Section 5-2. Because of the unbiasing step in the Converted Measurement Kalman Filter (CMKF), it is expected that the differences between the algorithms will be larger for increasing sensor noise, in favor of the modification. For this application and sensor accuracies, either of the two algorithms can be used.

5-2 Recommendations

In this section, some recommendations for future work are discussed.

Cooperation with Simulation Software

Implementing the algorithm in cooperation with the simulation software allows us to repeat the simulation many times, creating an evaluation similar to Monte Carlo simulations. This way, both the proposed ship model and the TDCMKF can be validated more reliably. However, work has to be done by the company creating the simulation software first, since the exporting protocol should be changed to allow data export of multiple vessels in the same session.

Additional Vision-based Algorithms

Next to the tracking improvement, the combination of the radar and the camera contains more opportunities. The detection algorithm computes a bounding box around the potential target, which is linked to the most probable track by the data association. The amount of pixels in the bounding box is small. Therefore, more vision-based algorithms can be applied relatively computationally cheap, by only processing the matched bounding box area. This creates a lot of opportunities to obtain more information about the target. For instance, the SIFT features discussed in Appendix A could be used to directly estimate the yaw rate. The relative shift of the key points over time can be reconstructed to a turn detection, or even a yaw rate measurement. Similar work was done in [59] and [60] on vehicles. Having direct knowledge about the turning behaviour, increases the confidence that a target is actually turning. This can assist the decision algorithm of the Beagle. The SIFT features can also be used to perform extended object tracking. This way, the length and width of the target can be estimated, which can be used to approximate the under water volume, required for the ship motion model. Other possibilities are to train a classifier on a data set. If the considered bounding box can be confirmed to contain a target, the false positive rate will go down. Furthermore, crucial parts of the target vessel can be identified, such as the engine. This information can be used by the defence mechanism of the Beagle, to target specific elements of the vessel.

Field Tests

The implementation was tested on simulated scenarios. However, extensive field testing should be done to validate the system in a real-life scenario. To do field tests, some work should be done, such as the setup of the required hardware. All sensor values have to be read out in a synchronized manner. Delays in the simulation software caused a temporal mismatch between the data streams. Although delays are also inevitable in a real-life system, it can be investigated whether a synchronization improvement can be made. Furthermore, the camera has to be stabilized. This can be done either digitally or mechanically. During the field tests, the performance can be compared between the standard motion model filter, and the ship motion model filter. This way, it can be shown whether the proposed motion model performs better for real target dynamics.

Appendices

Appendix A

Feature Extraction

In this thesis, only the RGB channels have been used for target detection. This was done to meet the real-time processing requirement. However, it is possible to extract more information from the camera image. This process is called feature extraction. In this chapter, some approaches for feature extraction are discussed.

The color channels are provided by the camera, and can thus be used instantaneously. This makes it computationally efficient, since no further image processing steps have to be done. Extraction of features often employs spatial or temporal information in the image sequence. The extracted features can then be used as extra input to detection or tracking algorithms. On the other hand, most feature extraction approaches are quite computation intensive.

In [13], some features extraction methods are suggested, such as: Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT) features and Haar-based features. These feature extraction methods are based on texture differences. The amount of relevant information certain features provide depends on multiple factors, such as the distance of the object or the environmental conditions. More complex features do not necessarily increase performance. The authors of [61] state that for distant objects, simple, less discriminative features could be sufficient or even better than more complex features. The suggested feature extraction methods are discussed below.

- LBP - The LBP is an operator which aims to describe the local spatial structure of an image [62]. At every pixel position (x, y), the LBP is defined as a set of binary comparisons between the gray scale value of the pixel and its eight surrounding pixels. Essentially, it checks whether the pixel is higher or lower compared to its surrounding pixels. This results in a new pixel intensity. Pixels which are part of an edge, will therefore have the same intensity, while the pixels in a plain area of the image will have a different intensity. This makes it possible to detect local lines and corners, irrespective of shade or illumination changes. The decimal form of the resulting 8-bit value can be found as

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n, \tag{A-1}$$

where the function $s(x)$ is defined as

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}. \tag{A-2}$$

It might happen that a certain part of the image contains an abrupt transition in intensity, an edge. Pixels on this edge will have a different LBP value, compared to the pixels at a distance from the edge. In most situations, this increases the observability of the edge.

- HOG - The Histogram of Oriented Gradients method is a grid based descriptor computed locally [63]. The image is divided into small connected cells. For each cell, a histogram of gradient directions is computed consisting of the gradient directions of each pixel within the cell. Each cell consists of q bins. The cell is normalized to ensure robustness against illumination differences. Figure A-1 shows a plot of HOG features over an image of a vessel. We clearly see the edges of the vessel as large gradient direction. This can be used to recognize or localize the vessel.

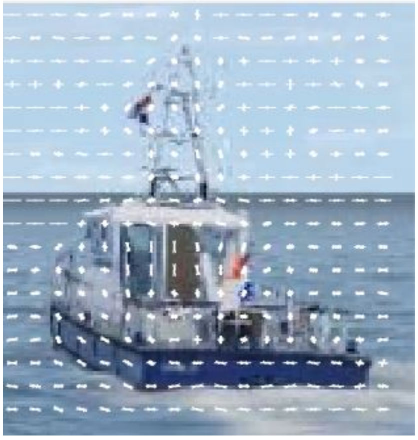


Figure A-1: HOG depiction of a simulated vessel

- Haar-like features - Haar-like features are widely used in face detection [64]. The method scans the image for matches with certain shapes. An example can be seen in Figure A-2. The pixels in the black region are summed, as well as the pixels in the white region. The difference between the two is the output of the feature. The larger the difference, the stronger the feature. Since this results in a very large amount of features, feature selection and cascading is performed by the Adaboost algorithm. Using a training set, the algorithm chooses strong features and cascades them using weights to obtain a strong classifier.

  An issue for the implementation of this approach in our application is that it needs quite a lot of training data. Also, occlusion due to waves or alternative vessel shape are likely to give issues for the computation of a stable Haar-like feature based classifier.

- SIFT features - This approach transforms an image into a large collection of local feature vectors, which are invariant to scaling, rotation, translation, and partially invariant to

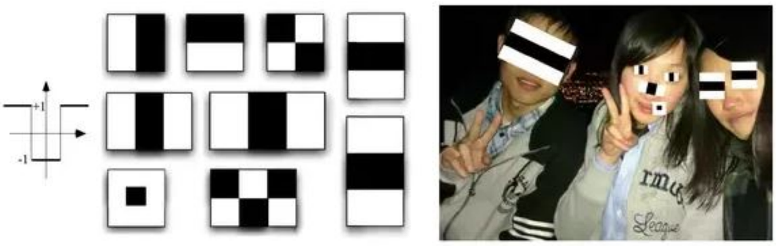


Figure A-2: Example of strong Haar-like features in face detection

illumination changes [65]. The approach searches for "interesting" points in the image, such as high contrast areas like edges. These points are called keys. Locations of keys are found as the minima and maxima of the result of different Gaussian functions in a scale space to a series of smoothed and resampled images. Candidate points which have low contrast, or appear on an edge are discarded to decrease ambiguity. The SIFT keys are stored in a buffer which is matched against new images. Nearest neighbors are identified by means of the Euclidean distance to find a candidate match. The probability of a correct match can be related to the ratio between the distance from the closest nearest neighbor to the second closest. In [65], all matches with a ratio above 0.8 are discarded. Next, the Hough transform is used to identify a cluster of features with a consistent interpretation [66]. Each feature "votes" for all object poses that are consistent with the feature. If a cluster of features is found which vote for the same pose of the object, the probability of the interpretation is increased. To account for scale and rotation, broad bin sizes of 30 degrees are used for orientation, and a factor 2 for scale. As a last step, the keys in the cluster are verified in which a linear least squares problem is solved. The transformation of a model point $[x, y]^T$ to an image point $[u, v]^T$ can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{A-3}$$

where the vector $\begin{bmatrix} t_x & t_y \end{bmatrix}^T$ represents the image translation, and the parameters m_1, m_2, m_3, m_4 represent the rotation, scale and stretch. This can be rewritten to contain the unknown parameters in the state vector, and considering more than one matched key point.

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}. \tag{A-4}$$

For scale changes of about a factor 2, and image rotations of around 30 to 45 degrees, the SIFT approach outperforms other local texture descriptors. However, it is computationally expensive. The computation of SIFT features may take up to, or even more than one second for an entire image. When performed on segmented parts of the image only, it will obviously be much faster. Figure A-3 shows the SIFT match between two images taken from an image sequence in which a target vessel is turning. Therefore, the target appearance slightly changes. It can be seen that the correct points are matched in the two images. It is possible to decrease or increase the threshold to obtain less or more matched points respectively.

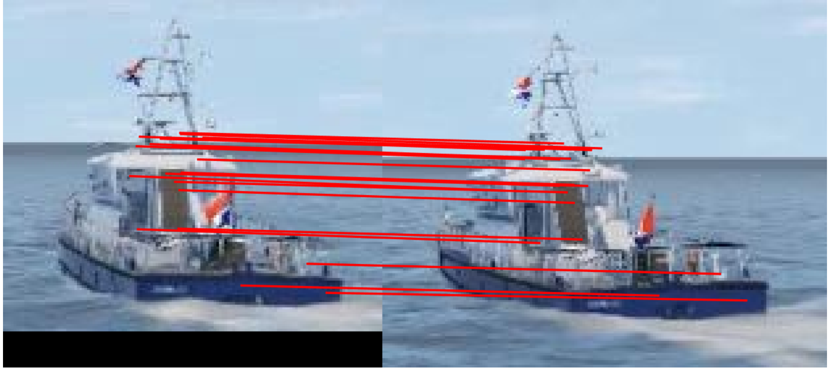


Figure A-3: SIFT match between two parts of image sequence with small time interval

Some features can be extracted after image segmentation, for instance: shape compactness, shape convexity, shape rectangularity or eccentricity. Of course, the extraction of these features requires computation power. Therefore, it might be better to use these features only when the color channels are not sufficient.

Appendix B

Sensor Application

A literature study was done prior to the project, in which multiple sensor methods were evaluated and compared. This gave a clear overview of the sensor advantages and disadvantages. The sensors were evaluated using performance indicators, such as: range, accuracy, condition robustness, cost, and practicality for the final product. The practicality for the final product considers to which extend the method is suited for more than just detecting and tracking the target vessel. The main goal of the overall project is to intercept piracy vessels in a non-lethal manner. To perform this task, it might be useful to know where the pirates and the engine of the vessel are located. Even though the specific interception method is outside the scope of this thesis, it is important to consider it. The evaluation of the different sensors on the performance indicators is summarized in this appendix.

B-1 Thermal Camera

The use of a thermal camera was investigated. In this section, the evaluation on the performance indicators is presented

Price and Range

The amount of available IR cameras is large. Estimating the range performance is difficult, since it depends on a lot of variables. The choice of camera specifications can be used to estimate a detection range. Factors that determine range capability can be seen in Table B-1. Multiple methods are available for estimating the detection range [67].

Cameras and lenses are often compared against the Johnson's criteria for detection recognition and identification [68]. The Johnson's criteria uses the resolving power of a camera as a level of "goodness" for target acquisition purposes. For a certain target to background contrast, resolving power is the highest spatial frequency passed by the sensor visible to the observer.

Table B-1: Factors determining range of thermal cameras

Controllable Factors	Uncontrollable Factors
Task	Weather and atmosphere
- Detection, tracking, recognition, identification - Confidence threshold	- Air condition (transmission, humidity) - Obscurants (rain, mist, fog)
Camera design	Target characteristics
- Optics focal length, f-number, aperture size, aberrations, transmission - Detector pixel pitch, intrinsic sensitivity - Noise suppression, blur compensation, contrast enhancing	- Size - Structure - Temperature - Motion - Emissivity
Placement of camera	Background characteristics
- Distance from camera to target - Location of the horizon in the scene	- Temperature - Texture and clutter

Johnson [69] multiplies the resolving power of the camera by the target size to get "cycles on the target". One of the interpretations which defines Detection, Recognition and Identification (DRI) following from Johnson's criteria results in a number of pixels required on the object to be able to make an accurate assessment.

To get an idea of the price range for thermal cameras, quotas have been requested for some thermal cameras of the brand Flir. Also, a detection range estimation is calculated based on Johnson's criteria, assuming ideal conditions. The given prices are obtained by SensorBV and are supplier recommendations, therefore they might deviate.

Table B-2: Price quotations for certain thermal camera models

Model	Range Estimation (m)	Price Quotation (€)
Flir M232	1050	3000
Flir M625	2000	15.000
Flir M400	15.000	47.000

Accuracy

Most thermal cameras are able to tilt and rotate internally with good angular precision [70]. The camera could be balanced using a gyro-stabilized platform, or by post processing the images. On land, it is likely that the system is very accurate in the determination of the bearing angle. At sea, the camera will start rotating around the rolling axes of the ship. Accuracy of the system highly depends on accuracy of information such as the distance from the camera to the different roll axes of the ship, and the ability to compensate for this. Because of the relatively low resolution of most IR cameras, especially distant target detections will not be accurate.

Condition robustness

Thermal cameras suffer from strong condition dependedness due to signal attenuation caused by obscurants like rain, mist and fog. An often used classification is the one used by the

International Civial Aviation Organisation (ICAO). This system classifies fog in 4 categories. An internal research document from the IR camera company Flir presented the following detection range results with respect to these categories [71].

Table B-3: Detection range (km) given a temperature difference between target and background of 10 °C

Fog category	Visual	IR
Cat I	1.22	5.9-10.1
Cat II	0.61	2.4
Cat IIIa	0.305	0.293
Cat IIIc	0.092	0.087

It can be seen that for a low amount of fog, the detection distance is still relatively large. Rain however, can significantly reduce target contrast. IR system performance degradation due to rain is very range sensitive. In the 100-500 meter range, a large drop in performance occurs, which is quite close to the ship.

In general, piracy vessels do not have Automatic Identification System (AIS) on board and can therefore only localize the ship visually. Therefore, heavy rain will highly obstruct their sight. Also, this will mainly occur in combination with high sea state, making it almost impossible to board a large merchant ship from the side.

Practicality for Final Product

We consider the situation in which the camera is mounted on an unmanned surface vessel which is targeted for non-lethal interception of the hostile pirate vessel. To fulfill this mission, it is useful to identify critical components within the target vessel. Dependent on the type of interception weapon used, is can be beneficial to know where the people in the vessel are located, or where the engine is located. These are both easily detectable using a thermal camera, since they are significantly warmer than their surroundings.

B-2 Visible-light camera

The use of a visible-light camera was investigated. In this section, the evaluation on the performance indicators is presented

Range

Real time obstacle detection on water is performed in [9]. The system operates with 640 x 480 images at about 12 Hz. It is able to detect obstacles the size of a buoy up to 300 meters. The cameras are places about 2 meters above the water. Increasing the resolution of the cameras might increase detection distance. The system is capable of detecting obstacles up to 12 knots (± 22 km/h). Increasing the frame rate will increase this value. Increasing the frame rate and resolution will obviously also increase the computational cost.

In [72], a depth map generation in an urban environment is presented. The resulting processing time on a NVIDIA Jetson TX1 (± €550) is 101.6 ms for a resolution of 3820 x 1080, and

55.6 ms for a resolution of 2560 x 720. This results in a maximum frame rate of around 10 and 18 fps respectively. These results give some idea of the potential increase possibilities in both resolution and frame rate.

Since the size of the target vessel will be bigger than that of a buoy, the detection distance will increase. The range will be a trade-off between computation power, image resolution and frame rate.

Accuracy

For the visible-light camera, the same holds as for the thermal camera. However, in general a visible-light camera has a higher resolution than a thermal camera. This can also be seen in the subsection on the cost of the camera. Therefore, the visible-light camera have a higher potential to be accurate, as long as the processing can be done fast enough to be real-time implemented.

Condition Robustness

Again, the condition robustness can again be read off in Table B-3. Even for Cat I fog, the maximum detection range is 1.22 km. Therefore. the visible light is nearly 5 times less visible in the distance than the IR light. In [73], the authors state that conditions like rain drops or direct sunlight into the camera can decrease detection rates by 15% to 80%. Visible-light cameras are thus highly sensitive to less ideal conditions. However, the same holds for visible-light cameras as for thermal cameras, the pirate vessel will also rely only on visible light detection.

Cost

We consider cameras with a frame rate of 30 fps. To get an indication of the price range, some quotations have been requested. These quotations can be seen in Table B-4. We see that the price of the visible-light cameras is significantly lower than the thermal cameras, while the resolution is a lot higher.

Table B-4: Indication of visible-light security cameras

Model	Resolution	Price Quotation (€)
Vivotek IB836BA-HF3	1920x1080	230
Vivotek IB9371-HT	2048x1536	600
Raymarine CAM200IP	1920x1080	600

The Raymarine camera is dedicated for marine environments and the Vivotek cameras are designed as on-shore security camera. The latter will therefore be more sensitive to the harsh sea conditions.

Practicality for Final Product

The practicality for the final product is comparable to that of the thermal camera. It could also be used to identify critical components in the target vessel. Other than most thermal cameras, most visible-light cameras are able to use optical zoom. This makes it possible to detect and recognize objects at a large distance. The stereo vision capabilities of two or more cameras make it a suitable option for distance estimation.

B-3 Passive Sonar

The use of passive sonar was investigated. In this section, the evaluation on the performance indicators is presented.

Range

Literature describing the use of passive sonar for target tracking mainly focus on either submarine targets, or surface targets measured from the sea bed. No literature was found on the detection of small vessels while measuring from the sea surface.

Also, the initial project description was to construct a passive sonar system to detect targets. Later on in the literature study, the scope has been widened to investigate multiple sensing options in order to select the most promising. Before this, a feasibility study was performed using a three hydrophone setup. Using this hydrophone array, tests were conducted on a nearby lake. It was estimated from the test results that the detection range is about 800 meters. What must be noted, is that this is without movement and engine noise of the measurement vessel. On the other hand, the engine of a piracy vessel will be sufficiently larger compared to the engine used in the test. If the emitted sound intensity is increased with a factor 4, the detection range will still not surpass 1500 meters. Additionally, it is very likely the case that increasing sea state will attenuate the signal significantly, which makes the detection range even smaller.

Accuracy

It was shown in a test on-shore that the bearing can be estimated accurately. However, the range estimates become more inaccurate for increasing range. Also, tests conducted in the water showed that close range measurements are difficult due to the high noise around the target vessel. Therefore, the measurements of targets in mid-range or further are inaccurate.

Condition Robustness

Since we are measuring near the sea surface, the sea state will have an effect on the detection capabilities of the system. Since the noise picked up by the hydrophones will increase, so will the signal strength have to do in order to maintain a high detection rate and range. An advantage is that weather conditions such as fog will not influence the detection distance.

Cost

The cost of the measurement setup parts can be seen in Table B-5. In the setup, only three hydrophones are used. This would imply that the cost of such a system is relatively low. In marine applications, sonar arrays usually consist of hundreds of hydrophones. This also significantly increases frame and data acquisition costs.

Table B-5: Cost of parts in measurement setup

Part	Cost (Euro)
Hydrophone (1x)	120
NI USB-6211	120
Frame per meter	4
Pre-amplifier (1x)	10

Practicality for Final Product

The system should be mounted on the hull of both the merchant ship as the hull of the Beagle. Mounting on the hull of the merchant ship could give issues after some time, since it will be under water for a large amount of time. It should be investigated how fast the sensors will be covered in dirt, plants and sea life. The mounting on the hull of the Beagle could significantly influence the stability and drag of the vessel, since it is travelling at high velocity. Also, it appears that lowering the hydrophones further in the water improves performance. This could influence the stability and drag even more. A positive feature is that the system is submerged, and therefore is difficult to disable by the pirates.

B-4 Radar

The use of radar was investigated. In this section, the evaluation on the performance indicators is presented.

Range

It was decided that an Frequency Modulated Continuous Wave (FMCW) radar system is used for the Situational Awareness System (SAS). This is because of the superior range accuracy compared to a pulse radar. The FMCW radar produced by Rockwell Collins was able to detect a cooperative semi-rigid boat up to about 8 km in sea state 3 [12]. A cooperative jet-ski at 5 km and a kayak at 1.6 km. It must be noted that the radar was standing on the shore and only had a Field Of View (FOV) of 90 degrees. This will improve the detection range. In [74], a buoy is detected at about 2 km from the radar. Although not a lot of literature is available on the detection range of small vessels, the available results seem promising for far-field detection of small targets.

Accuracy

The FMCW radar systems which were found, all have a range accuracy of a few meters. The azimuth accuracy is between 0.6 degrees and 1 degree. At a target distance of 3 km, this results in a radial accuracy of about 30 to 50 m respectively. This means that the range of the systems seem sufficient, but the angular accuracy might not be good enough to navigate the Beagle remote controlled.

Condition Robustness

As described in [11], range performance of radar systems is not largely influenced by rain, or even heavy rain. The same holds for fog and mist. The sea state does influence the performance of radar systems, especially for small vessels. The vessels are not visible since they are hidden behind waves. This makes that their detection fluctuates, making them difficult to distinguish from waves, especially for human operators.

Cost

Table B-6 shows the cost of common commercial FMCW radar systems. It is expected that these will perform less good than the specific purpose FMCW radar system described in this section. However, this still gives some idea of the price range of these systems.

Table B-6: Cost of commercial FMCW radar systems

Model	Cost (€)
Raymarine Quantum	1725
Lowrance 4G Broadband	1640
B&G 4G Broadband	2400

Practicality for Final Product

The radar systems presented provide good resistance against harsh marine environments. One of the drawbacks of a radar system is that its angular accuracy is not sufficient to be used as a stand-alone sensor. Especially since no additional information about the size and critical locations on the target vessel can be obtained. Therefore, the radar system does not provide extra practicality for the final product.

B-5 Lidar

The use of radar was investigated. In this section, the evaluation on the performance indicators is presented.

Range

One of the challenges for autonomous vehicles in the past few years has been to increase the maximum operational velocity. To do this, the vehicle must be able to gather enough information about its surrounding environment to make safe decisions [75]. Since it is desirable to obtain 360 degree FOV of the environment, only these cameras have been considered in the

Table B-7: Maximum detection ranges for different lidar models

Model	Range [m]	Sample rate [10^6/s]
Velodyne HDL-64E S2	120	1.3
Neptec OPAL-ECR	240	0.2
Renishaw Merlin	250	0.036

comparison. Table B-7 shows some maximum detection ranges of lidars on the market.

A lidar used in some works on Unmanned Surface Vessel (USV), is the Velodyne HDL-64E S2 [76][77][78]. This was mainly in obstacle avoidance research. This device is also used on Google's self driving car. However, the maximum detection range makes it difficult for speeds higher than 60 km/h at this point. In the case of oncoming pirate vessels, relative speeds surpassing 120 km/h are not unthinkable. The Neptec Lidar has a more promising detection range, but considerably lacks sample rate and thus point density [75]. This might make it difficult to reliably detect and track targets. Short range, high sample rate lidars might be a good option when the target is localized in the shadow range of the radar.

Accuracy

One of the large advantages of using lidar over other sensors is that they have high accuracy, as can be seen in Table B-8. The angular accuracy of for instance the Velodyne HDL-64E S2 at its maximum detection range results in a horizontal accuracy of about 15cm, which is sufficient for our application.

Table B-8: Range and angular accuracy for different lidar models

Model	Range accuracy [cm]	Angular accuracy [deg]
Velodyne HDL-64E S2	2	0.09
Neptec OPAL-ECR	1.5	0.036
Renishaw Merlin	1	0.01

Condition Robustness

The influence of different weather phenomena on the detection range of automotive lidar was discussed in [79]. Although the tests were conducted with a lidar with less power, the proportions can probably still be assumed to be in the same range. The degradation of the detection range due to increasing amount of rain ranges from 0% to about 50% for rain from 0 mm/h to 10 mm/h respectively. The condition robustness seems to be comparable to the camera robustness.

Cost

The cost of different lidar systems can be seen in Table B-9. Compared to radar systems, this is significantly higher. There also exist lidar systems in the lower price range around

€6000. However, these systems are meant for short range and provide a lower point density. Therefore, they will not be sufficient for our application at this point and will not further be considered.

Table B-9: Cost of different Lidar systems

Model	Cost (€)
Velodyne HDL-64E S2	65.000
Neptec OPAL-ECR	60.000

Practicality for Final Product

The ability to provide excellent accuracy compared to other sensors make that it could be possible to use the lidar data for specific attacks on parts of the target vessel, such as the engine. It should be investigated whether it is possible to distinguish critical parts of the target vessel. The high cost could make it difficult to incorporate the sensor in the final product due to commercial considerations.

B-6 Sensor Fusion

All sensors and approaches presented have advantages and disadvantages. This makes it difficult to use only one sensor to cover all demands. Also, all sensors suffer from measurement noise. Adding more sensors could improve the target position estimation. In fact, multiple sensors can provide important redundancy for perceiving the world around the Beagle [77]. A data fusion module could be implemented which uses the information from all sensors and somehow combines those into the best possible position estimate. Multiple sensor combinations were evaluated. The final result of this evaluation can be seen in Table B-10. Based on this evaluation, it was decided to use a fusion of radar and visible-light camera in this thesis.

Table B-10: Evaluation of sensor combinations

Combination	Range	Accuracy	Condition Robustness	Cost	Practicality for Final Product
Radar and lidar	+	++	+	-	-
Visible-light camera and radar	+	+	+	++	+
Radar, lidar, cameras	+	++	+	- -	+
Visible-light cameras and thermal cameras	-	-	-	+	+
Lidar and cameras	- -	++	- -	-	+

Appendix C

Detailed Results

In this appendix, the tracking results are presented in more detail. The results of the different filter models are presented in Table C-1. The best performing filter on a certain scenario is highlighted in green.

Table C-1: Results for different filter models, corresponding to Figure 4-20, 4-21, and 4-22 respectively

	KF		**EKF**		**IMM**		
Scenario\Filter	CV-NC	CV-C	CTCV-NC	CTCV-C	CTCV-NC	CTCV-C	CVCR-C
Position [m]							
Scenario 1 - Sea state 1	10.83	6.43	7.61	4.72	5.30	2.99	2.87
Scenario 1 - Sea state 3	7.22	4.85	14.65	5.57	5.36	4.43	5.04
Scenario 2 - Sea state 1	33.28	15.00	19.87	5.69	14.55	4.12	4.05
Scenario 2 - Sea state 3	33.04	14.53	19.54	6.35	16.41	4.40	4.67
Scenario 3 - Sea state 1	9.16	4.32	10.03	3.71	3.97	3.74	3.76
Scenario 3 - Sea state 3	9.18	4.15	5.55	3.30	4.50	3.71	3.63
Heading [rad]							
Scenario 1 - Sea State 1	0.282	0.225	0.183	0.199	0.193	0.198	0.143
Scenario 1 - Sea state 3	0.254	0.193	0.274	0.268	0.212	0.188	0.161
Scenario 2 - Sea state 1	0.260	0.245	0.151	0.293	0.148	0.187	0.134
Scenario 2 - Sea state 3	0.258	0.247	0.160	0.256	0.215	0.173	0.153
Scenario 3 - Sea state 1	0.288	0.222	0.347	0.179	0.178	0.163	0.175
Scenario 3 - Sea state 3	0.293	0.239	0.234	0.173	0.174	0.157	0.169
Velocity [m/s]							
Scenario 1 - Sea state 1	2.37	2.78	0.94	1.47	0.76	1.65	0.78
Scenario 1 - Sea state 3	2.29	2.44	1.40	2.64	0.90	2.04	0.83
Scenario 2 - Sea state 1	4.15	3.79	3.20	3.54	2.69	2.10	2.01
Scenario 2 - Sea state 3	4.17	4.05	2.96	2.60	3.42	1.65	1.64
Scenario 3 - Sea state 1	1.68	1.41	1.30	1.13	1.00	1.13	1.12
Scenario 3 - Sea state 3	1.54	1.41	1.07	1.03	0.73	0.94	0.98

Bibliography

[1] J. Tian, "Small infrared target fusion detection based on support vector machines in the wavelet domain," *Optical Engineering*, vol. 45, no. 7, 2006.

[2] A. Shack, "One Dimensional Mixture." http://www.aishack.in/tutorials/expectation-maximization-gaussian-mixture-model-mixtures/.

[3] G. Bishop, "An Introduction to the Kalman Filter." https://www.cs.unc.edu/{~}welch/media/pdf/kalman{_}intro.pdf, 2001.

[4] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design," *2015 IEEE Intelligent Vehicles Symposium (IV)*, no. Iv, pp. 1094–1099, 2015.

[5] J. Liu, R. Hekkenberg, E. Rotteveel, and H. Hopman, "Literature review on evaluation and prediction methods of inland vessel manoeuvrability," *Ocean Engineering*, vol. 106, pp. 458–471, 2015.

[6] A. Friedman, "U.S Navy Seal - Somali Pirates." https://www.youtube.com/watch?v=1fgoZXjB9ys, 2010.

[7] *SOLAS - International Convention for the Safety of Life at Sea.* 1974.

[8] D. Hermann, R. Galeazzi, J. C. Andersen, and M. Blanke, "Smart sensor based obstacle detection for high-speed unmanned surface vehicle," *International Federation of Automatic Control*, vol. 28, no. 16, pp. 190–197, 2015.

[9] H. Wang, Z. Wei, S. Wang, C. S. Ow, K. T. Ho, B. Feng, and Z. Lubing, "Real-time obstacle detection for Unmanned Surface Vehicle," *Defense Science Research Conference and Expo (DSR), 2011*, pp. 1–4, 2011.

[10] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Annual Reviews in Control Unmanned surface vehicles : An overview of developments and challenges," *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016.

[11] IALA, *Preperation of Operational and Technical Performance Requirements.* 2015.

[12] J. P. Wasselin and S. Mazuel, "FMCW radar system for detection and classification of small vessels in high sea state conditions," *Proceeding of the 42nd European Microwave conference*, pp. 29–32, 2012.

[13] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment : A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[14] "PETS2016." http://www.cvg.reading.ac.uk/PETS2016/a.html.

[15] R. J. Rajesh and P. Kavitha, "Camera gimbal stabilization using conventional PID controller and evolutionary algorithms," *IEEE International Conference on Computer Communication and Control, IC4 2015*, pp. 1–6, 2016.

[16] M. S. H. Adeel Yousaf, Khurram Khurshid , Muhammad Jaleed Khan, "Real Time Video Stabilization Methods in IR Domain for UAVs – A Review," *2017 Fifth International Conference on Aerospace Science & Engineering (ICASE)*, vol. 2017.

[17] A. Heya, K. Hirata, and N. Niguchi, "Dynamic Modeling and Control of Three-Degree-of-Freedom Electromagnetic Actuator for Image Stabilization," *IEEE Transactions on Magnetics*, vol. PP, pp. 1–5, 2018.

[18] C. Zhao, A. Sain, Y. Qu, Y. Ge, and H. Hu, "Background Subtraction based on Integration of Alternative Cues in Freely Moving Camera," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–14, 2018.

[19] S. Brutzer, H. Benjamin, and G. Heidemann, "Evaluation of Background Subtraction Techniques for Video Surveillance," *CVPR 2011*, 2011.

[20] J. Zhang and S. Sclaroff, "Exploiting Surroundedness for Saliency Detection: A Boolean Map Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 889–902, 2016.

[21] S. Arivazhagan and L. Ganesan, "Automatic target detection using wavelet transform," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 17, p. 853290, 2004.

[22] T. Cane and J. Ferryman, "Saliency-Based Detection for Maritime Object Tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1257–1264, 2016.

[23] D. Bloisi and L. Iocchi, "Argos— A Video Surveillance System For Boat Traffic Monitoring In Venice," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 7, pp. 1477–1502, 2009.

[24] D. B. Rubin, A. P. Dempster, and N. M. Laird, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal ofthe Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[25] A. Mittal, N. Paragios, and C. Marne, "Motion-Based Background Subtraction using Adaptive Kernel Density," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 0–7, 2004.

[26] Q. Zhu, "Dynamic Video Segmentation via a Novel Recursive Kernel Density Estimation," *2011 Sixth International Conference on Image and Graphics Dynamic*, 2011.

[27] M. Grinberg, *Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking.* 2017.

[28] P. C. Mahalanobis, "On the generalized distance in statistics," National Institute of Science of India, 1936.

[29] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A Study of a Target Tracking Algorithm Using Global Nearest Neighbor," *International Conference on Computer Systems and Technologies*, 2003.

[30] J. P. Snyder, *Flattening the Earth: Two Thousand Years of Map Projections.* 1993.

[31] M. Buhren and B. Yang, "A Global Motion Model For Target Tracking In Automotive Applications," *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pp. 313–316, 2007.

[32] W. Yi, L. Kong, J. Yang, and X. Deng, "A tracking approach based on dynamic programming track-before-detect," *Radar Conference, 2009 IEEE*, pp. 1–3, 2009.

[33] L. Stone, R. Streit, and T. Corwin, *Bayesian multiple target tracking - Second Edition.* Artech House Radar Series, Boston [Massachusetts]: Artech House, 2014.

[34] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide-Baseline Stereo From Maximally Stable Extremal Regions," *Image and Vision Computing*, vol. 22, no. 10 SPEC. ISS., pp. 761–767, 2004.

[35] C. Feichtenhofer and A. Pinz, "Spatio-Temporal Good Features To Track," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 246–253, 2013.

[36] P. M. Djuric, "Particle filtering," *IEEE Signal Processing Magazine*, pp. 271–331, 2003.

[37] Z. Shu, G. Liu, Z. Xie, and Z. Ren, "Real Time Target Tracking Based on Nonlinear Mean Shift and Particle Filters," *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, 2017.

[38] W. Sun, C. Zhao, L. Chen, D. Li, Y. Bai, and W. Jia, "Optik Learning based particle filtering object tracking for visible-light systems," *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 19, pp. 1830–1837, 2015.

[39] A. F. Genovese, "The Interacting Multiple Model Algorithm for Accurate State Estimation of Maneuvering Targets," *John Hopkins APL Technical Digest*, vol. 22, no. 4, 2001.

[40] V. J. X. Rong Li, "Survey of Maneuvering Target Tracking . Part I : Dynamic Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, 2003.

[41] D. Laneuville, "Polar versus cartesian velocity models for maneuvering target tracking with imm," in *2013 IEEE Aerospace Conference*, pp. 1–15, March 2013.

[42] D. F. Sebastian Thrun, WolfRam Burgard, *Probabilistic Robotics*. 2005.

[43] A. Zomotor, *Fahrwerktechnik: Fahrverhalten*. Vogel-Fachbuch: Kraftfahrzeugtechnik, 1991.

[44] Y. Wang, "Host – Target Vehicle Model-Based Lateral State Estimation for Preceding Target Vehicles Considering Measurement Delay," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4190–4199, 2018.

[45] M. A. Abkowitz, "Lectures on Ship Hydrodynamics - Steering and Manoeuvrability," tech. rep., 1964.

[46] K. Nomoto, T. Taguchi, K. Honda, and S. Hirano, "On the steering qualities of ships," *International Shipbuilding Progress*, vol. 4, no. 35, pp. 354–370, 1957.

[47] E. M. Lewandowski, *The Dynamics of Marine Craft : Maneuvering and Seakeeping*. 2004.

[48] S. B. Denny and E. Hubble, "Prediction of craft turning characteristics," *Marine Technology*, vol. 28, pp. 1–13, 1991.

[49] *Rules for Classification of Ships/High Speed, Light Craft and Naval Surface Craft*. Det Norske Veritas (DNV), 2011.

[50] K. Granstr, M. Baum, and S. Reuter, "Extended Object Tracking : Introduction, Overview and Applications," *Journal of Advances in Information Fusion*, pp. 1–30, 2016.

[51] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, 2001.

[52] D. F. Bizup, "The Over-Extended Kalman Filter - Don't Use It!," *Proceedings of the Sixth International Conference of Information Fusion*, 2003.

[53] D. Lerro, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, pp. 1015–1022, 1993.

[54] A. Karsaz and N. Pariz, "Coordinates Transformation for Target Maneuver Detection," *2007 IEEE International Conference on Control and Automation*, pp. 2978–2983, 2007.

[55] S. V. Bordonaro, *Converted Measurement Trackers for Systems with Nonlinear Measurement Functions*. PhD thesis, 2015.

[56] J. Domhof, R. Happee, and P. Jonker, "Multi-Sensor Object Tracking Performance Limits by the Cramer-Rao Lower Bound," *20th International Conference on Information Fusion*, pp. 0–7, 2017.

[57] "ICC Piracy Armed Robbery Report 2019." https://www.icc-ccs.org/piracy-reporting-centre/live-piracy-report.

[58] J. Stastny, S. Cheung, G. Wiafe, K. Agyekum, and H. Greidanus, "Application of RADAR Corner Reflectors for the Detection of Small Vessels in Synthetic Aperture Radar," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 3, pp. 1099–1107, 2015.

[59] A. Barth and U. Franke, "Tracking Oncoming and Turning Vehicles at Intersections," *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 861–868, 2010.

[60] M. Gabb, A. Kaliuk, T. Ruland, and L. Otto, "Feature-Based Monocular Vehicle Turn Rate Estimation from a Moving Platform," *2013 IEEE Intelligent Vehicles Symposium*, pp. 0–5, 2013.

[61] B. Van den Broek and P. Schwering, "Persistent Maritime Surveillance Using Multi-Sensor Feature Association And Classification," *Proceedings of SPIE - The International Society for Optical Engineering*, no. May, 2012.

[62] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, July 2002.

[63] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE Computer Society, 2005.

[64] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings International Conference on Image Processing*, vol. 1, pp. I–I, IEEE, 2002.

[65] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. of the International Conference on Computer Vision, Corfu*, 1999.

[66] D. H. Ballard, "Generalizing the Hough Transform To Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1980.

[67] J. Love, "The Truth about range data How to assess thermal camera range capability for site design purposes The Truth About Range Data : How to assess thermal camera range capability for site design purposes."

[68] AIS, "Long-range surveillance cameras & Johnson's criteria." http://www.aissecuritysolutions.com/white-paper-on-long-range-surveillance-cameras.pdf.

[69] W. K. Weihe, "Classification and Analysis of Image-Forming Systems," *Proceedings of the IRE*, vol. 47, no. 9, pp. 1593–1604, 1959.

[70] A. Sorbara, A. Odetti, M. Bibuli, E. Zereik, and G. Bruzzone, "Design of an obstacle detection system for marine autonomous vehicles," *MTS/IEEE OCEANS 2015 - Genova: Discovering Sustainable Ocean Energy for a New World*, 2015.

[71] FLIR, "Seeing through fog and rain with a thermal imaging camera," p. 4, 2005.

[72] Y. chin Tsai, K. hung Chen, Y. Chen, and J.-h. Cheng, "Accurate and Fast Obstacle Detection Method for Automotive Applications Based on Stereo Vision," *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 18–21, 2018.

[73] D. M. Gavrila and V. Philomin, "Real-Time Object Detection For Smart Vehicles," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

[74] A. Morska and R. Wawruch, "Comparative Analysis of the Pulse and FMCW Radars Detection Possibilities," *Scientific Journals Maritime University of Szczecin*, vol. 29, no. 101, pp. 158–163, 2012.

[75] A. Schneider, Z. Lacelle, A. Lacaze, K. Murphy, and R. Close, "Results and Conclusions: Perception Sensor Study for High Speed Autonomous Operations," *Proceedings of the SPIE*, vol. 9859, 2016.

[76] J. F. C. Sauzé, *Robotic Sailing - Proceedings of the 4th International Robotic Sailing Conference.* 2012.

[77] L. Elkins, D. Sellers, and W. R. Monach, "The Autonomous Maritime Navigation (AMN) Project : Field Tests, Autonomous and Cooperative Behaviors, Data Fusion, Sensors, and Vehicles," *Journal of Field Robotics*, vol. 27, no. 6, pp. 790–818, 2010.

[78] J. Han, J. Kim, and N.-s. Son, "Persistent Automatic Tracking of Multiple Surface Vessels by Fusing Radar and Lidar," *OCEANS 2017 - Aberdeen*, pp. 2–6, 2017.

[79] R. H. Rasshofer, M. Spies, and H. Spies, "Radio Science Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science*, vol. 9, pp. 49–60, 2011.

List of Acronyms

IMO	International Maritime Organization
SAS	Situational Awareness System
AIS	Automatic Identification System
GMM	Gaussian Mixture Model
SB	Saliency-Based
FPS	Frames Per Second
KDE	Kernel Density Estimation
DRI	Detection, Recognition and Identification
ICAO	International Civial Aviation Organisation
IMU	Inertial Measurement Unit
DC	Digital Compass
SNR	Signal to Noise Ratio
RaDAR	Radio Detection And Ranging
FMCW	Frequency Modulated Continuous Wave
FOV	Field Of View
LiDAR	Light Detection And Ranging
SoNAR	Sound Detection And Ranging
PDF	Probability Density Function
USV	Unmanned Surface Vessel
HOG	Histogram of Oriented Gradients

EKF	Extended Kalman Filter
DGPS	Differential Global Positioning System
LBP	Local Binary Pattern
SIFT	Scale Invariant Feature Transform
EM	Expectation-Maximization
IMM	Interacting Multiple Model
RMSE	Root Mean Square Error
CV	Constant-Velocity
CT	Coordinated-Turn
CTCV	Coordinated-Turn, Constant-Velocity
CR	Constant-Rudder
CMKF	Converted Measurement Kalman Filter
STD	Steady Turning Diameter

List of Symbols

Abbreviations

$\dot{r}$	Range rate
$\tilde{\mu}$	State switching probability
α	Bearing
λ	Model probability
μ	Mean
ω	Yaw rate
ψ	Heading
σ	Standard deviation
A	State transition matrix
C	Measurement matrix
K	Kalman Gain
P	Error covariance
p	State switching matrix
Q	Process noise covariance matrix
R	Measurement noise covariance matrix
r	Range
t	Time
v	Velocity
x	Position in eastern direction
y	Position in northern direction
z	Observation