School of Electrical Engineering and Computing

A Vision System for Mobile Maritime Surveillance Platforms

Thomas Albrecht

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

May 2012

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

Author	Date

A Vision System for Mobile Maritime Surveillance Platforms

by

Thomas Albrecht

Submitted to the School of Electrical Engineering and Computing
in December, 2011 in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Mobile surveillance systems play an important role to minimise security and safety threats in high-risk or hazardous environments. Providing a mobile marine surveillance platform with situational awareness of its environment is important for mission success. An essential part of situational awareness is the ability to detect and subsequently track potential target objects.

Typically, the exact type of target objects is unknown, hence detection is addressed as a problem of finding parts of an image that stand out in relation to their surrounding regions or are atypical to the domain. Contrary to existing saliency methods, this thesis proposes the use of a domain specific visual attention approach for detecting potential regions of interest in maritime imagery. For this, low-level features that are indicative of maritime targets are identified. These features are then evaluated with respect to their local, regional, and global significance. Together with a domain specific background segmentation technique, the features are combined in a Bayesian classifier to direct visual attention to potential target objects.

The maritime environment introduces challenges to the camera system: gusts, wind, swell, or waves can cause the platform to move drastically and unpredictably. Pan-tilt-zoom cameras that are often utilised for surveillance tasks can adjusting their orientation to provide a stable view onto the target. However, in rough maritime environments this requires high-speed and precise inputs. In contrast, omnidirectional cameras provide a full spherical view, which allows the acquisition and tracking of multiple targets at the same time. However, the target itself only occupies a small fraction of the overall view. This thesis proposes a novel, target-centric approach for image stabilisation. A virtual camera is extracted from the omnidirectional view for each target and is adjusted based on the measurements of an inertial measurement unit and an image feature tracker. The combination of these two techniques in a probabilistic framework allows for stabilisation of rotational and translational ego-motion. Furthermore, it has the specific advantage of being robust to loosely calibrated and synchronised hardware since the fusion of tracking

and stabilisation means that tracking uncertainty can be used to compensate for errors in calibration and synchronisation. This then completely eliminates the need for tedious calibration phases and the adverse effects of assembly slippage over time.

Finally, this thesis combines the visual attention and omnidirectional stabilisation frameworks and proposes a multi view tracking system that is capable of detecting potential target objects in the maritime domain. Although the visual attention framework performed well on the benchmark datasets, the evaluation on real-world maritime imagery produced a high number of false positives. An investigation reveals that the problem is that benchmark data sets are unconsciously being influenced by human shot selection, which greatly simplifies the problem of visual attention. Despite the number of false positives, the tracking approach itself is robust even if a high number of false positives are tracked.

Acknowledgements

Firstly, I would like to thank my supervisors A/Prof Tele Tan and Prof Geoff West for their valuable advise and guidance throughout my research. Their detailed comments as well as their encouraging and personal guidance have been a great contribution for the success of this thesis. In addition, I would also like to thank Dr Thanh Ly from the Australian Defence Science and Technology Organisation for his commitment to this project. DSTO supported this work and provided me with a scholarship, which is hereby gratefully acknowledged.

From the School of Engineering, I wish to thank Patrick, Simon, and Philipp for many good and constructive discussions. Their perceptive comments were a major factor for the shape and clarity of papers and this thesis. Furthermore were the frequent and extensive lunch and coffee breaks always a welcome distraction.

All my friends deserve a big thank you for their support and encouragement throughout the past years. In particular, I would like to thank Suzy for taking the tremendous task of proofreading this thesis.

I would like to thank Natalie for her love and patience and for making everything worthwhile.

Lastly, and most importantly, I wish to thank my parents and my brüdi for their faith and unlimited support to me.

Copyright Acknowledgements

Imagery downloaded from the internet has been used in this chapter for testing purposes in accordance with Section 40 of the Australian Copyright Act of 1968, which allows fair dealing of artistic work for purpose of research or study. The following persons are herewith acknowledged as the respective copyright holders:

Test image in Figure 2.9 is ©Immen.

Test image in Figure 4.4(a) is ©Walter Quirtmair. Test image in Figure 4.5(a) is ©Aragami. Test image in Figure 4.6(a) is ©David Biagi. Test images in Figures 4.1, 4.3, 4.7(a), and 4.19(d) are ©Immen. Test images in Figures 4.9(a), 4.12(d), and 4.15(i) are ©unknown. Test images in Figures 4.9(b) and 4.19(g) are ©Mark Daly. Test images in Figures 4.9(c) and 4.17(f) are ©Chris Howel. Test images in Figures 4.9(d) and 4.19(i) are ©Waterboys. Test image in Figures 4.9(e) is ©Geert van Kesteren. Test image in Figure 4.9(f) is ©Willem Kroon. Test images in Figures 4.9(g),4.10, 4.12(c), 4.15(e), 4.17(c), and 4.17(e) are ©Ken Smith. Test images in Figures 4.9(h) 4.17(a) are ©Tom Gulbrandsen. Test images in Figures 4.9(i) 4.15(b) are ©Charlie Chambers. Test images in Figures 4.12(a) and 4.19(h) are ©Martin. Test image in Figure 4.12(b) is ©Jos Telleman. Test image in Figure 4.12(e) is ©Ray Smith. Test image in Figure 4.12(f) is ©Joochen Wegener. Image in Figure 4.15 is ©Phil English. Test images in Figures 4.12(g) and 4.15(c) are ©Glen Kasner. Test images in Figures 4.12(h) and 4.15(f) are ©Gerolf Derbes. Test images in Figures 4.12(i) and 4.17(d) are ©Marc Piche. Test images in Figures 4.15(a) and (h) are ©VAV. Test images in Figures 4.15(d) and 4.17(h) are ©Emily8. Test image in Figure 4.17(g) is ©E. Vroom. Test images in Figures 4.17(i) and 4.19(e) are ©Juan Carlos C. Test image in Figure 4.19(a) is ©Jarrod David. Test image in Figure 4.19(b) is ©Angelov. Test image in Figure 4.19(c) is ©Benoir Donne. Test image in Figure 4.19(f) is ©jacek.

Test images in Figures 5.1 and 5.2 is ©Immen. Test image in Figure 5.12(a) is ©Juan Carlos C. Test image in Figure 5.12(b) is ©Lukasz Blaszczak. Test image in Figure 5.12(c) is ©Cornelia Klier. Test image in Figure 5.12(d) is ©Captain Peter. Test image in Figure 5.12(e) is ©Alain Fierens. Test images in Figures 5.16(a) and 5.16(d) are ©Geert van Kesteren. Test image in Figure 5.16(b) is ©Dragec. Test image in Figure 5.16(c) is ©E. Vroom. Test image in Figure 5.16(e) is ©Frank Schlunsen. Test image in Figure 5.16 is ©West Jp.

Contents

List of Figures

List of Tables

Published Work

This thesis is based on the following works, published over the past three years:

- Albrecht, T. and Tan, T. and West, G.A.W. and Ly, T. (2010). Omnidirectional Video Stabilisation On a Virtual Camera Using Sensor Fusion. *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision.*

- Albrecht, T. and Tan, T. and West, G.A.W. and Ly, T. and Moncrieff, S. (2011). Vision-based Attention in Maritime Environments. *Proceedings of the 8th International Conference on Information, Communications and Signal Processing.*

- Albrecht, T. and West, G.A.W. and Tan, T. and Ly, T. (2011). Visual Maritime Attention Using Multiple Low-Level Features and Naïve Bayes Classification. *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications* (Best Student Paper Award).

- Albrecht, T. and West, G.A.W. and Tan, T. and Ly, T. (2010). Multiple Views Tracking of Maritime Targets. *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications.*

Chapter 1

Introduction

In high-risk, hazardous, inaccessible, or remote areas, mobile surveillance systems play an important role to minimise security and safety threats. They can navigate around obstacles and follow potential target objects. When equipped with an omnidirectional camera, they can cover a vast area and ensure full 360° situational awareness while minimising the need for significant infrastructure. Fixed cameras and pan-tilt-zoom (PTZ) cameras, which have traditionally been used for surveillance, have only limited spatial range and small fields of view (FOV). The range of these cameras can be enhanced by using lenses with a high zooming factor, but this then reduces the field of view further. To a certain extent, this can be compensated for by using high resolution image sensors, although this then demands more processing power. Especially in open outdoor areas, highly dynamic changes within the line of sight need to be taken into account. An approach that is being investigated by several research groups is the installation of multiple fixed and/or PTZ cameras (Javed and Shah, 2008; Devarajan et al., 2008; Soto et al., 2009; Utsumi et al., 2009). However, this approach is only valid as long as a target is within the field of view of the camera network. Furthermore, obstacles blocking the line of sight of a camera are not uncommon. To overcome these constraints, mobile platforms are needed.

One important subset of mobile platforms are maritime surveillance platforms, which can be deployed for port surveillance or monitoring of coastal areas. However, the maritime environment introduces additional challenges to the camera system. While any outdoor sensor system is subject to environmental disturbances, maritime platforms face significant and continual perturbation; gusts, wind, swell, or waves can cause the platform to move drastically, causing severe ego-motion and consequently substantial and chaotic changes in the camera's view. One solution is to equip the mobile platform with an omnidirectional camera to ensure that the platform will not lose sight of a target, but this will also require neutralising the effects of ego-motion, a process called "stabilisation". A traditional approach for stabilisation is the use of hydraulic or electro-mechanic tilt platforms on which the camera system is mounted (Masten, 2008). The platform is tilted based on the measurements of an accelerometer or gyroscope, contained within an Inertial Measurement Unit (IMU) in order to keep the camera's view unchanged despite the motion of

the host platform. However, this approach typically requires highly accurate calibration and synchronisation between the IMU and digital imaging system to ensure the IMU's corrections are applied to the right frames and transformed accurately to the camera's coordinate system. Whilst possible, such calibration requires precise assembly that must be durable enough to prevent any physical shifting of the components which would cause de-calibration, a difficult task in rough environments. Moreover, if translational disturbances are present as well, the target's bearing will change. To compensate for this, the translational component needs to be estimated. However, affordable IMUs, implemented as microelectromechanical systems (MEMS), cannot measure translational motion reliably due to significant errors introduced by doubling the integration of the accelerometer readings. Digital image registration has been used to resolve this by registering features over time, finding the optimum affine transformation that minimises the error between features, and applying it to subsequent frames (Battiato et al., 2007; Yang et al., 2009). With significant ego-motion, this can be computationally expensive as the search window must be very large so as to not risk losing the target.

However, in many situations, it is not feasible to process the entire omnidirectional view, as the actual region of interest at any one time typically only occupies a small fraction of the overall view. Selecting a region of interest in omnidirectional video is a laborious and even confusing task for a human operator. While a narrow field of view camera delivers imagery analogous to the human vision system, an omnidirectional camera provides the operator with a panoramic view of the entire scene at once. While this theoretically is an advantage that can provide full situational awareness, the human vision system in fact has limitations when it comes to perception beyond the typical attentive field (Pashler, 1995, 1999), risking oversight of potentially dangerous targets. To deal with this problem, it is important to automate this process by using early processing stages to direct attention where further investigation is needed. An automated system that highlights and extracts candidate regions to the operator as well as neglects insignificant parts of the image can therefore be used to reduce the workload and increase the efficiency of a human operator. Computer vision disciplines that are related to this are: image complexity (Peters and Strickland, 1990), object detection (Lampert et al., 2008; Felzenszwalb et al., 2009; Everingham et al., 2010; Alexe et al., 2010), saliency (Itti et al., 1998; Liu et al., 2007; Achanta et al., 2009; Achanta and Süsstrunk, 2010), and visual attention (Sun and Fisher, 2003; Hu et al., 2008; Frintrop et al., 2010). While image complexity refers to the algorithmic complexity of detecting objects within an image, object detection is concerned with the finding of specific objects in an image; it is typically task driven. Saliency, on the other hand, describes areas that are distinctive within the image. Finally, visual attention can be seen as a pre-attentive phase in a vision system. It reacts to low-level stimuli and is used to focus further processing on a region with high response. The use of *a priori* scene

knowledge and thus tuning the system for the expected scenes can be used to improve detection accuracy.

This thesis develops a framework that consists of two parts: First, a stabilisation module is developed that allows fusion of omnidirectional camera and IMU using a probabilistic model that is specifically designed to achieve accurate rotational and translational stabilisation despite only rough calibration and synchronisation. The framework uses IMU measurements as an initial guess and refines the estimations using an image registration method. This way, the two components do not need to be in a permanent configuration but can be quickly assembled without the need to recalibrate the system. This, for example, allows the use of hardware which is not rigidly connected in a single housing but is easily assembled on a *per mission* basis. Furthermore, because the stabilisation process puts the target in the centre of the stabilisation process, it allows for rotational and translational disturbances and allows simultaneous and independent stabilisation and tracking of multiple targets. Secondly, a visual attention framework is developed that allows early detection of regions of interest in maritime scenes to be tracked. The framework uses domain specific knowledge to improve accuracy over generic detectors. Domain knowledge is essential for differentiating between relevant and irrelevant parts of a scene. While salient detectors are only concerned with the presence of low-level stimuli, a task-specific description allows guiding visual attention towards parts of the scene and neglect regions that are salient but irrelevant in terms of the task. The proposed framework is tuned to maritime scenes but has the potential to be applied to any domain by selecting the appropriate features or by retraining. The two frameworks are eventually fused and it is demonstrated that the stabilisation and visual attention approaches have the potential to allow detection and robust tracking of multiple objects in a real world omnivision maritime scenario that is unfiltered by the human shot selection bias present in most saliency-style datasets.

1.1 Aims and Approach

This thesis is concerned with the research into and development of algorithms for a vision system of an unmanned maritime surveillance platform. The system is oriented towards the development of a target detection and tracking function in fully autonomous vehicles, though it may also be used to aid a human operator in target detection and threat evaluation. The objectives of this thesis are:

1. The development of an image stabilisation approach that allows for robust stabilisation of omnidirectional imagery in challenging maritime outdoor conditions despite

loose calibration and synchronisation.

2. The development of an early processing stage that is capable of directing visual attention to candidate regions of interest in maritime imagery.

3. The implementation of a multi target tracking method that utilises the proposed visual attention framework to detect and track multiple moving objects simultaneously in omnidirectional imagery captured by a moving maritime surveillance platform.

The first aim is addressed by combining an omnidirectional camera with an IMU in a probabilistic sensor fusion approach. From the omnidirectional camera, a region with limited field-of-view is extracted, forming a virtual camera. Stabilisation is achieved by continuously adjusting the orientation of the virtual camera based on measurements of the IMU and an image feature tracker. The system is successful despite weak calibration of the relative locations of the IMU and camera and imprecise synchronisation of IMU and video frames.

To identify regions of interest, a visual attention framework is proposed that combines domain specific low-level features using multiple distance measurements. The approach is then extended using machine learning and a domain specific background segmentation technique to further improve detection performance, and shown to outperform state-of-the-art non-domain specific approaches for detection.

Based on the previous findings for stabilisation and visual attention, a multi-view tracking approach is developed that uses independent virtual cameras extracted from omnidirectional imagery. The system is tested on a very challenging omnidirectional video captured from a fast-moving boat. Objects of interest (nearby boats) are automatically discovered by a visual attention detector and subsequently tracked with a very high degree of stabilisation despite the significant motion of the camera. However, findings show that benchmark datasets to evaluate visual attention provide little indication of eventual performance in real world footage, and the problem is traced to the influence of human shot selection in the datasets.

1.2 Significance and Contribution

This thesis makes three main contributions in the field of sensor fusion and computer vision:

1. The use of virtual cameras for omnidirectional image stabilisation allowing multiple target tracking using loosely calibrated and synchronised hardware.

2. The development of a domain specific visual attention framework that can be used to detect areas that are important in terms of *scene* rather than *image*.

3. The evaluation of the proposed approaches on a community standard dataset (Liu et al., 2007), a domain specific dataset assembled from imagery contributed by the general public, and a very challenging real world data set.

The contributions and their significance are detailed in the following.

1.2.1 Image Stabilisation Using Virtual Cameras

When combining IMU and omnidirectional camera, both devices need to be synchronised to ensure that measurements are taken at the same time instant. Combination also requires calibration, that is estimating the transformation between the two devices so that measurements can be converted between both coordinate systems. Existing image stabilisation approaches that utilise IMUs to measure camera disturbance require precise (typically hardware-based) synchronisation and rigid coupling to prevent shifting and subsequent recalibration. Without this, the IMU is not able to measure motion at the same time instant where the camera image was taken due to the latency between sensors. Depending on the situation, the subsequent stabilisation process would suffer from an offset, or worse, a "stabilisation" in the wrong direction, actually worsening the process. Furthermore, the need for calibration is an essential disadvantage for temporary assembled units as disassembling and reassembling requires recalibration every time. The sensor fusion approach proposed in this thesis allows for robust image stabilisation without the need for precise calibration or synchronisation.

While current approaches stabilise scenes by applying transformations to the camera images, i.e. using the camera coordinate system as the reference frame, the proposed approach puts the target at the centre of the stabilisation process. This is important for two reasons: firstly, target-centred stabilisation allows the creation of independently stabilised views for multiple targets. Secondly, and most importantly, in a scene where target objects vary in distance from the platform, a scene based stabilisation approach will fail as the difference in distance causes significantly different motion in the projected camera image. This is also true if a near object is to be stabilised in front of a distant background (parallax). Different stabilisation parameters are therefore required for different viewing

directions or objects.

The proposed approach is significant because:

- It requires only an approximate calibration between the hardware components, making it ideally suitable for quick assembly of components.
- Sensor drift, as well as wear and tear of the components is part of the design and does not affect the stabilisation system, therefore does not require constant recalibration.
- It proposes the use of independent virtual cameras that allow for independent stabilisation of multiple target objects.
- The use of target-centred instead of camera-centred coordinates allows stabilisation robust to both rotation and translation.

1.2.2 Domain Specific Visual Attention

This thesis focuses on the problem of detecting regions of interest in the scene from omnidirectional views of maritime environments. Such scenes contain a variety of potentially salient information, such as vessels, coast, or boat wake. Which of these is important depends on the application domain that the saliency is used for. Thus generic saliency detectors are a poor choice and in fact a goal-oriented visual attention approach is more appropriate since saliency is a description of simple low-level features without any relation to the domain whilst visual attention lifts the concept to a higher level and tries to address the scene rather than the pixels. Hence, the proposed approach utilises features that are specifically designed to respond to visual attention in the domain. This way, only regions of the image that "stand out" with respect to the domain are detected.

The proposed approach is significant because:

- It utilises *a priori* scene knowledge of the maritime domain (or a domain) to detect regions in a scene that are important in terms of visual attention rather than saliency, an approach that cannot be made without domain specific designs, and shows that this will outperform state-of-the-art approaches.
- It provides a visual attention framework that should be suitable to any domain – while the presented application is trained for the maritime domain, it may be

adapted into any domain by selecting appropriate features and/or retraining on an adequate dataset.

1.2.3 Evaluation using Real World Data

It is essential to evaluate algorithms that are developed for outdoor scenes on real-world data. While standard datasets are a fundamental part of quantitative evaluation and comparison of algorithms, a good performance on a dataset does not guarantee that it is suitable to be used in real-world conditions. Especially for omnidirectional cameras, lighting conditions of outdoor settings are challenging as the omnidirectional camera captures *all* aspects of the scene. Thus, issues arise such as facing towards and away from the sun at the same time. Furthermore, the current standard saliency dataset (Liu et al., 2007) is assembled by researchers and is shown to be unconsciously biased due to the human-photographer's influence on shot selection and image choice. While it contains a variety of object types, all objects are salient as per definition and at prominent position in the image (see Chapter 6 for details). This greatly simplifies the problem of detection, in contrast to a deployed autonomous system which will need to deal with data captured at atypical locations in unforeseen circumstances where objects are likely to be small and at random locations relative to the entire scene.

The proposed evaluation is relevant because:

- It demonstrates the effectiveness of the stabilisation framework in a challenging outdoor environment with significant disturbances present.
- It evaluates the visual attention framework on unfiltered data, where humans have not been able to select views or shots.
- It shows that even though training is built on an independent dataset, it is able to successfully detect objects of interest with a strong response to those objects in real world data without the need for retraining.

1.3 Thesis Outline

The goal of the research presented in this thesis is the development of computer vision algorithms that support operators of mobile maritime surveillance platforms. The remainder of this thesis is organised as follows:

This chapter, Chapter 1, provided the reader with an introduction into the field, outlined the research questions that will be addressed in this thesis, and emphasised the significance of this thesis.

In Chapter 2 background information relevant to this thesis are discussed. Different types of omnidirectional camera systems are derived and the concept of inertial measurement units and their components is explained. The chapter continues with a discussion about related work in tracking and visual attention. The chapter concludes with the introduction of evaluation methods.

Chapter 3 deals with methods for stabilisation of omnidirectional camera systems. Related algorithms will be described and an approach that is robust and overcomes deficiencies in loosely calibrated systems is proposed and evaluated.

Chapter 4 is devoted to a saliency inspired attention detector, tuned for maritime scenes. Generic algorithms are evaluated on maritime imagery. For comparison the proposed detector is also tested on a standard dataset and it is shown that it gives comparable results to state of the art generic detectors.

In Chapter 5 the proposed visual attention framework is extended with a further domain specific classifier and tested on a domain specific dataset. A detector that outperforms generic approaches in challenging maritime scenes is proposed.

The proposed frameworks for visual attention and omnidirectional image stabilisation are combined to track multiple targets in real-world maritime environments in Chapter 6.

Chapter 7 concludes the research and outlines open research questions for future work.

Chapter 2

Background

The scope of this thesis involves a number of different computer vision and engineering principles. This includes camera calibration, inertial sensor systems, sensor fusion, image stabilisation, tracking, visual attention, and machine learning. An overview of the current state of the art in each of the respective fields and reviews of the relevant research conducted is given in this chapter. The chapter also familiarises the reader with some necessary mathematical and physics background.

The chapter is organised as follows: Section 2.1 introduces the notation and different coordinate systems used in this thesis. This is followed by a description of omnidirectional camera systems and inertial sensors in Sections 2.2 and 2.3 respectively. Section 2.4 gives an overview of image tracking techniques, which are utilised for image stabilisation and multi target tracking in Chapter 3 and 6 respectively. An introduction to visual attention and an overview of related work in this area is given in Section 2.5, which is the main topic of Chapter 4 and Chapter 5. Section 2.6 familiarises the reader with machine learning techniques used for feature combination and classification in Chapter 5. This is followed by a brief review of colour models in Section 2.7 and an overview of the evaluation process in Section 2.8. The chapter concludes with a summary given in Section 2.9.

2.1 Coordinate Spaces

This thesis contains references to different coordinate systems arising from the fusion of multiple sensors, each with their own coordinate system. Thus, this section will introduce the various coordinate systems and define the notation and transformations between coordinate systems for the rest of the thesis.

2.1.1 Homogeneous Coordinates

Homogeneous coordinates (Ballard and Brown, 1982) allow non-linear transforms to be carried out in a projective space using standard matrix operations. Let $X = (X_1, X_2, X_3)$ be a point in $\mathbb{R}^3$, then the vector $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \lambda)^T$ with $\tilde{x}_1 = \lambda X_1$, $\tilde{x}_2 = \lambda X_2$, $\tilde{x}_3 = \lambda X_3$, where $\lambda \in \mathbb{R}$ and $\lambda \neq 0$ is called its homogeneous coordinate. This means that $\tilde{\boldsymbol{x}}$ represents a local vector to the very same point X for any λ. In other words, the point X in $\mathbb{R}^3$ is actually represented by the line, $\tilde{\boldsymbol{x}}$, in projective space, $\mathbb{R}^4$. While λ can be chosen arbitrarily, typically $\tilde{\boldsymbol{x}}$ is normalised such that $\lambda = 1$:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} = \frac{1}{\lambda} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \lambda \end{pmatrix} \tag{2.1}$$

Unless stated otherwise, normalisation is assumed when referring to homogeneous coordinates in the remainder of this thesis.

In the following, the notation of Craig (2005) is adopted: The orientation of a coordinate system, $\{A\}$, with respect to another coordinate system, $\{B\}$, is denoted as the 3×3 rotation matrix, ${}^A_B\mathbf{R}$. The 3×1 column-vector ${}^A_B\boldsymbol{t}$ is the translational offset of $\{A\}$ with respect to $\{B\}$. Both can be combined using homogeneous coordinates resulting in the 4×4 transformation matrix ${}^A_B\mathbf{T}$ as

$${}^A_B\mathbf{T} = \begin{pmatrix} {}^A_B\mathbf{R} & {}^A_B\boldsymbol{t} \\ 0 & 1 \end{pmatrix}. \tag{2.2}$$

The inverse transformation is subsequently defined as

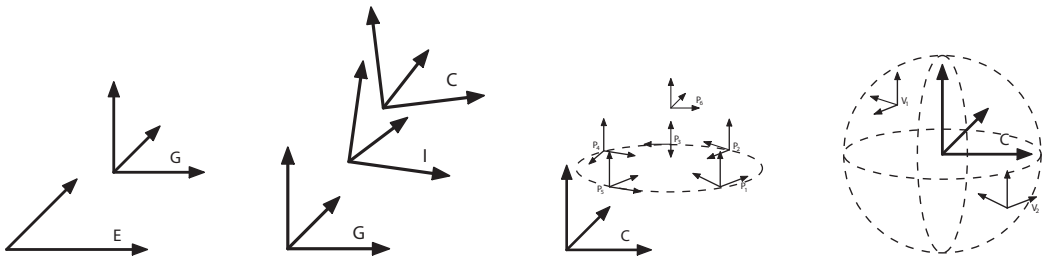
$$({}^A_B\mathbf{T})^{-1} = {}^B_A\mathbf{T}. \tag{2.3}$$

The coordinates of a point in $\mathbb{R}^3$ are only valid in conjunction with a reference coordinate system. To indicate this, the homogeneous coordinate of a point is always defined in terms of the underlying coordinate system. If ${}^A_B\mathbf{T}$ is the transformation of $\{A\}$ with respect to $\{B\}$, then ${}^A\boldsymbol{p}$ and ${}^B\boldsymbol{p}$ are the homogeneous coordinates for the very same point in $\mathbb{R}^3$:

$${}^A\boldsymbol{p} = {}^A_B\mathbf{T}\ {}^B\boldsymbol{p}. \tag{2.4}$$

2.1.2 Coordinate Systems

Within this thesis, a number of coordinate systems are used (Figure 2.1). The 2D position of the mobile platform is measured by the GPS receiver. Its output is given in *Earth Coordinates* as longitude and latitude. At the current position of the platform, *Global Coordinates* is a sphere centred on the platform that spans a right-handed coordinate system, which is aligned with North and the circles of latitude, effectively representing the omnivision sphere of view. The IMU outputs the orientation of *Inertial Coordinates*, with respect to Global Coordinates. The six perspective cameras of the *Ladybug* camera system (see Section 2.2.4) acquire images in *Perspective Camera Coordinates* that are defined for each of the cameras individually. These are then mapped into a unified *Camera Coordinate System* with its origin in the centre of the omnidirectional camera. The alignment between the IMU and *Ladybug* is denoted by the transform between Inertial Coordinates and *Camera Coordinates*. Finally, for each Virtual Camera (see Section 2.1.2.6), *Virtual Camera Coordinates* are defined by a transform with respect to Camera Coordinates. This section introduces the different coordinate systems in detail and gives transformations that allow converting between them.



(a) Earth and Global Coordinate Systems

(b) Global, Inertial, and Camera Coordinate Systems

(c) Perspective Camera Coordinate System

(d) Virtual Camera Coordinate System

Figure 2.1: Coordinate Systems

2.1.2.1 Earth Coordinate System

Exact computations on the earth's surface can be complex. In this thesis, we are dealing with close range distances within the line of sight (typically a couple of hundred meters), which allows us to adopt a flat earth approximation. The vicinity of a fixed reference point (Φ_0, λ_0) on the earth's surface can be approximated using a planar projection, resulting in a mapping where the circles of latitude and the lines of longitude are equidistant, straight

and cross at right angles (Snyder, 1987). As the circumference of the circles is dependent on Φ_0, the length of a radian and the radius of the curvature are computed as functions of the reference latitude as $r'(\Phi_0)$ and $r''(\Phi_0)$ respectively (Snyder, 1987). For the parameters of the equatorial radius and flattening of the earth, the World Geodetic System (WGS84) by the US Department of Defense (2000) is used. A point $(\Phi\,, \lambda)$ at sea level altitude can be expressed in respect to a reference point $(\Phi_0,\, \lambda_0)$ in $\{E\}$ as

$$ {}^{E}\boldsymbol{p} = \begin{pmatrix} r'(\Phi_0) & 0 \\ 0 & r''(\Phi_0) \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Phi - \Phi_0 \\ \lambda - \lambda_0 \end{pmatrix}. \tag{2.5} $$

Earth Coordinates are used for tracking multiple maritime objects in Chapter 6.

2.1.2.2 Global Coordinate System

The global coordinate system, $\{G\}$, is a projection of earth coordinates, $(\Phi,\, \lambda)$, onto the unit sphere with its origin (sphere centre) at the current position of the platform. The y-axis is aligned with the line of longitude, λ, and pointing towards North. A point in $\{E\}$, ${}^{E}\boldsymbol{p}$ can be projected into $\{G\}$ by computing its spherical angles

$$ \theta = \tan^{-1} \frac{{}^{E}p_2}{{}^{E}p_1} \qquad\qquad \phi = \cos^{-1} \frac{{}^{E}p_3}{\|{}^{E}\boldsymbol{p}\|} \tag{2.6} $$

and then mapping it onto the unit sphere:

$$ {}^{G}\boldsymbol{p} = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}. \tag{2.7} $$

This transformation will be denoted as ${}^{G}_{E}\mathbf{T}$ in the remainder of this thesis. The stabilisation described in Chapter 3 is based on global coordinates.

2.1.2.3 Inertial Coordinate System

The inertial coordinate system, $\{I\}$, is defined with respect to $\{G\}$. Its orientation is measured by the IMU at every time step t, denoted as the homogeneous transformation

${}^{G}_{I}\mathbf{T}_t$. The point ${}^{G}\boldsymbol{p}$ in $\{G\}$ can be transformed into $\{I\}$ as:

$$ {}^{I}\boldsymbol{p} = {}^{I}_{G}\mathbf{T}_t \, {}^{G}\boldsymbol{p}. \tag{2.8} $$

Chapter 3 will show how the transform is determined at each time step t.

2.1.2.4 Camera Coordinate System

The camera coordinate system is defined with the origin in the centre of the omnidirectional camera, i.e. the camera's viewpoint. The transformation of $\{C\}$ with respect to $\{I\}$ is denoted as ${}^{C}_{I}\mathbf{T}$. Note that this transformation is constant as the camera and IMU are rigidly connected. An estimate is formed by observing a horizontal pattern with the camera and gravity by the IMU in static poses (Hol et al., 2010). In contrast to Hol et al. (2010), no subsequent optimisation is performed but a probabilistic approach is used instead allowing the used of loosely synchronised hardware, see Chapter 3 for details. A point in $\{G\}$ can be expressed in $\{C\}$ as:

$$ {}^{C}\boldsymbol{p} = {}^{C}_{G}\mathbf{T}_t \, {}^{G}\boldsymbol{p}. \tag{2.9} $$

2.1.2.5 Perspective Coordinate System

A perspective camera coordinate system, $\{P_{n=1\ldots6}\}$, is defined for each of the six perspective cameras of the Ladybug camera system. ${}^{C}_{P_n}\mathbf{T}$ describes the transformation between $\{P_n\}$ and $\{C\}$. Remember that the origin of $\{C\}$ is the shared viewpoint of all perspective cameras. In accordance with the pinhole camera model, ${}^{C}_{P_n}\mathbf{T}$ is also called the extrinsic parameters of the perspective cameras. The Ladybug camera system is pre-calibrated and ${}^{C}_{P_n}\mathbf{T}$ is provided by the manufacturer. The conversion of a point in $\{G\}$ to $\{P_n\}$ is given as:

$$ {}^{P_n}\boldsymbol{p} = {}^{P_n}_{G}\mathbf{T}_t \, {}^{G}\boldsymbol{p}. \tag{2.10} $$

2.1.2.6 Virtual Camera Coordinate System

This thesis extracts rectangular views from the omnidirectional camera system that mimicks traditional pan-tilt-zoom cameras for target tracking and visualisation. These views are referred to as a *virtual camera*, see Chapter 3 for further details. A virtual camera

coordinate system, $\{V_n\}$, where n is the index of the virtual camera is defined at the centre of each virtual camera with respect to C. ${}^{V_n}_{C}\mathbf{T}$ describes the transformation from $\{C\}$ to the virtual camera $\{V_n\}$ at time step t. $\{V_n\}$ spans a right handed coordinate system on the unit circle with the origin at the centre of the virtual camera, that is ${}^{C}_{V_n}\mathbf{R} * (1,\ 0,\ 0)^T$. A point in $\{G\}$ can be expressed in $\{V_n\}$ as:

$$ {}^{V_n}\boldsymbol{p} = {}^{V_n}_{G}\mathbf{T}_t\ {}^{G}\boldsymbol{p}. \tag{2.11} $$

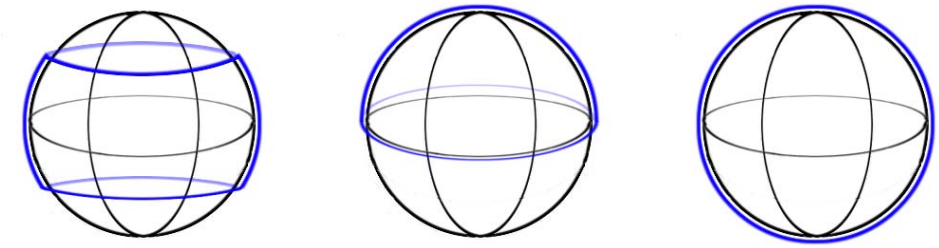
2.2 Omnidirectional Vision

Omnidirectional cameras overcome the restrictions of limited field of view of perspective cameras and are able to capture an entire scene from a single viewpoint. The name "omnidirectional vision" is used as an umbrella term for cameras with three different types of field of view (Figure 2.2):

(a) *Panoramic cameras* that cover 360° in the horizontal but do not provide full coverage of the top or bottom parts of the sphere.

(b) *Half-spherical cameras* that cover an entire hemisphere, i.e. 360° in the horizontal and 180° in the vertical.

(c) *Full-spherical cameras*, that cover an entire sphere, i.e. 360° in the horizontal and 360° in the vertical. The field of view of these cameras is often referred to as 720°. Most of these cameras, however, have a small restriction in the field of view due to the camera mounting.



(a) Panoramic (b) Half-Spherical (c) Full-Spherical

Figure 2.2: Field of view of omnidirectional cameras.

2.2.1 Types of Omnidirectional Cameras

Catadioptric cameras as shown in Figure 2.3(a) are widely used to create panoramic images using a single camera (Nayar, 1997; Yagi, 1999; Svoboda and Pajdla, 2000; Geyer and Daniilidis, 2000). A catadioptric camera consists of a curved mirror that is attached in front of the camera. The mirror reflects the lightrays coming from all directions towards the camera sensor. Due to the curving, catadioptric cameras have a higher resolution in the centre of the image than in the periphery. While these cameras provide full 360° view in the horizontal, they only have limited field of view in the vertical due to the mountings blocking the field of view at the poles, this type of camera can be used to capture panoramic images.

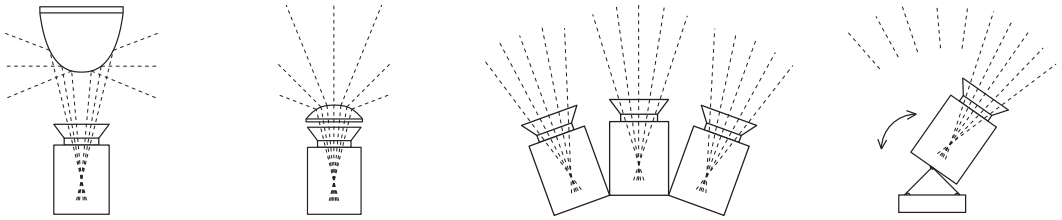
A half-spherical view can be produced by using a single camera equipped with a wide angle lens, e.g. fisheye lenses with a short focal length (Slater, 1996; Schneider et al., 2009). The field of view of these systems is dependent on the optical characteristics of the lens but cannot exceed a hemisphere as inherent to the optical principle of the lens, see Figure 2.3(b).

Both catadioptric and wide angle cameras cannot be calibrated using the standard camera model (Heikkila and Silven, 1997) but require non-linear calibration methods (Faugeras et al., 2004). Further drawbacks of single camera approaches are limited image resolution and inflexibility when it comes to different lighting conditions. This is an important issue with omnidirectional cameras used outdoors as lighting conditions can significantly differ depending on direction.

Another approach for creating a panoramic image is the simultaneous use of multiple perspective cameras that are aligned around a single viewpoint as shown in Figure 2.3(c), e.g. reported by Sato et al. (2004) or the commercially available *Ladybug 2* camera system used in this thesis. In contrast to catadioptric or wide angle lens cameras, they allow adjustment of parameters, such as shutter speed, for each camera individually, which is particularly important in outdoor settings, as there can be different ambient lighting conditions in different directions. Instead of special lenses or catadioptric systems, a set of perspective cameras that are aligned in a ring around a single point of view is used in the *Ladybug 2* enabling it to capture a full spherical view from a single viewpoint.

A related technique to create an omnidirectional image are mosaicked panoramas captured using PTZ-cameras. Here, images are acquired using pan and tilt movements over time (Sinha and Pollefeys, 2006). Even though these techniques allow a full omnidirectional

field of view, they do not provide instantaneous views in all directions, they are therefore not considered in this thesis



(a) Catadioptric (b) Wide-angle (c) Multi Camera (d) PTZ

Figure 2.3: Types of omnidirectional cameras.

2.2.2 Omnidirectional Mappings

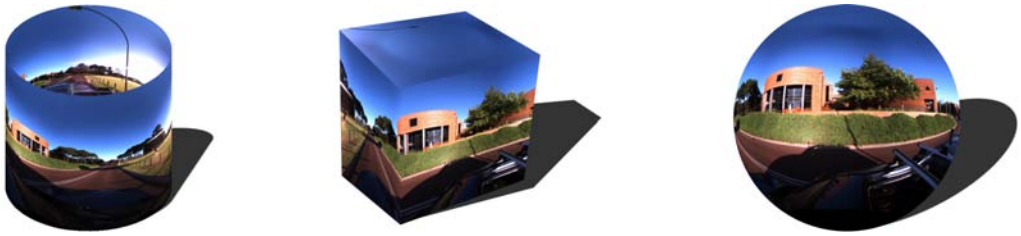
The two most common 2D representations of omnidirectional imagery are the log-polar and panoramic mappings (Salomon, 2006). Log-polar (Figure 2.4(a)) offers a high resolution at the centre of the image, which decreases logarithmically towards the image borders. It is mostly used for wide angle cameras, as these cameras provide the same resolution characteristics due to their optics. The panoramic mapping (Figure 2.4(b)), on the other hand, is easier to grasp for a human as the image seems less distorted. Yet in fact the panorama representation only provides mappings near the equatorial line and becomes inaccurate towards the poles. This is of less concern when cameras are used that only capture a panoramic view, so a catadioptric camera with limited vertical field of view is a reasonable choice. Figure 2.4 shows these representations.

(a) Log-polar (b) Panoramic

Figure 2.4: Types of 2D panoramic mappings

For full spherical cameras the distortions can become drastic and 3D mappings are required. There are three common types of 3D mapping (Yagi, 1999): cylinder, cube or sphere. An obvious representation is the cylindrical mapping as shown in Figure 2.5(a). On closer inspection, while it wraps, it is in effect a rolled up panorama and suffers from

the same distortions as the 2D panoramic mapping. A second representation is the cubic representation (Figure 2.5(b)), which maps the omnidirectional image onto six sides of a cube with 90° separation. It has the advantage that the images on each side are rectified and not distorted, but it does suffer from drastic distortions on the cube borders. The third common mapping is the spherical mapping (Figure 2.5(c)), which maps the captured scene onto a unit-sphere. Because it represents the scene as it was captured, it does not suffer from any distortions and it represents the omnidirectional image in a continuous coordinate space, which is essential for the image stabilisation approach proposed in Chapter 3. Hence this thesis utilises the spherical mapping approach.



(a) Cylindrical (b) Cubic (c) Spherical

Figure 2.5: Types of 3D panoramic mappings

2.2.3 Perspective Camera Model

The relationship between a point in 3D world coordinates (X, Y, Z) and its projection point onto a 2D plane (u, v) for the individual cameras can be described using the Thales theorem as

$$u = \frac{X}{Z} \qquad v = \frac{Y}{Z}. \tag{2.12}$$

Introducing the focal length, f, as the distance between the projection plane and the optical centre of the camera models the perspective camera:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{2.13}$$

This ideal camera model does not hold in the real world; lens distortions and misalignments of sensors, etc. need to be taken into account. This process is called rectification of the image. The relationship between distorted (u, v) and rectified ($\tilde{u}$, $\tilde{v}$) pixel coordinates can

be expressed as

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \tag{2.14}$$

where $\mathbf{K}$ is defined as the camera matrix (Heikkila and Silven, 1997) containing the intrinsic parameters of the camera.

$$\mathbf{K} = \begin{pmatrix} f_x & \alpha\, f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.15}$$

f_x and f_y are the focal length in x and y directions, while α represents the rotational offset between both axes. (c_x, c_y) is the principal point of the image plane. Standard programs are available to estimate the camera matrix, e.g. the Camera Calibration Toolbox for Matlab (Bouguet, 2004).

2.2.4 The Ladybug Camera Model

The Ladybug camera system consists of six perspective cameras. While the cameras are identical in construction, slight differences in manufacturing are taken into account by calibrating each perspective camera individually, resulting in a set of intrinsic parameters, $\mathbf{K}_{1,\ldots,6}$, according to Equation (2.15). Equations (2.13) and (2.14) yield a projection of a world coordinate (X, Y, Z) onto a rectified pixel coordinate ($\tilde{u}$, $\tilde{v}$). Note that in images, pixel coordinates are given with the origin in the upper left corner and the positive x-axis to the right and down of the image. Then, with the image size $w \times h$, the same coordinate can be expressed in the right handed 3D perspective camera coordinate system as

$${}^{P_n}\boldsymbol{p} = \begin{pmatrix} \tilde{u} \\ -\tilde{v} \\ 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -w \\ h \\ 0 \end{pmatrix}. \tag{2.16}$$

All cameras of the Ladybug 2 camera system are aligned around a single viewpoint. Therefore, it is reasonable to select the viewpoint as the origin of a joint coordinate system. An extrinsic calibration identifies the transformation from the perspective camera coordinate system into a joint camera coordinate system. The Ladybug 2 camera is pre-calibrated, and the manufacturer provides the extrinsic parameters as a translation and rotation

between the joint coordinate system and the individual perspective cameras as in:

$$ {}^{C}_{P_n}\mathbf{T} = \begin{pmatrix} {}^{C}_{P_n}\mathbf{R} & {}^{C}_{P_n}\boldsymbol{t} \\ 0 & 1 \end{pmatrix}. \tag{2.17} $$

A pixel coordinate (u, v) in perspective camera P_n can thus be converted into global coordinates as

$$ {}^{G}\boldsymbol{p} = {}^{G}_{C}\mathbf{T}\ {}^{C}_{P_n}\mathbf{T} \left[\begin{pmatrix} \mathbf{K}_n & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ -v \\ 0 \\ 0 \end{pmatrix} + \tfrac{1}{2} \begin{pmatrix} -w \\ h \\ 0 \\ 0 \end{pmatrix} \right], \tag{2.18} $$

where $\mathbf{K}_n$ is the camera matrix and $w \times h$ is the image size of the $n - th$ perspective camera.

2.3 Inertial Measurement Unit

Knowing the position and orientation of a mobile platform is crucial, especially when it is unmanned. While surfaced, a satellite aided navigation systems such as GPS can reliably measure the position and, over time, the velocity of the platform's vehicle. However, in order to stabilise a camera system, accurate information about the vehicle's precise orientation and motion is needed. An inertial measurement unit, which can measure the rate of turn and acceleration, can reliably estimate inter-frame motion (Lawrence, 1998; Everett, 1995). Typically, the sensors of an IMU are only sensitive to one axis, so that three orthogonally aligned sensors are used to provide full three dimensional orientation. IMUs are combined sensors, comprising gyroscopes, accelerometers, and magnetometers, which are described in detail in the following sections. The sensors are typically fused with an Extended Kalman Filter (Maybeck, 1979), yielding the sensor orientation in global coordinates (Corke et al., 2007), ${}^{I}_{G}\mathbf{R}_t$, at time step t.

2.3.1 Gyroscope

Gyroscopes of two types are commonly used: Strap-down systems make use of the conservation of angular momentum in a mechanical setup. Microelectromechanical based systems (MEMS), on the other hand, make use of the Coriolis effect that is induced by

forces acting on vibrating or oscillating structures.

A mechanical gyroscope can best be described as a disc, mounted in gimbals, that is spinning with a constant angular velocity, $\omega = const.$ Any mass particle, m_i, located on the disc has an angular momentum of

$$\mathbf{L}_i = m_i \, (\mathbf{r}_i \times \mathbf{v}_i), \tag{2.19}$$

where $\mathbf{r}_i$ is the position vector of the particle with respect to the centre of the disc and $\mathbf{v}_i$ is its velocity vector. With $\mathbf{v}_i = \omega \mathbf{r}_i$ for each particle, the overall angular momentum of the spinning disc, $\mathbf{L}$, can thus be written as

$$\mathbf{L} = \omega \int m_i \mathbf{r}_i^2 \, di. \tag{2.20}$$

Because

$$\frac{\mathrm{d}\,\mathbf{L}}{\mathrm{d}\,t} = \mathbf{r} \times \mathbf{F}_{ext}, \tag{2.21}$$

the angular momentum is preserved if no external force, $\mathbf{F}_{ext}$, is applied. Equation (2.21) also means that if a force perpendicular to $\mathbf{L}$ is applied, a torque $\tau = \mathrm{d}\,\mathbf{L}/\mathrm{d}\,t$ can be observed and results in a rotation around an axis in direction of $\tau \times \mathbf{L}$, called precission, which is proportional to the projection of the angular velocity causing $\mathbf{F}_{ext}$. Optical or capacitive sensors are used to measure the angular velocity or period of precession in strap-down inertial systems.

A vibrating structure gyroscope is based on an oscillating structure, e.g. a quartz crystal. The structure is mounted in a plane and vibrates with a frequency ω – typically defined by the frequency of the AC voltage, $V_{AC} = \hat{V} \sin(\omega t)$. If a force perpendicular to the axis of oscillation is applied, the Coriolis force causes a precession around the axis of oscillation according to Equation (2.21). In a MEMS setup, a piezo element acts as the vibrating structure.

MEMS gyroscopes are based on the Coriolis effect. A proof mass, m, is placed in a rotating inertial frame with a constant angular velocity, ω. The velocity vector, $\mathbf{v}$, of the mass is perpendicular to the position vector, $\mathbf{r}$, originating at the centre of the frame. Because $\omega = const$, only the direction, not the magnitude of $\mathbf{v}$ changes over time. For the proof mass, a Coriolis force, $\mathbf{F}_C$, can be observed as

$$\mathbf{F}_C = 2m\mathbf{v} \times \omega. \tag{2.22}$$

Accordingly, the Coriolis acceleration, $\mathbf{a}_C$ is

$$\mathbf{a}_C = 2\mathbf{v} \times \omega. \tag{2.23}$$

In a MEMS vibrating structure gyroscope, a proof mass is suspended within a poly-silicium frame and brought to oscillation. Any external force acting perpendicular to the axis of oscillation induces a Coriolis acceleration that can be measured using changes in the capacitive behaviour of the proof mass.

A slightly different approach is used for a MEMS wheel. Here, a micro-structure of a classic spinning wheel is build out of a capacitive material in MEMS technology. The wheel is rotated with a constant velocity. When an external force is applied perpendicular to the rotation axis, the magnitude of the angular moment of the wheel, $\mathbf{L} = I \times \omega$, does not change because $\omega = const.$ However, the direction of $\mathbf{L}$ changes and the wheel keeps its rotation axis perpendicular to the applied force. This tilting can be measured as a change in the capacitive behaviour of the wheel.

The rotation measured by the gyroscope $\tilde{\omega} = (\tilde{\omega}_1, \tilde{\omega}_2, \tilde{\omega}_3)$ can be expressed as a rotation matrix by computing the matrix exponential on the skew symetric matrix:

$$\tilde{\mathbf{R}} = \exp \begin{pmatrix} 0 & -\tilde{\omega}_3 & \tilde{\omega}_2 \\ \tilde{\omega}_3 & 0 & -\tilde{\omega}_1 \\ -\tilde{\omega}_2 & \tilde{\omega}_1 & 0 \end{pmatrix}. \tag{2.24}$$

When the orientation of the sensor from the previous timestep is known, the current orientation can be computed using $\tilde{\mathbf{R}}$:

$$\mathbf{R}_t = \tilde{\mathbf{R}}\, \mathbf{R}_{t-1}. \tag{2.25}$$

2.3.2 Accelerometer

Acceleration can be measured by determining the displacement of a proof body using the spring equation of Hooke's law. When a proof body is connected to a reference body through a spring, the displacement, $\mathbf{d}$, caused by a force, $\mathbf{F}$, can be computed via

$$\mathbf{F} = -k\,\mathbf{d}, \tag{2.26}$$

where k is the spring constant. The force can be computed using the mass of the proof body, m, and the vector of acceleration, $\mathbf{a}$, as

$$\mathbf{F} = m\,\mathbf{a}. \tag{2.27}$$

When the sensor is held static, the proof body is only subject to gravity, $\mathbf{g}$, as an acceleratic force. Solving Equations (2.26) and (2.27) for $\mathbf{g}$, the gravity vector can be estimated as

$$\mathbf{g} = -k\,\mathbf{d}\,m^{-1}. \tag{2.28}$$

For use in MEMS, two types of sensors are suitable: capacitive and piezoresistive. On a capacitive accelerometer, a conductive structure is used as a proof body such that the displacement causes a change in the capacitive characteristics of the sensor. For piezoresistive accelerometers, materials that change their resistivity characteristics when strained are used. Physical strain on semiconductive material causes a change of the band gap, which results in a change of resistivity that can be measured.

Acceleration is the second derivative of position. Thus, the change of position, $\Delta\mathbf{s}$, can be formed by double integration of the measured acceleration over time:

$$\Delta\mathbf{s} = \iint a_t \; dt. \tag{2.29}$$

A position estimate, $\mathbf{s}_t$, can now be computed, when a reference position, $\mathbf{s}_0$, at timestep, t_0, is known:

$$\mathbf{s}_t = \mathbf{s}_0 + \int_{t_0}^{t}\int_{t_0}^{t} a_t \; dt. \tag{2.30}$$

However, due to the errors induced by bias or drift of the sensor as well as errors due to double integration, accelerometers need to be very precise when used for position estimation. Typically, the precision of inexpensive MEMS sensors is not sufficient for this task.

2.3.3 Magnetometer

The earth's geodynamo generates a magnetic field with the lines of force originating in the southern and finishing in the northern hemisphere. Close to the surface, the magnetic field is almost homogeneous. Thus, if conductive material is placed within this static magnetic

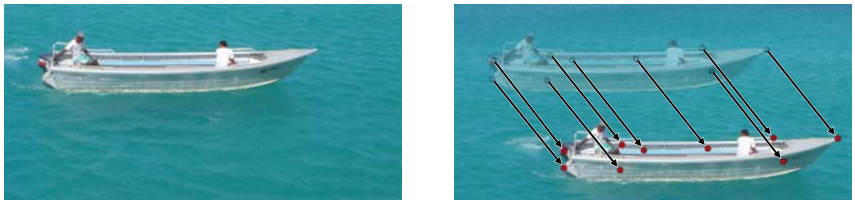
field, $\mathbf{B}$, a Hall-voltage, U_H, can be measured as

$$U_H = A_H \frac{I\,\mathbf{B}}{d}, \tag{2.31}$$

where A_H is the Hall Coefficient and I is electric current. Using three orthogonal Hall-effect sensors, the vector of the earth magnetic field relative to the sensor can be estimated, giving an indication about its orientation. Note that the magnetic field varies with position, and is subject to fluctuation. Also disturbances due to metallic objects or electric devices can affect the measurement.

2.4 Tracking

Object tracking is the process of successive estimation of the location of a target in a video over time (Yilmaz et al., 2006). In contrast to object detection, which is only concerned with the localisation in a single frame, irrespective of a possible information gain from prior frames, object tracking tries to find the transition between the states of each frame (Figure 2.6). This allows building a model of the object trajectory, predict movement, and reduce noise. A state space model encodes the position in the state of the model and incorporates measurements as state updates. This way, false measurements (outliers) and even object occlusions can be handled by the object tracker.



(a) t (b) t+1

Figure 2.6: The target object is tracked from frame t to frame $t + 1$ by matching single features between frames.

Yilmaz et al. (2006) describes three classes of object trackers: 1. point trackers, 2. silhouette trackers, and 3. kernel trackers. Point trackers are concerned with identifying and tracking of individual feature points (e.g. SIFT features, (Lowe, 2004)), while silhouette trackers make use of matching of shape description (e.g. contours, (Yilmaz et al., 2004)). These techniques are of no concern in this thesis. Kernel trackers are appearance based. That is they make use of a description of the appearance of the object to re-identify and track it over frames. In this thesis the Kanade-Lucas-Tomasi (KLT) feature tracker (Shi

and Tomasi, 1994) is employed for tracking objects and parts thereof.

2.4.1 State Space Model

If the position $\boldsymbol{s}$ of an object in the previous time step, $\boldsymbol{s}_{t-1}$, and its velocity, $\dot{\boldsymbol{s}}_t$, is known then the current position, $\boldsymbol{s}_t$, can be estimated using the kinematics equation as

$$\boldsymbol{s}_t = \boldsymbol{s}_{t-1} + \dot{\boldsymbol{s}}_{t-1} + \epsilon, \tag{2.32}$$

where ϵ is the uncertainty of the model.

However, these states are not directly observable, since observations are themselves also uncertain. Hence, state space models progress by utilising the kinematic equation to predict the next state and subsequently incorporate the observation to update this predicted state estimate. Formally, the state vector, $\boldsymbol{x}$, is defined as follows

$$\boldsymbol{x}_t = \begin{pmatrix} \boldsymbol{s}_t \\ \dot{\boldsymbol{s}}_t \end{pmatrix}, \tag{2.33}$$

with the observation vector, $\boldsymbol{y}$, only consisting of measured position, since velocity is usually not directly measurable.

The transition from Equation (2.32) can then be put into linear algebra form by defining the following state space transition matrix, $\mathbf{F}$,

$$\boldsymbol{F} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \tag{2.34}$$

which yields the following state space model description of the system

$$\boldsymbol{x}_t = \mathbf{F}\boldsymbol{x}_{t-1} + \epsilon. \tag{2.35}$$

An equivalent viewpoint is to consider that the states forms a Markov chain with transition probability, $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, and observation probability, $P(\boldsymbol{y}_t|\boldsymbol{x}_t)$. This can be described using a Kalman or Extended Kalman filter (EKF) (Kalman, 1960; Maybeck, 1979), however, as this requires the observations follow a Gaussian distribution, which is not always met, the EKF is insufficient for typical tracking applications. An alternative that does not have this restriction is the Particle filter.

2.4.2 Particle Filters

Particle filters (Doucet et al., 2000) are based on sequential Monte Carlo Simulations and are often used for tracking (Hue et al., 2002). In a particle filter a set of N samples (particles), $x_t^{(i)}$ (where i is the particle index), is used in conjunction with weights (probabilities), $w_t^{(i)}$, to provide a discrete approximation of the state distribution, allowing it to be non-Gaussian distributed.

The most common particle filter variant is the bootstrap filter (Gordon et al., 1993). The algorithm for updating a bootstrap particle filter over time proceeds as follows: Particles are sampled from an initial distribution of

$$\boldsymbol{x}_1^{(i)} \sim P(\boldsymbol{x}_1|\boldsymbol{y}_1) \quad \forall i \in \{1,\ldots,N\}, \tag{2.36}$$

where $\sim$ means "sampled from". Each particle is then weighted according to how well it matches the observation

$$\tilde{w}_1^{(i)} = P(\boldsymbol{y}_1|\boldsymbol{x}_1 = \boldsymbol{x}_1^{(i)}) \tag{2.37a}$$

$$w_1^{(i)} = \frac{\tilde{w}_1^{(i)}}{\sum_{j=1}^{N} \tilde{w}_1^{(j)}}, \tag{2.37b}$$

where $\tilde{w}$ are the unnormalised weights and w are the final weights.

Predictions are made by sampling from the transition probability given the particle's current state,

$$\boldsymbol{x}_t^{(i)} \sim P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1}^{(i)}) \quad \forall i \in \{1,\ldots,N\}, \tag{2.38}$$

Updates occur by updating the particle's weights to reflect their fit to the new observation. Note that particles are not actually moved during the update. The weights are updated as follows:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \cdot P(\boldsymbol{y}_t|\boldsymbol{x}_t = \boldsymbol{x}_t^{(i)}) \tag{2.39a}$$

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}. \tag{2.39b}$$

The prediction and update cycle repeats as new frames arrive. One issue with the particle filter is that of degeneration. This is where all but a few particles will eventually have

zero weight due to only a few particles correctly predicting the next position of the object. These zero-weight particles are in poor areas of the distribution and thus are a waste of processing time to maintain. To solve this, the concept of resampling is used to multiply high-weight particles and remove low-weight particles from the approximation (Doucet et al., 2000).

2.5 Visual Attention

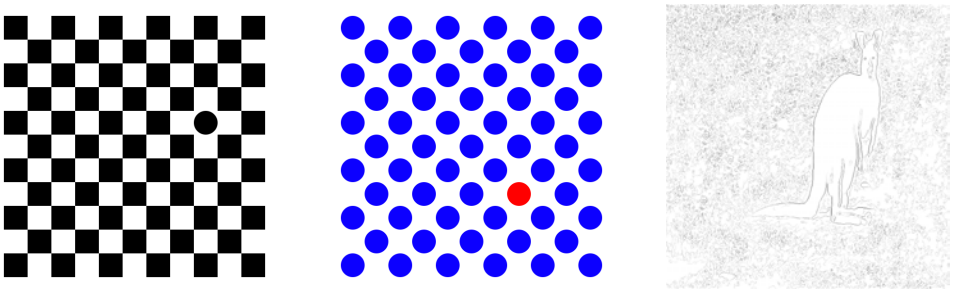
Not all parts of an image are relevant with respect to the overall content. Desimone and Duncan (1995) described two phenomena in regards to attention for human vision: 1. limited processing capacity and 2. selectivity. They illustrated this with an experiment where subjects were given a task and presented with a number of task relevant and irrelevant objects. The authors found that the probability of identifying target objects decreases with the number of task-relevant target objects being present, indicating that limited processing capacity has to be split between the targets. Secondly, an increase in non-target objects did not affect identification performance of target objects, indicating an ability to selectively discard expected irrelevant regions of the retina image. The objective in this thesis is to build an artificial system that imitates this behaviour and is capable of detecting relevant and irrelevant regions within high resolution omnidirectional imagery. Guiding visual attention to relevant regions can not only focus higher-level processing onto relevant areas but also can relieve a human operator from monotonic and tiring scanning of the entire image. This section explains feature based and task driven visual attention and puts them into the context of computer vision. The section also discusses and compares related work carried out in this research area.

2.5.1 Approaches

Humans use their sense of sight as a non-invasive sensor to obtain information about the visual appearance, the colour, and the shapes of their surroundings. While the field of view of the human vision system is about 120–160° horizontally, the central focus, the *fovea*, has a field of view of only 3° (Goldstein, 2007). By moving the eyes the desired scene is put in focus, while the periphery is still monitored by a pre-attentive system for external stimulus (saccade) (Braun, 1994). Typically, the human vision system constantly alternates between fixation and saccade. This "attentive observation of the environment" (Pashler, 1998) can be addressed using two different processes: bottom-up and top-down visual attention.

Bottom-Up Visual Attention

The bottom-up process is stimulus driven: local differences in features like shape, colour, contour, texture, size or orientation are used to identify candidate regions of interest in an image. This is done on a pre-object level, i.e. no knowledge about the appearance of possible objects is necessary, merely the presence (or absence) of low-level features is evaluated and used to guide visual attention to candidate regions. Figure 2.7 shows



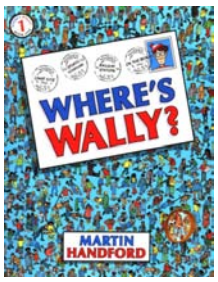
(a) Shape (b) Colour (c) Texture

Figure 2.7: Examples of low-level driven bottom-up visual attention. Stimuli caused by difference in shape, colour, or texture.

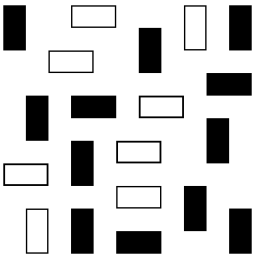
examples of images containing regions that differ in shape, colour, or contour. Without any task knowledge, the attentive region in (a) can be identified as the circle by evaluating the difference of shape in the image. The same is valid for the red circle in (b) when evaluating the difference in colour and the unidentified object in (c) when evaluating the difference in texture.

Top-Down Visual Attention

Visual attention in a top-down process is described on a higher level. Instead of low-level feature differences, specific patterns are defined that describe the potential target object. A prominent example that can describe the constant scanning of the human vision system for these patterns is depicted in Figure 2.8(a): the child book "Where's Wally?" (Handford, 1987) contains images depicting dozens of people in various scenes with a character, "Wally", hidden amongst them. The young reader is presented with the task of finding Wally wearing his red-and-white jumper in each of the images. Contrary to the bottom-up approach, a clear task is given with the description of the appearance of the object. Without this task, it would not be possible to identify the target as evaluation of the differences in low-level features alone would yield ambiguous results.



(a) Where's Wally?



(b) Find all the black horizontal or white vertical bars

Figure 2.8: Examples of task driven top-down visual attention.

Figure 2.8(b) shows a structure with black and white bars that are horizontally and vertically aligned without any recognisable pattern. A feature based bottom-up approach would not be able to detect any areas of visual attention due to insignificant differences in the image. Using a top-down approach that defines black horizontal and white vertical bars as targets, however, allows searching for these patterns and guiding visual attention to identified candidate regions.

2.5.2 Visualisation of Classifier Responses

The response of a Bayesian classifier is probabilistic, i.e. it is normalised to $0 \ldots 1$. For visualisation purposes heatmaps are used in this thesis to visualise the responses of detectors and classifiers. The colour ranging from blue to red indicates the value at each point of the map. A high value translates to a high probability for the depicted class. Figure 2.9 shows a heatmap that is used to depict the spatial probability of a maritime object in an image.

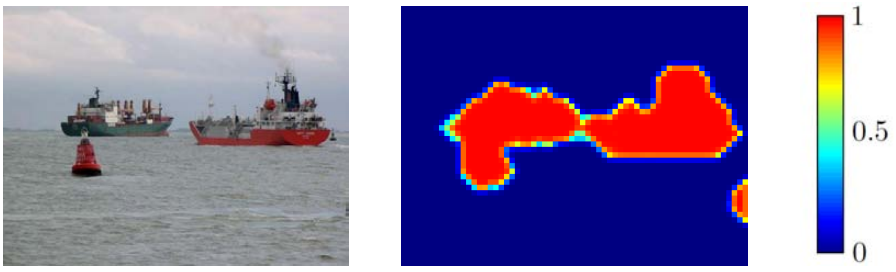


Figure 2.9: Heatmap depicting the spatial probability of a an area containing a maritime object. The heatmap relates the magnitude to a colour ranging from blue to red.

2.5.3 Visual Attention in Computer Vision

The concept of visual attention has been adapted by the computer vision community. Four different, yet overlapping terms have emerged for this kind of processing within the past few decades:

1. **Image Complexity.** Peters and Strickland (1990) define *Image Complexity* as the "inherent difficulty of performing the task associated with it". Their work is placed in the domain of automatic target recognisers, in which the associated task is assumed to be detecting the target within an image. Image complexity refers to the complexity of detecting a target object within the image. The measurement is a mapping indicating the complexity of an image as a monotonic probability.

2. **Object Detection.** *Object Detection* refers to the finding of foreground objects within an image. Object detection is typically class driven (Everingham et al., 2010; Lampert et al., 2008; Chum and Zisserman, 2007), although class independent object detectors have been proposed that are not concerned with the exact type of the object and only compute the probability of an object being present at a specific location within the image (Alexe et al., 2010). Here, a measurement is given to indicate the probability of a region or subwindow of the image containing an object.

3. **Image Saliency.** The *Saliency Map* of an image shows distinctive areas within the image (Itti et al., 1998; Hou and Zhang, 2007; Achanta and Süsstrunk, 2010). Using a Bayesian formulation and the assumption that distinct areas are in fact foreground objects, the saliency map can be interpreted as a probability map that indicates the probability of a region containing a target object.

4. **Visual Attention.** *Visual Attention* is about the detection of unknown, undefined, or unspecified objects (or regions) within the image (Sun and Fisher, 2003; Hu et al., 2008; Frintrop et al., 2010). In contrast to the aforementioned methods, no assumptions are made of the specifics of the object present. Again, a map that indicates the probability of the presence of an object is computed as the result of the visual attention approach.

Both saliency and visual attention are concerned with the problem of finding regions of interest in the scene. However, their exact difference is not well defined. Thus for clarity, this thesis sets the following definitions: saliency is the problem of finding regions that are *unusual* in the image. In contrast, visual attention is the problem of finding regions that are *important* for the problem domain at hand. Thus this thesis defines their

difference as being that visual attention is more task-orientated than saliency; the latter is a purely bottom-up approach, whereas the former requires some top-down information that characterises the problem domain and its goals.

In the early work of Itti et al. (1998), the authors proposed an attention system inspired by the integration of multiple feature maps, as suggested by Koch and Ullman (1985), and the neural architecture of the human vision system. Human vision is most sensitive to contrast changes between a dense centre and a larger surrounding region (Jobson et al., 1997). Itti et al. compute multiple Gaussian pyramidal levels of the input image and decompose each level in colour, intensity, and orientation. They then compute the centre-surround contrast for each feature using the across-scale difference between two levels of the Gaussian pyramid, where the coarser scale functions as the surrounding area. The resulting maps are normalised to a fixed range and eventually combined in a winner-take-all neural network to compute a saliency map, indicating the most prominent saliency regions. Itti et al. demonstrated the strength of the proposed method in 1998; however, their method has been outperformed by recent saliency approaches. Figure 2.10 shows a sample image and the corresponding saliency map as computed by this algorithm. The saliency map is visualised using a heatmap that relates the magnitude to a colour ranging from blue to red (see Section 2.5.2).

(a) Original Image (b) Saliency Map

Figure 2.10: Saliency map computed using centre-surround differences across different scales (Itti et al., 1998).

Harel et al. (2007) proposed a bottom-up graph-based method for saliency detection, designed to predict target fixation of the human vision system in static imagery. The authors perform extraction of the features: colour, intensity, and orientation, as proposed by Itti et al. (1998). However, they do not make use of the full Gaussian pyramid as suggested by Itti et al. In a biologically plausible approach, Harel et al. then compute the dissimilarity between the feature responses as the distance of the logarithmic ratio between points of the image per feature. This results in a Markovian representation of each feature channel, called *activation maps* by the authors. The activation maps are eventually

normalised and combined using a graph-based approach, where the probability between nodes is computed using the distance between the pixels. The node which is most unique with respect to the neighbourhood can then be estimated as the node with the highest weight in the graph. The authors compare their proposed approach to existing methods and show that it yields a better receiver operator characteristic. However, the dissimilarity measure used to compute the activation maps is based on global scene analysis, which faces the problem that images with noisy or complex backgrounds may produce high local contrast and thus yield a higher response in the activation maps. In their experimental investigation the authors found that the detection is biased towards the centre of the image, which actually reflects the human eye's bias, but for the purpose of detecting salient regions in any part of an image, is an unwanted effect.

(a) Original Image (b) Saliency Map

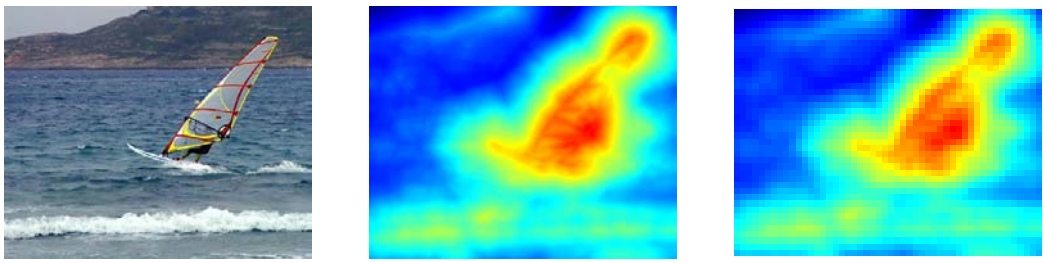
Figure 2.11: Uniqueness of a pixel towards the entire image is computed using a logarithmic ratio of the features. The resulting activation maps are used to compute the saliency map using a graph-based approach (Harel et al., 2007).

Instead of estimating salient regions by finding features that describe foreground, Hou and Zhang (2007) took a different approach and explored the unique properties of the background. They argue that background consists of frequently occurring features and by suppressing them, the foreground can be emphasised. The authors took an information theoretical approach and compute the log spectrum as the log of the amplitude of the Fourier spectrum of a down-sampled version of the input image. Assuming that the image background consists of mostly redundant frequency components, they then compute the spectral residual to extract the frequency components of the foreground. The saliency map is eventually created by mapping the spectral residual back into the spatial domain. Hou and Zhang show that their proposed method outperforms the reference method of Itti et al. (1998) and that it is able to find regions of unique appearance. However, the spectral residual approach fails to detect large objects with respect to the image size since the dominating frequencies of the object will be treated as redundant background. Figure 2.12 shows a sample image and the corresponding saliency map as computed by Hou and Zhang (2007).

(a) Original Image (b) Saliency Map

Figure 2.12: Saliency map computed using the spectral residual of the full-scale and a corresponding downsampled image following Hou and Zhang (2007).

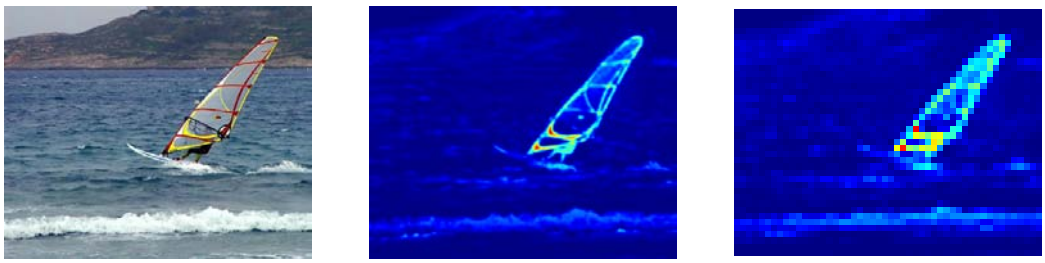
Rosin (2009) based his parameter-free approach for salient region detection purely on edges. The author argues that areas with strong edges are salient and therefore can be detected by computing their density. As a first step Rosin uses the Sobel operator to compute the edge image. From the edge image he then computes the edge magnitude at each position and then performs a threshold decomposition. For each threshold level, he computes the distance transform and eventually combines all distance transform maps into the final saliency map using summation. The author compares his approach to several other approaches, including Itti et al. (1998), Liu et al. (2007), and Ma and Zhang (2003) and shows roughly similar performance. The obvious advantages of his proposed method is that it is simple to compute, purely based on intensity, and is parameter free. However, as the method's only feature cue is edge density, it is entirely dependent on the edge distribution within the image and will fail if a lot of strong edges are present in the background or if the salient object has a low edge contrast. Figure 2.13 shows the saliency map of a sample image as computed by the algorithm of Rosin (2009). In an extension, he proposed the combining of edge detection at multiple scales and the use of opponent colours instead of pure intensity levels and multi-scale difference of Gaussians. However, he did note that the inclusion of colour did not increase performance.



(a) Original Image (b) Saliency Image (c) Saliency Map

Figure 2.13: Saliency image computed using threshold decomposition on the edge density and subsequent distance transform for each threshold level. A saliency map with 8×8 block size computed using block integration is depicted for comparison (Rosin, 2009).

Achanta et al. (2009) introduce a bottom-up approach for salient region detection where they follow the concept of centre surround contrast introduced by Itti et al. (1998) and obtain a saliency map of the input image using features of luminance and colour. As a first step, they estimate the mean CIELAB vector of a Gaussian blurred version of the input image. They then compute the Euclidean distance between the CIELAB vector at each pixel and the mean CIELAB vector of the image. Their approach outputs full resolution maps with well-defined boundaries for the salient objects. However, if the image background is complex or objects are large with respect to the image size, the background gets highlighted as the salient object. This is because the CIELAB mean is meant to represent the average background. This will then be dominated by the object and so will treat it as the background. Achanta and Süsstrunk (2010) address this issue and compute the CIELAB mean over a maximum symmetric surrounding window rather than the entire image, justified by the assumption that the size of the salient object is in relation to its position in the image. The size of the window is symmetric with respect to the pixel and is bound to a maximum by the image border for the most centre pixel. The saliency of each pixel is then computed as the Euclidean distance between the CIELAB vector of the pixel and the CIELAB mean of the maximum symmetric surrounding window. The authors showed that their method outperforms the approaches proposed by Itti et al. (1998), Ma and Zhang (2003), Harel et al. (2007), Zhang et al. (2008), and Achanta et al. (2009) in both precision and recall performance. Figure 2.14 shows a sample image and



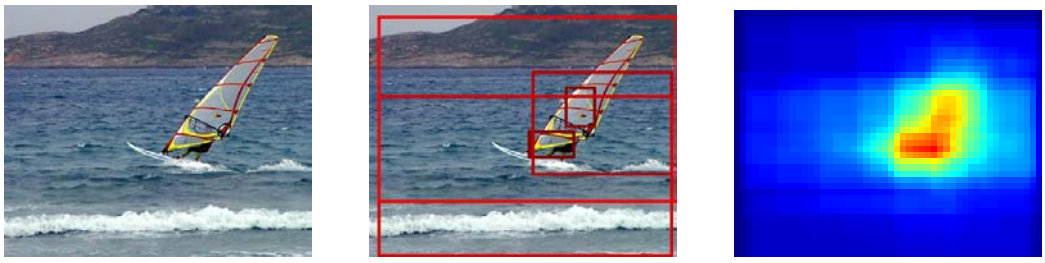
(a) Original image (b) Saliency Image (c) Saliency Map

Figure 2.14: Saliency image computed using the Euclidean distance between a pixel and its mean maximum symmetric surrounding region in CIELAB space. A saliency map with 8×8 block size computed using block integration is depicted for comparison (Achanta and Süsstrunk, 2010).

the corresponding saliency map as computed by Achanta and Süsstrunk (2010). As the algorithm produces full resolution saliency maps, the output is scaled down to map size using bicubic interpolation (Figure 2.14(c)).

The method proposed by Alexe et al. (2010) is not a visual attention detector per se. In fact the authors presented it as a method for detecting generic objects in an image. However, the result of the method is a measurement of an object being present within a

given region. Instead of directly estimating the spatial location of the object within the image, Alexe et al. randomly sample a number of windows (hundreds to thousands) and compute the probability for each window to contain an entire object, called the *objectness* of the window. They suggest that every object (regardless of its class) has either a closed boundary, a unique appearance relative to its surrounding area, or is unique within the entire image. They proposed the use of four different cues that respond to these properties and combine them in a supervised machine learning approach. Following Hou and Zhang (2007), they compute a saliency map for each colour channel in multiple scales. They further compute the colour contrast between each window and its surrounding area in CIELAB colour space. As a third cue, they compute the edge density in border proximity of each window. Last but not least, they make use of the image segmentation technique proposed by Felzenszwalb and Huttenlocher (2004) to compute superpixels of each window. All cues are then combined in a Naïve Bayes classifier. Alexe et al. compare their method to generic object detectors proposed by Dalal and Triggs (2005), Felzenszwalb et al. (2009), and Lampert et al. (2008) as well as the saliency approach of Hou and Zhang (2007) and Itti et al. (1998), showing that their method outperforms all of the aforementioned. Figure 2.15 shows a sample image and the five windows with the highest *objectness* score.



(a) Original Image (b) Top Windows (c) Saliency Map

Figure 2.15: A number of random windows are sampled and an *objectness* score that indicates the probability of a window containing an entire object is computed. (b) shows the five windows with the highest score for the test image – a brighter frame indicates a higher *objectness* value. The saliency map (c) is created by overlaying and normalising 1000 windows (Alexe et al., 2010).

Note that the larger boxes are correctly rated higher according to the *objectness* criterion which requires the window to contain an entire object, which is true for the larger ones but false for the smaller windows only covering parts of the surfer. However, this behaviour is disadvantageous in terms of precision and recall for objects that are not of rectangular shape and thus only cover parts of a bounding box. To compare this algorithm, the scores of 1000 windows have been summed over at their respective window locations and the result has been normalised to indicate the probability of an object as the original algorithm does not produce a probability map. The resulting heatmap is shown in Figure 2.15(c), which gives a reasonable indication of the presence of an interesting object.

2.6 Machine Learning

Machine learning is a vast field of techniques that are used to estimate a relation between input and output based on observations (Michalski et al., 1985; Michie et al., 1994). The field consists of mostly two different approaches: unsupervised and supervised learning. The former deals with the generation of a model that provides a best fit for a given set of unlabelled observations – this approach is of no concern in this thesis. The purpose of supervised learning is to estimate a relation model between known input and output mappings. Training creates a model that then can be used to predict the class of testing data.

In this thesis, a classifier is used as a method to classify features and to fuse features together into a single response. The features are the observable information (input), while the class is the target of classification (output). Although there is a very large body of classifiers (Russell and Norvig, 2010), a Bayesian approach has been chosen as its response is probabilistic and allows for uncertainty to be incorporated in the classifier.

One of the simplest Bayesian classifiers is Naïve Bayes, which assumes feature variables are statistically independent. When this assumption is true, Naïve Bayes has been shown to be a powerful classifier despite its simplicity and speed of training (Russell and Norvig, 2010). Formally, a Naïve Bayes classifier is a generative model where the class *generates* the observable features, i.e. the class is causing the observations. This is initially expressed as a joint probability of random variables, the features, Y_1, Y_2, $\ldots$, Y_n and the class X – where each variable can take a particular set of values. The joint probability can then be factorised as

$$P(X, Y_1, Y_2, \ldots, Y_N) = P(Y_1|X, Y_2, \ldots, Y_N) \cdot P(Y_2|X, Y_3, \ldots, Y_N) \cdot \ldots \cdot P(Y_N|X). \quad (2.40)$$

This can be simplified by utilising the assumption of independence between features

$$P(X, Y_1, Y_2, \ldots, Y_N) = P(Y_1|X) \cdot P(Y_2|X) \cdot \ldots \cdot P(Y_n|X) \quad (2.41a)$$

$$= \prod_{n=1}^{N} P(Y_n|X). \quad (2.41b)$$

With discrete distributions the factors are in fact conditional probability tables that can be easily learned from training data by counting the occurrence of each feature value with each class and subsequently normalising to $0 \ldots 1$.

Classification in Naïve Bayes is the process of inference where one evaluates the following probability using Bayes Rule (Russell and Norvig, 2010) to calculate the probability that a class, c, matches the observed features.

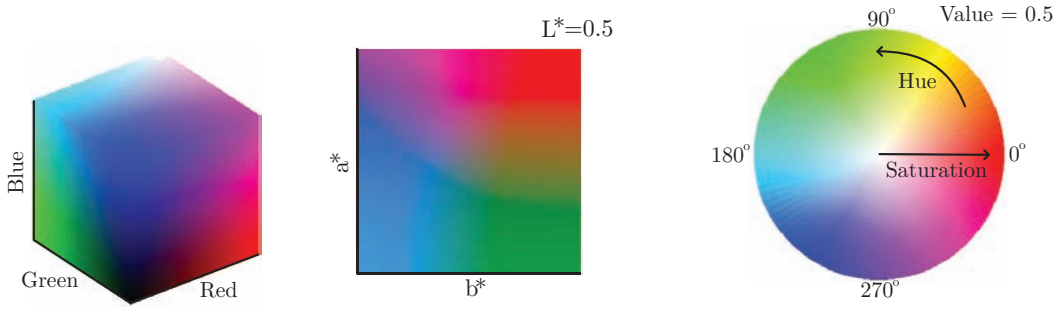
$$P(X = c|Y_1, Y_2, \ldots, Y_N) = \frac{P(Y_1, Y_2, \ldots, Y_N, X = c)}{P(Y_1, Y_2, \ldots, Y_N)} \tag{2.42a}$$

$$= \frac{P(Y_1, Y_2, \ldots, Y_N, X = c)}{\sum_x P(X = x, Y_1, Y_2, \ldots, Y_N)} \tag{2.42b}$$

$$= \frac{\prod_{n=1}^{N} P(Y_n|X = c)}{\sum_x \prod_{n=1}^{N} P(Y_n|X = x)} \tag{2.42c}$$

Classification is a matter of calculating this for all classes and selecting the class with the highest probability.

2.7 Colour Models



(a) RGB (b) L*a*b* (c) HSV

Figure 2.16: RGB, LAB, and HSV colour space. Note that the colour models depicted in this figure are only approximate as this thesis is printed using CMYK, while the online version is rendered in RGB (making L*a*b* and HSV approximates). CMYK is a subtractive colour model that is commonly used by the printing industry, it is not of any interest in this thesis and is not addressed any further. The interested reader is referred to Galer and Horvat (2005).

2.7.1 RGB/sRGB

The widely used RGB colour model (Figure 2.16(a)) is an additive model based on the concept of primary colours (red, green, and blue), which are mixed to yield the desired colour. The model forms a three dimensional cube with all colour components ranging from 0–100%.

The Ladybug camera system that is utilised in this thesis uses a BGGR sensor. For every pixel in the image, it has four different subpixels sensitive to blue, green, green, and red – the two subpixels for green are used to imitate the human high sensitivity to green. Colour processing algorithms such as *k-nearest neighbours*, *linear*, or *bicuibic* interpolation are used to process the independent subpixels and compute the RGB colour of the pixel.

However, a drawback of the RGB colour model is its device dependency; the same colour values in RGB can actually produce different results on different monitors, scanners, and cameras. Therefore, a device independent RGB model, the so-called sRGB model, has been proposed. Typically device manufacturers provide conversion functions to convert from RGB to sRGB colours, $(R_s, G_s, B_s) = f(R, G, B)$, as part of a calibration process. However, neither the RGB or sRGB model are linear and supposedly one dimensional changes (e.g. intensity) require adjustment on all three channels. Other colour models are therefore widely used to overcome this issue.

2.7.2 CIELAB

CIELAB, whose actual name is $L^*a^*b^*$ (1976), as published by the Commission Internationale de l'éclairage (CIE) in 1976, is a three dimensional colour model spanning a manifold that is build on the concept of complementary colours (Figure 2.16(b)). L^* is luminance and a^* the green-red and b^* the blue-yellow components respectively. The aim of the model is that the perceived difference in colours as observed by a human is reflected in a linear difference in CIELAB space. This is achieved by using a logarithmic scale of the red and green components in the spectral distribution.

To convert to CIELAB space, the spectral distribution (X, Y, Z) needs to be computed first. According to the International Commission on Illumination (2004) a colour in sRGB space (R_s, G_s, B_s) can be converted into spectral power values as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{pmatrix} \begin{pmatrix} R_s \\ G_s \\ B_s \end{pmatrix}. \tag{2.43}$$

The spectral power values have been empirically determined to be consistent with the response of a cone cell in the human eye to red, green, and blue colour. Fairchild (2005)

then computes the L^*, a^*, and b^* channels as

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16, \tag{2.44a}$$

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right), \tag{2.44b}$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right), \tag{2.44c}$$

with

$$f(\omega) = \begin{cases} \omega^{\frac{1}{3}} & \omega < 0.008856, \\ 7.787\omega + \frac{16}{116} & \text{otherwise}, \end{cases} \tag{2.44d}$$

where X_n, Y_n and Z_n are the normalised values of the calibrated whitepoint and computed according to Equation 2.43. The linearity of CIELAB allows computing differences of colour using just the Euclidean distance, making the model well suited to be used for efficient computing of colour differences later in Chapter 4.

2.7.3 HSV

HSV, as shown in Figure 2.16(c), is a conical colour model that encodes colour in a single channel (Hue) – in contrast to the aformentioned RGB/sRGB (three channels) and CIELAB (two channels). The relative brightness of the primary colour is encoded in a second channel (Saturation) and the intensity of the image given in a third channel (Value). Gonzalez and Richard (2002) compute Hue, H, Saturation, S, and Value, V, from sRGB

as

$$H = \begin{cases} 0 & R_s = G_s = B_s, \\ \frac{1}{3}\pi * (0 + \frac{G_s - B_s}{\max(R_s, G_s, B_s) - \min(R_s, G_s, B_s)}) & \max(R_s, G_s, B_s) = R_s, \\ \frac{1}{3}\pi * (2 + \frac{G_s - B_s}{\max(R_s, G_s, B_s) - \min(R_s, G_s, B_s)}) & \max(R_s, G_s, B_s) = G_s, \\ \frac{1}{3}\pi * (4 + \frac{G_s - B_s}{\max(R_s, G_s, B_s) - \min(R_s, G_s, B_s)}) & \max(R_s, G_s, B_s) = B_s, \end{cases} \tag{2.45a}$$

$$S = \begin{cases} 0 & R_s = B_s = G_s = 0, \\ \frac{\max(R_s, G_s, B_s) - \min(R_s, G_s, B_s)}{\max(R_s, G_s, B_s)} & \text{otherwise}, \end{cases} \tag{2.45b}$$

$$V = \max(R_s, G_s, B_s). \tag{2.45c}$$

In this thesis, the HSV colour model is used in Chapter 5 to model the colour of sea and sky. The assumption is that sea and sky appear in a blue base colour. Nuances are only variation of this colour, thus only Saturation will change and Hue will remain constant.

2.8 Classification

The proposed visual attention framework is a binary classifier. For every part of a test image, a probability value, p, indicating the presence of a maritime object is computed. If the probability is equal to or above a threshold, $p \geq p_{thresh}$, the image part in question is added to the set of objects P, otherwise it is treated as background. For the purposes of evaluation, there are several standard terms and approaches when analysing the effectiveness of a classifier. This section describes the analysis methods that will be used in this thesis.

Confusion Matrix. The performance of the framework is evaluated by estimating the number of correctly and incorrectly classified instances on images with known ground truth. For evaluation, it is of interest if the classifier identifies objects correctly, but also if it has a tendency to over-segment or miss objects. For this, the number of instances classified as *True Positives*, *False Positives*, *True Negatives*, and *False Negatives* are computed and arranged in a 2×2 matrix, called the confusion matrix. With P as the set of predicted maritime objects and G as the set of actual maritime objects (ground truth), the entries of the confusion matrix are computed as:

- *True Positives (tp)* are the number of *Object* instances correctly classified as class *Object*, computed by the intersection of P and G:

$$tp = P \cap G, \tag{2.46}$$

- *False Positives (fp)* are the number of *Background* instances incorrectly classified as class *Object*, which is expressed as the relative complement of P in G:

$$fp = G \setminus P, \tag{2.47}$$

- *True Negatives (tn)* are the number of *Background* instances correctly classified as class *Background*. The set of True Negatives is the symmetric difference of sets G and P:

$$tn = (P \setminus G) \cup (G \setminus P), \tag{2.48}$$

- *False Negatives (fn)* are the number of *Object* instances incorrectly classified as class *Background*, computed as the relative complement of P in G:

$$fn = P \setminus G. \tag{2.49}$$

Following the notation of Kohavi and Provost (1998), Table 2.1 shows the confusion matrix used to evaluate the proposed framework.

		Predicted	
		Object	Background
Actual	Object	True Positives (*tp*)	False Negatives (*fn*)
	Background	False Positives (*fp*)	True Negatives (*tn*)

Table 2.1: Confusion matrix for the proposed framework. The matrix shows the classification prediction for classes *Object* and *Background*.

Precision and Recall. To depict the performance of a binary classifier, it is common to plot the *True Positive Rate (tpr)* over the *False Positive Rate (fpr)* as the Receiver Operator Characteristics (ROC). For this, the confusion matrix is recomputed for different threshold values, $p_{thresh} \in [0, 1]$, from the probability map. Then, *tpr* and *fpr* are

estimated and plotted as

$$tpr = \frac{tp}{tp + fn} \qquad fpr = \frac{fp}{tn + fp}. \tag{2.50}$$

However, especially in foreground classification tasks datasets are often imbalanced as foreground (class *Object*) typically has fewer instances than background (class *Background*). Davis and Goadrich (2006) showed that ROC plots can be "[...]overly optimistic" in these cases. They propose to plot *Precision (pre)* over *Recall (rec)* to evaluate a classifier for these datasets instead. Precision is a measure for accuracy of detection, that is the proportion of correctly predicted objects over the set of all predicted objects. Recall, on the other hand, is a measure of recognition. It is computed as the ratio of correctly predicted objects over the set of actual predicted objects:

$$pre = \frac{tp}{tp + fp} \qquad rec = \frac{tp}{tp + fn}. \tag{2.51}$$

Precision and recall are influenced mutually. If a classifier is tuned for high detection, more false positives are detected, i.e. the precision will decrease and vice versa. A Precision/Recall plot visualises this in a curve where precision is plotted over recall by computing *pre* and *rec* for different thresholds of the probability map in the same fashion as *tpr* and *fpr*.

F-Score. The *F-Score* (Lewis and Gale, 1994) has been introduced to have a single value for comparison that incorporates both the precision and recall performance of a classifier. It is defined as

$$F_\beta = \frac{(\beta^2 + 1) \cdot pre \cdot rec}{\beta^2 \cdot pre + rec}, \tag{2.52}$$

where $\beta = 1$ is called the F_1-score with equal weights on precision and recall. Emphasis can be given to precision by selecting $\beta \leq 1$, or recall by selecting $\beta \geq 1$. For evaluating classifiers that emphasise recall, a value of $\beta = 2$, which weights recall twice as much as precision, is commonly accepted. Hence the F-Scores used in this thesis are:

- F_1*-Score* that combines precision and recall with equal weights:

$$F_1 = \frac{2 \cdot pre \cdot rec}{pre + rec}, \tag{2.53}$$

- F_2-*Score* that puts more emphasis on the recall of the classifier:

$$F_2 = \frac{5 \cdot pre \cdot rec}{4 \cdot pre + rec}. \tag{2.54}$$

2.9 Summary

The design and implementation of a fully autonomous vision system used for maritime surveillance operations requires the detail understanding of the features, limitations, and capabilities associated with each related subsystem. This facilitates the systemic and integrated design approach that is used to ensure successful future development of the system. This Chapter reviewed the related fields to this research and familiarised the reader with the methods and approaches relevant to this thesis.

Section 2.1 of this chapter introduced the different coordinate systems and derived coordinate transformations that will be used throughout this thesis. In particular, *earth*, *global*, *inertial*, *camera*, *virtual camera*, and *perspective camera* coordinates have been defined.

Next, the different types of omnidirectional cameras were described in Section 2.2. The difference between *panoramic*, *half-spherical*, and *full-spherical* omnidirectional cameras was explained, the advantages of using a multi-camera system for omnidirectional vision in combination with a 3D full-spherical mapping were shown. Following this, the perspective camera model was derived and a model for an omnidirectional camera system using multiple perspective cameras was developed. This camera model forms the base of the research in Chapters 3 and 6.

Section 2.3 dealt with the measurement of ego-motion using inertial sensors that will be utilised in Chapters 3 and 6. Notably, the physical concept of gyroscopes, accelerometers, and magnetometers was derived and the fusion of these sensors using an EKF was discussed.

Object tracking and the state space model of object tracking was the topic of Section 2.4. Different classes of object trackers were explained and the state space model of the tracking process was derived. Then, particle filters as a technique for predicting object movements were introduced. They are used for sensor fusion and image stabilisation in Chapters 3 and 6 of this thesis.

In Section 2.5 visual attention has been introduced and the difference between bottom-up

and top-down attention was described. Visual attention was then put into the context of computer vision and a clear distinction between visual attention and saliency was made. Then, relevant related work was discussed and approaches for evaluation and comparison of the proposed framework in Chapters 4 and 5 were presented.

Section 2.6 formally described a Bayesian Network that is utilised in Chapters 4 and 5 for feature combination.

The chapter concluded with a review of three colour models that will be used in Chapters 4 and 5 as well as a review of evaluation methods for classifiers.

Chapter 3

Virtual Cameras for Omnidirectional Video Stabilisation

The main advantage of using mobile platforms for surveillance is that they allow access to high-risk, hazardous, or remote areas without endangering human operators. Of the camera systems that can be mounted on a maritime platform, fixed narrow field of view cameras are very sensitive as they can easily lose track of targets when the platform is subject to environmental disturbances that cause the platform to move or shake. Stabilisation techniques can be applied to compensate for disturbances and ensure a target remains static in the view. A common approach is to use a pan-tilt-zoom (PTZ) camera for image stabilisation as it would be capable of performing a counter motion to ensure the target stays in view. However, disturbances may be sudden and require immediate and high speed reactions from the computer (or operator) to ensure the target remains in view. Equipping the mobile platform with an omnidirectional camera can overcome this, as it provides a real-time full-spherical view, which will show the target regardless of the platform's orientation. Therefore, the use of an omnidirectional camera removes the need for moveable mechanical platforms and performs stabilisation purely digitally (Battiato et al., 2007; Yang et al., 2009). This is particularly important for maritime platforms given that the environment is inherently unstable due to the rolling motion of waves.

The use of an inertial measurement unit (IMU) can assist the stabilisation process as it measures the unpredictable ego-motion of the platform and therefore is able to reduce the search space of feature matching in the vast omnidirectional video. Such a maritime platform has six degrees of freedom that must be considered for stabilisation. These can be broken down into two components: rotation and translation. While the rotational component only changes the orientation of the camera system with respect to the global frame, translational ego-motion causes perspective changes between the camera and global frame. As established in Section 2.3, an IMU can reliably detect rotational changes and therefore compute the orientation in relation to the world frame. However, its usefulness

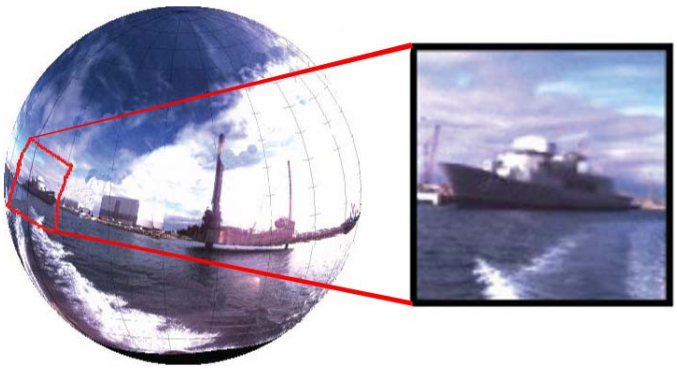


Figure 3.1: Full-spherical representation of omnidirectional image and extracted virtual camera.

for handling translational errors is limited by the need to perform double integration, where errors will very quickly add up and make the estimate unreliable. Nevertheless, rapid disturbances are mostly rotational due to the lower inertial forces required to cause rotation of the platform compared to translation, and these rotational motions can be measured reliably using an IMU. This thesis proposes to utilise this to drastically reduce the search space for a subsequent feature matching algorithm that can then be used to compensate for the translational component of the disturbances.

Typically only a limited field of view of the omnidirectional image is required to fulfil a surveillance task because the target only occupies a small part of the field of view. The stabilisation process can thus be simplified by placing the target region in the centre of the stabilisation process and only stabilising based on this part of the omnidirectional image. The importance of such a *target-centric* stabilisation approach is two-fold. First, the effects of parallax, where objects close to the camera appear to move faster than the distant background, makes global stabilisation a difficult problem to define. Specifically, if parts of the image are moving then there are several stabilisation alignments possible, one for the background and one for each moving target. Moreover, parallax will also affect the background since waves and the platform's own wake will be closer than the coastline, which itself will not be at a uniform distance. Second, translation of the camera platform ensures that the background itself will shift, and this shift cannot be resolved by rotational stabilisation of the omnidirectional view. Thus this thesis takes the approach of target-centric stabilisation, where all targets will be independently stabilised. This is accomplished by extracting a limited field of view around each target from the omnidirectional image, effectively forming multiple virtual perspective cameras (Figure 3.1), one for each target. Stabilisation on each target based on image and IMU sensor fusion thus ensures each target remains static within their respective virtual views even when significant ego-motion (rotation and/or translation) is present. The virtual camera views also

have the benefit of reducing the information load on the computer (and/or operator) as well as limiting the demands on bandwidth and further processing.

The use of an IMU to assist in stabilising an image leads to the need for sensor fusion, where measurements from multiple sensors are utilised together. In a sensor fusion approach, tight coupling between two sensors is a commonly accepted requirement. However, even tightly connected assemblies will shift over time due to vibrations, shaking, or shock on the devices. Furthermore, a tight coupling typically prevents the user from disassembling and reassembling devices, which especially for mission-based setups is highly inconvenient. Calibrated devices also imply that the devices have to be synchronised, i.e. measurements must be taken at precisely the same time, an assumption that many approaches to calibration and sensor fusion require as a precondition. However, synchronisation is in fact a difficult engineering problem. A single manufacturer may ensure that different devices produce similar timings whose differences are low enough to support high precision synchronisation, but devices from different manufacturers are not likely to have such compatibility. This reduces the ability to choose "best-of-breed" devices from specialist manufacturers and forces the consumer's reliance on the precision of the engineering process.

The contribution of this chapter is to alleviate the problems of precise calibration and synchronisation by solving the calibration and synchronisation problems *in combination* with tracking and stabilisation, compensating for errors and/or drift in both calibration and synchronisation by incorporating this as an uncertainty into the stabilisation and tracking process. For this, a vision system consisting of an omnidirectional camera that is connected to an IMU is proposed. The system is capable of efficiently maintaining a stabilised view towards a target by extracting a virtual camera from the omnidirectional image. An omnidirectional camera is utilised to ensure an uninterrupted view onto the target. Stabilisation is achieved by continuously adjusting the orientation of the virtual camera. For this, the inertial sensor provides an estimate of the system's ego-motion. Image registration techniques are then used to refine the estimate and compensate for target motion. For sensor fusion, a probabilistic model is used to allow the use of loosely calibrated and synchronised hardware.

The remainder of this chapter is organised as follows: First, the difference between calibration and synchronisation of a sensor system is explained and issues with calibration and synchronisation are discussed in Section 3.1; the section continues with an introduction of the approach that is utilised to estimate an approximate calibration between the two sensors. Section 3.2 is devoted to the proposed stabilisation approach, the section introduces virtual cameras and derives point conversions between omnidirectional and virtual cam-

eras. The section closes with the description of the stabilisation framework. Section 3.3 gives a brief overview of the utilised system with experiments conducted in Section 3.4. The chapter concludes with a summary given in Section 3.5.

3.1 Calibration and Synchronisation

When integrating camera and IMU, knowing the transformation between the coordinate systems in which the respective sensors perform their measurements is essential. Estimating the spatial transformation (typically offset and rotation but could also include affine transformations) between the two coordinate systems is called *calibration* between the two sensors. The result of the calibration process is a transformation, expressed in homogeneous coordinates as described in Section 2.1.1. Additionally, the temporal offset between measurements is important – measurements at both sensors need to be performed at the same time instant or, if that is not possible, the temporal offset between the measurements needs to be estimated. This process is called the *synchronisation* between the two sensors. Both procedures are described in the following.

3.1.1 Calibration

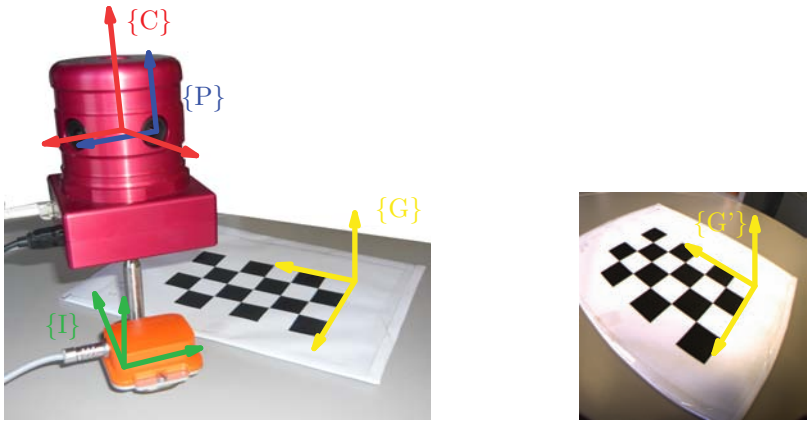
Most calibration approaches utilise a concept of Horn (1987) or Horn et al. (1988), who proposed finding the transformation between two coordinate systems by solving the least-squares problem of a number of measurement-tuples over both systems. Lobo and Dias (2003) observed the direction of gravity and the image horizon in a number of poses or made use of a turntable (Lobo and Dias, 2007) to estimate the relation between a camera and an inertial sensor by applying Horn's method. Recently, Mirzaei and Roumeliotis (2008) and Hol et al. (2010) estimated the transformation between camera and inertial coordinate systems by measuring acceleration and angular velocity while tracking image features on a horizontally aligned pattern. These approaches assume synchronised hardware, i.e. measurements of camera and IMU arrive at the same time instant.

For calibration, first, the intrinsic and extrinsic parameters of the perspective camera must be estimated – see Section 2.2.3 for details. The extrinsic parameters describe the transformation between the 2D image plane of the perspective camera and the global coordinate system. For an omnidirectional camera that has all cameras arranged around a central viewpoint, like the utilised Ladybug camera system, the perspective camera model can be applied to estimate the extrinsic parameters for each camera separately as

described in Section 2.2.4. This process can be executed for any of the six perspective cameras, but will be described for the first camera, $\{P_1\}$, of the Ladybug camera system. Because ${}^{P_n}_{C}\mathbf{T}$ is provided for all $n = 1, \ldots, 6$ cameras, it is sufficient to compute ${}^{P_1}_{G}\mathbf{T}$, as the remaining can be computed as

$$ {}^{P_n}_{G}\mathbf{T} = {}^{P_n}_{C}\mathbf{T} \; {}^{C}_{P_1}\mathbf{T} \; {}^{P_1}_{G}\mathbf{T}. \tag{3.1} $$

In the remainder of this chapter, only the first camera of the Ladybug camera system will be utilised for calibration, so that for the sake of readability the coordinate system of the this camera, $\{P_1\}$, will be denoted as $\{P\}$ from now on.



(a) Calibration pattern (b) Image in perspective camera

Figure 3.2: Coordinate systems utilised for calibration. $\{G'\}$ in (b) denotes the projection of $\{G\}$ into $\{P\}$.

Calibrating the inertial sensor and omnidirectional camera is now a means of estimating the relative transformation between the coordinate systems $\{I\}$ and $\{C\}$, i.e. estimating ${}^{I}_{C}\mathbf{T}$. Because ${}^{P}_{C}\mathbf{T}$ is given and ${}^{P}_{G}\mathbf{T}$ is also known as it is the result of the extrinsic calibration of the perspective camera, the calibration process is fully described by

$$ {}^{I}_{C}\mathbf{T} = {}^{I}_{G}\mathbf{T}_t \; {}^{G}_{P}\mathbf{T}_t \; {}^{P}_{C}\mathbf{T}. \tag{3.2} $$

Note that this formulation implicitly includes a strong time dependency: both, ${}^{I}_{G}\mathbf{T}_t$ and ${}^{G}_{P}\mathbf{T}_t$ assume measurements taken at the very same time step t. However, ${}^{I}_{G}\mathbf{T}_t$ is measured by the IMU while ${}^{G}_{P}\mathbf{T}_t$ is observed by the perspective camera, thus a synchronisation between the two sensors is a precondition for the optimisation to work (synchronisation between two devices will be discussed in Section 3.1.2).

The pixel coordinate (u, v) in the perspective camera is denoted as the point ${}^{P}\boldsymbol{p}=(u, v, 0, 0)^T$

in homogeneous coordinates in $\{P\}$. At the same time, this point can be described using $\{I\}$ as a base system, ${}^{I}\boldsymbol{p}$. For $n = 1, \ldots, N$ measurements of points in $\{P\}$ and $\{I\}$, ${}^{P}\boldsymbol{p}_n$ and ${}^{I}\boldsymbol{p}_n$, the relative pose can be estimated by computing the transformation, $\mathbf{T}$, that maximises (Horn et al., 1988):

$$\sum_{n=1}^{N} \frac{1}{\|{}^{P}\boldsymbol{p_n}\|} \, {}^{P}\boldsymbol{p_n} \Big[\mathbf{T} \, {}^{I}\boldsymbol{p}_n \Big], \tag{3.3}$$

where $\mathbf{T}$ is in fact the desired relative pose transformation between the perspective camera, $\{P\}$, and inertial coordinate system, $\{I\}$. ${}^{I}_{C}\mathbf{T}$ can subsequently found by placing ${}^{I}_{C}\mathbf{T} = \mathbf{T}$ into Equation (3.2).

Time-independent measurements can be taken by measuring a static pose over multiple capture frames. That is, the camera and IMU are held static for a couple of seconds for each measurement step. This way, both devices measure the same conditions over a longer period of time and precise timing is not necessary. In other words, synchronisation can be assumed in this case as the measurement time is lengthened to guarantee an overlap, thereby eliminating the need for precision timing. The approach of Hol et al. (2010) uses the measurements of static poses as an initial guess for initialisation of an Extended Kalman Filter (EKF). The EKF is used to optimise the calibration parameters by tracking a pattern and measuring the inertial motion while the assembly is moved in a random pattern while keeping the pattern in sight of the camera. However, this approach cannot be applied to the system developed in this thesis because it requires precisely synchronised hardware. Unfortunately, without some type of synchronisation (either precise or via lengthened measurements), a calibration solution cannot be found since the optimisation of Hol et al. (2010) will not converge.

3.1.2 Synchronisation

Synchronisation of two sensors is typically treated as an engineering problem and solved in hardware by using a trigger that is connected to both sensors. In the case of a camera and IMU, when the trigger is fired, the image sensor of the camera starts capturing the image. The time needed to capture the image depends mostly on the shutter speed of the camera, which again varies with the environmental conditions. Simultaneously, the inertial sensor starts integrating the acceleration and angular velocity. The duration of this process is defined by the (constant) measurement intervals of the IMU.

Figure 3.3 depicts three possible scenarios of how data could arrive from different sensors

over time. In Figure 3.3(a), both sensors are perfectly synchronised. Measurements arrive at precisely the same time instant – this is the desired behaviour. In (b), a time lag between measurements Y_n and Z_n is observable. However, the lag is a constant delay with measurements Z_n always arriving with a constant latency to their respective measurement Y_n. If this latency is known, it can be eliminated by the fusion algorithm. In (c), a non-synchronised system is depicted. Contrary to (b), the measurements of both Y and Z arrive at varying time intervals with no observable constant latency. Note that in (c) Z_3 represents a lost measurement and is therefore missing in the diagram.

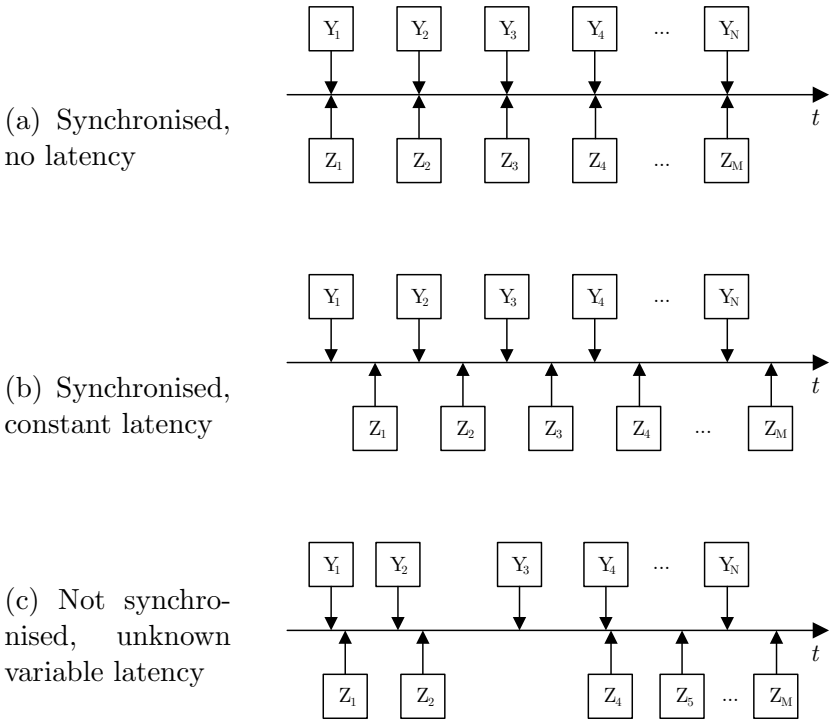


Figure 3.3: Temporal relationship between measurements of two sensors.

The Ladybug camera system, for example, has a latency that averages at 3.4 frames when recording with 25 frames per second. There are a number of reason why the latency is not a constant factor: the Ladybug camera system does not utilise a hardware trigger; instead the camera can only be triggered with a software trigger. Moreover, the Ladybug camera continuously captures images with a fixed frame rate – triggering requests the image that is created with the *next* capture process. Also, the camera performs a JPEG compression of the image before transmitting the data. The utilised JPEG compression, however, does not have a constant run time. Furthermore, this latency does not include disk access or computation time on the computer, which in the case of a high data rate can be significant – see Section 3.3 for a detailed overview of the hardware utilised in this thesis. The IMU on

the other hand only produces a small amount of data, typically less than 10kb per second, which will be processed much faster. The discrepancy between the two devices becomes more significant the more variation is in the latency during measurements, because this timing cannot be measured from outside of the devices. It can in fact be delayed by $0 \dots 40ms$ ($\frac{1}{25fps}$). With disturbances that cause e.g. angular velocities of $100^{\circ -1}$, this could add up to an error increasing by up to 3.3° per second.

Therefore the hardware utilised in this thesis exhibits the temporal relationship as depicted in Figure 3.3(c), and the devices cannot practicably be synchronised precisely. Hence, no algorithms (including calibration algorithms) that assume synchronised hardware can be applied to the hardware utilised in this thesis, since calibration optimisations such as Hol et al. (2010) will not converge to a solution.

Thus, instead of trying to precisely estimate the latency and synchronisation between the sensors, this thesis proposes to handle errors in synchronisation (and by extension, calibration) by modelling them with uncertainty and incorporating this as part of the stabilisation process itself. This model will lead to a more flexible and convenient approach as it allows loose coupling of sensors with a rough estimate of synchronisation and calibration.

3.2 Stabilisation

The stabilisation approach proposed in this thesis does not require precise calibration or synchronisation. Instead an approximate calibration in static poses as described in Section 3.1.1 is performed to provide a (constant) rough estimate of the transformation between $\{C\}$ and $\{I\}$, ${}^{C}_{I}\mathbf{T}$. A probabilistic model is then utilised to cope with varying time offsets and measurement uncertainties.

Instead of following a traditional image stabilisation approach and stabilising the entire omnidirectional view, only a region of the omnidirectional image with a limited field of view will be stabilised. Rotational movements of the camera are measured by an IMU, which provides an initial estimate of the ego-motion of the camera. Image registration is then used to refine these estimates. The calculated ego-motion is then used to adjust an extract of the omnidirectional video, forming a virtual camera that is focused on the target being tracked.

3.2.1 Virtual Cameras

Sun et al. (2005) used a virtual camera to detect and track a person in a wide angle panoramic video. Designed for indoor lecture halls, the camera system itself is kept static. Mauthner et al. (2006) proposes a method for region matching in omnidirectional images. They extract virtual perspective camera images for each detected region to avoid distortions introduced by the omnidirectional image. Virtual cameras have been used to extract regions of interest in a high-resolution football video which is convenient for watching on small devices (Seo et al., 2007). A system that detects and tracks speakers in an office conference call scenario was proposed by Fiala et al. (2004). They extracted perspective views from full omnidirectional video that were then sent to the remote participant instead of the full omnidirectional video. The extracts were automatically adjusted based on video target tracking and target detection using beam forming on a microphone array. Onoe et al. (1998) used a head tracker as a user input to estimate the desired viewing direction of an operator in an omnidirectional video. They then extracted a perspective view from the panoramic image and presented it to the user. Their proposed system also had an automatic *follow-me*-mode that continuously adjusted the orientation of the perspective view based on a background subtraction technique.

In this thesis the concept of a virtual camera as an extract of a higher resolution image is utilised for the purpose of stabilisation. For this, the high resolution image has to be created first allowing a continuous representation of the captured images in a single coordinate system. Given that the intrinsic and extrinsic parameters of the six perspective cameras of the Ladybug camera system are known, it is possible to map every pixel of every camera onto a unit sphere free of distortion. This is a very natural representation of the omnidirectional image as it represents the omnidirectional image as it was captured, with the camera in the centre.

Applying Equation (2.18), a pixel coordinate (u, v) in a perspective camera, $\{P\}$, can thus be mapped onto the unit sphere that is spanned by $\{C\}$ as

$$ {}^{C}\boldsymbol{p} = {}^{C}_{P}\mathbf{T} \left[\begin{pmatrix} \mathbf{K} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ -v \\ 0 \\ 0 \end{pmatrix} + \tfrac{1}{2} \begin{pmatrix} -w \\ h \\ 0 \\ 0 \end{pmatrix} \right], \tag{3.4} $$

where $\mathbf{K}$ is the camera matrix and $w \times h$ is the image size of the perspective camera. This mapping is performed for every pixel of every camera, yielding an omnidirectional image mapped onto the unit sphere as depicted in Figure 3.1.

Creating a virtual camera simply means reversing the mapping and extracting a perspective image from the unit sphere that contains the desired image. Instead of extracting a distorted camera image by literally applying the inverse of Equation (3.4), a rectified image should be extracted. This ensures that the image of the virtual camera contains an image where straight lines are straight and right angles are orthonormal (orthographic projection). The camera matrix, $\mathbf{K}$, contains the parameters to compensate for the distortions of a perspective camera. By omitting this parameter in the inverse projection, the extracted image remains rectified. This is advantageous for subsequently applying computer vision algorithms as they typically require rectified images as input.

The actual parameters of the virtual camera are the desired field of view and the resolution of the camera. As established, the virtual camera simulates a perspective camera model and is thus limited to a maximum field of view of $\leq 180°$, however, this is not a real limitation because the reason for using a virtual camera is that a limited field of view is actually desired. Therefore one would rather extract two or three virtual cameras with smaller fields of view instead of one camera that covers the entire view. As the coordinate system of the sphere is continuous, no theoretical limit for the resolution exists. However, the data (image) is built from a discrete (limited) number of pixels as implied by the capturing camera(s) and therefore a practical limit for the resolution exists. For example, when the perspective cameras of the omnidirectional camera system capture a field of view of 72° with 1024 pixels each, it would make no sense to extract a virtual camera with a field of view of 30° and a resolution of 2048 pixel. Nevertheless, extracting such a virtual camera would be possible by interpolating the sampling from the sphere.

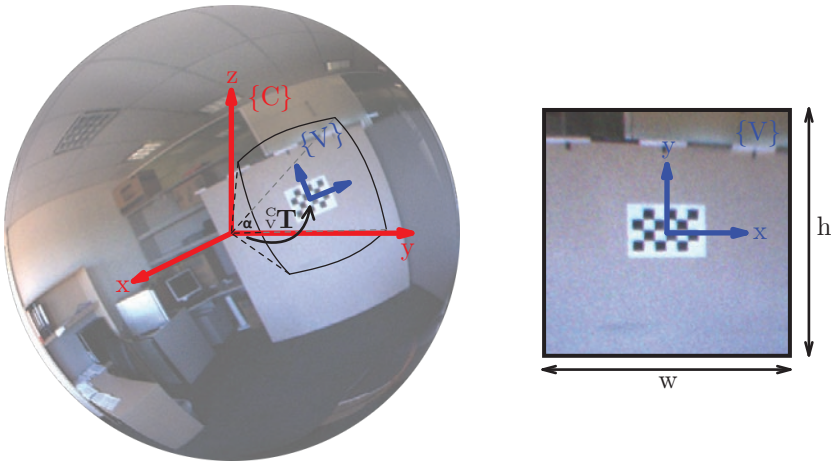


Figure 3.4: Parameters of a virtual camera.

A perspective camera has three physical properties: the field of view and resolution was mentioned before, but it also has an orientation – the direction the camera "faces". These three parameters are also used to describe a virtual camera (Figure 3.4). The virtual

camera therefore has:

- An orientation. This is the transformation of the virtual camera coordinate system, $\{V\}$, with respect to the camera coordinate system, $\{C\}$, which spans the unit sphere that contains the omnidirectional image. Note that the orientation of the virtual camera with respect to the camera coordinate system can be changed over time (e.g. due to tracking a moving target), hence the transformation needs to be denoted as time-dependent, ${}^{V}_{C}\mathbf{T}_t$.
- A field of view, α, which is the vertical angle of the area extracted from the omnivision sphere (the horizontal angle can be calculated based on α and the aspect ratio implied by the resolution of the virtual camera below).
- A resolution of $h \times w$. The resolution describes the sampling interval of the virtual camera from the sphere.

Since the virtual camera will later be used to perform target-centric stabilisation and tracking, it is necessary to define mappings between the omnidirectional and virtual camera, and define how the virtual camera can "slide" over the "surface" of the omnidirectional view as it tracks a target.

3.2.1.1 Camera to Virtual Camera Coordinates

Let ${}^{C}\boldsymbol{p}$ be a point in $\{C\}$. Then the projection of ${}^{C}\boldsymbol{p}$ in virtual camera coordinates, $\{V\}$, depends on the orientation, ${}^{C}_{V}\mathbf{T}_t$, as well as the resolution, $h \times w$, and field of view, α, of the virtual camera.

Transforming the coordinate towards the virtual camera yields

$$(x,\, y,\, z,\, 0)^T = {}^{V}_{C}\mathbf{T}_t \; {}^{C}\boldsymbol{p}, \tag{3.5}$$

which can then be projected as pixel $(u,\, v)$ into the virtual camera using orthographic projection

$$u = \frac{1}{2}\Big(w - h\frac{x}{z}\tan\big(\frac{\alpha}{2}\big)^{-1}\Big) \tag{3.6a}$$

$$v = \frac{h}{2}\Big(1 - \frac{y}{z}\tan\big(\frac{\alpha}{2}\big)^{-1}\Big) \tag{3.6b}$$

yielding ${}^{V}\boldsymbol{p} = (u,\, v,\, 0,\, 0)^T$.

3.2.1.2 Virtual Camera to Camera Coordinates

A point in a virtual camera, ${}^{V}\boldsymbol{p} = (u,\, v,\, 0,\, 0)^T$, can subsequently be transformed into $\{C\}$ by applying the inverse transformation. Projecting the pixel coordinates onto the unit sphere at identity yields the point $(x,\, y,\, z)$ in 3D coordinates:

$$x = -(w - 2u) \cdot (w^2 + h^2 \cdot cs^2 + 4x(-w + x) + 4y(-h + y))^{-\frac{1}{2}}, \tag{3.7a}$$

$$y = -(h - 2v) \cdot (w^2 + h^2 \cdot cs^2 + 4x(-w + x) + 4y(-h + y))^{-\frac{1}{2}}, \tag{3.7b}$$

where $cs = \sin\left(\frac{\alpha}{2}\right)^{-1}$. The third coordinate is subsequently computed by normalising the coordinate onto the unit sphere,

$$z = -(1 - x^2 - y^2)^{\frac{1}{2}}. \tag{3.7c}$$

Then, the computed point, $(x,\, y,\, z)$, is rotated towards the virtual camera as

$${}^{C}\boldsymbol{p} = {}^{C}_{V}\mathbf{T}_t\ (x,\, y,\, z,\, 0)^T, \tag{3.8}$$

yielding the point in camera coordinates.

3.2.2 Initialisation of a Virtual Camera

As a first step, an appropriate virtual camera is created with the desired target object in the centre of the view. Let ${}^{C}\boldsymbol{p}_{t=0}$ be the position of the object at time step $t = 0$, then the initial orientation of the virtual camera can be estimated by computing Rodrigues' formula for the transformation of the unit vector in camera coordinates, ${}^{C}\boldsymbol{p}_0$, towards the object's position as

$$\mathbf{R}_{\tilde{\boldsymbol{\Omega}}}(\beta) = \mathbf{I}_3 + \sin\beta \cdot \tilde{\boldsymbol{\Omega}} + (1 - \cos\beta) \cdot \tilde{\boldsymbol{\Omega}}^2, \tag{3.9}$$

where the skew symmetric matrix, $\tilde{\boldsymbol{\Omega}}$, is defined as

$$\tilde{\boldsymbol{\Omega}} = \begin{pmatrix} 0 & -\tilde{\boldsymbol{\omega}}_3 & \tilde{\boldsymbol{\omega}}_2 \\ \tilde{\boldsymbol{\omega}}_3 & 0 & -\tilde{\boldsymbol{\omega}}_1 \\ -\tilde{\boldsymbol{\omega}}_2 & \tilde{\boldsymbol{\omega}}_1 & 0 \end{pmatrix}, \quad \text{with } \tilde{\boldsymbol{\omega}} = {}^{C}\boldsymbol{p}_0 \times {}^{C}\boldsymbol{p}_{t=0}, \tag{3.10}$$

and β is the angle between ${}^{C}\boldsymbol{p}_0$ and ${}^{C}\boldsymbol{p}$ that can be computed as the $\cos^{-1}$ of the dot-product.

In homogeneous notation, the initial orientation of the virtual camera can subsequently be denoted as

$$ {}^{V}_{C}\mathbf{T}_{t=0} = \begin{pmatrix} \mathbf{R}_{\tilde{\Omega}}(\beta) & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.11} $$

The remaining parameters for the virtual camera are the resolution and field of view. The field of view is partly given by the size and distance of the target and the desired oversegmentation. The resolution should be chosen with the performance of the desired post-processing image processing algorithms in mind. Figure 3.1 shows an example of an initialised virtual camera.

3.2.3 Feature Registration for Stabilisation

After creating the virtual camera, $\{V\}$, the stabilisation process is initialised by registering features within the object's region. As mentioned in Section 2.4, the evaluation of image registration and tracking processes is of no concern in this thesis – instead, a well established existing method is utilised for these purposes. For feature registration the method of Shi and Tomasi (1994) is employed. Feature registration is performed in the virtual camera, yielding a set of features, ${}^{V}\boldsymbol{f}_{n,t=0}$, where $n = 1, \ldots, N$ is the number of features and $t = 0$ denotes the initial time step.

3.2.4 Problem Statement

The approach taken here for image stabilisation is to detect the target movement by tracking the target and then instead of applying an inverse transform to the camera image, change the orientation (parameters) of the virtual camera to adjust to the change of scene caused by the camera's ego-motion. In other words, stabilisation is addressed as a problem of adjusting the orientation of the virtual camera, ${}^{C}_{V}\mathbf{T}_t$, such that the observed features remain ideally static within the virtual camera view over time, i.e.

$$ {}^{V}\boldsymbol{f}_{n,t+1} = {}^{V}\boldsymbol{f}_{n,t}, \tag{3.12} $$

holds for the subsequent time step, $t+1$. A feature, ${}^{V}\boldsymbol{f}_{n,t}$, can be expressed in $\{C\}$ as

$$ {}^{C}\boldsymbol{f}_{n,t} = {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}, \tag{3.13} $$

therefore, Equation (3.12) can be rewritten with respect to $\{C\}$ as

$$ {}^{C}_{V}\mathbf{T}_{t+1}\ {}^{V}\boldsymbol{f}_{n,t+1} = {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}, \tag{3.14} $$

where ${}^{C}_{V}\mathbf{T}_{t}$ is the old and ${}^{C}_{V}\mathbf{T}_{t+1}$ the new orientation of the virtual camera. Note that $\{C\}$ is not stabilised and changes with respect to $\{G\}$ if the camera is moved. Therefore, for Equation (3.14) to be true, a correction term, $\Delta\mathbf{T}$, has to be introduced to transform ${}^{C}_{V}\mathbf{T}_{t}$ into ${}^{C}_{V}\mathbf{T}_{t+1}$ in order to satisfy Equation (3.12) if camera ego-motion is present:

$$ {}^{C}_{V}\mathbf{T}_{t+1} = \Delta\mathbf{T}\ {}^{C}_{V}\mathbf{T}_{t}, \tag{3.15} $$

and subsequently

$$ {}^{C}_{V}\mathbf{T}_{t+1}\ {}^{V}\boldsymbol{f}_{n,t+1} = \Delta\mathbf{T}_{t+1}\ {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}. \tag{3.16} $$

For the purpose of stabilisation, the problem comes down to estimating and applying $\Delta\mathbf{T}$, which provides the orientation update of the virtual camera such that the features in $t+1$ remain at the same position as in t. Note that with no ego-motion present, $\Delta\mathbf{T}$ will be the identity matrix and the virtual camera will remain at the previous orientation.

3.2.5 Feature Correspondence Under Camera Motion

The features, ${}^{V}\boldsymbol{f}_{n,t}$, estimated in the virtual camera can be denoted in global coordinates, $\{G\}$ (Figure 3.5),

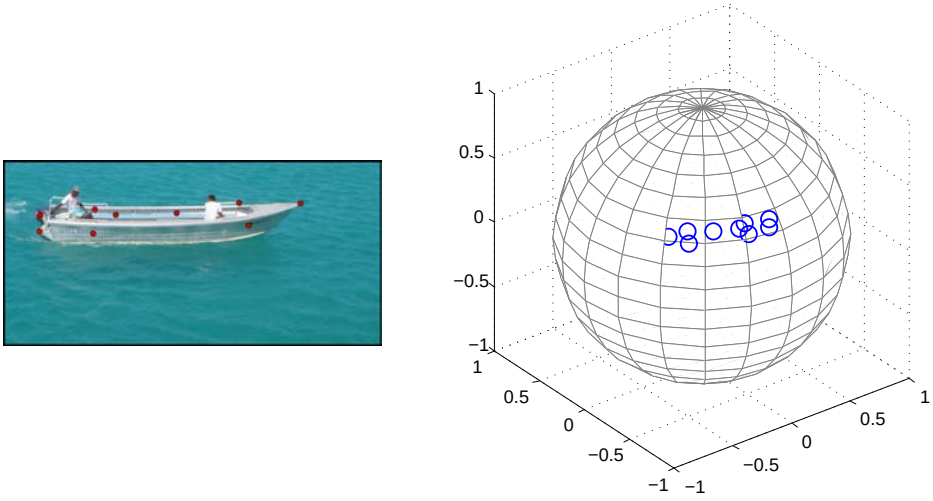
$$ {}^{G}\boldsymbol{f}_{n,t} = {}^{G}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}. \tag{3.17} $$

Note that ${}^{G}_{V}\mathbf{T}_{t}$ is time-varying because

$$ {}^{G}_{V}\mathbf{T}_{t} = {}^{G}_{I}\mathbf{T}_{t}\ {}^{I}_{C}\mathbf{T}\ {}^{C}_{V}\mathbf{T}_{t}. \tag{3.18} $$

Therefore, for a stationary target the position of the features must remain constant if expressed in $\{G\}$, regardless of the camera's ego-motion:

$$ {}^{G}\boldsymbol{f}_{n,t+1} = {}^{G}\boldsymbol{f}_{n,t}. \tag{3.19} $$



(a) Virtual camera (b) Sphere

Figure 3.5: Features estimated in virtual camera and then mapped onto unit sphere.

Combining Equations (3.14)–(3.19) then yields the transition between time steps t and $t+1$ for all features in the virtual camera as:

$$ {}^{C}_{V}\mathbf{T}_{t+1}\ {}^{V}\boldsymbol{f}_{n,t+1} = \underbrace{{}^{C}_{I}\mathbf{T}\ {}^{I}_{G}\mathbf{T}_{t+1}\ {}^{G}_{I}\mathbf{T}_{t}\ {}^{I}_{C}\mathbf{T}}_{\Delta\mathbf{T}}\ {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}. \tag{3.20} $$

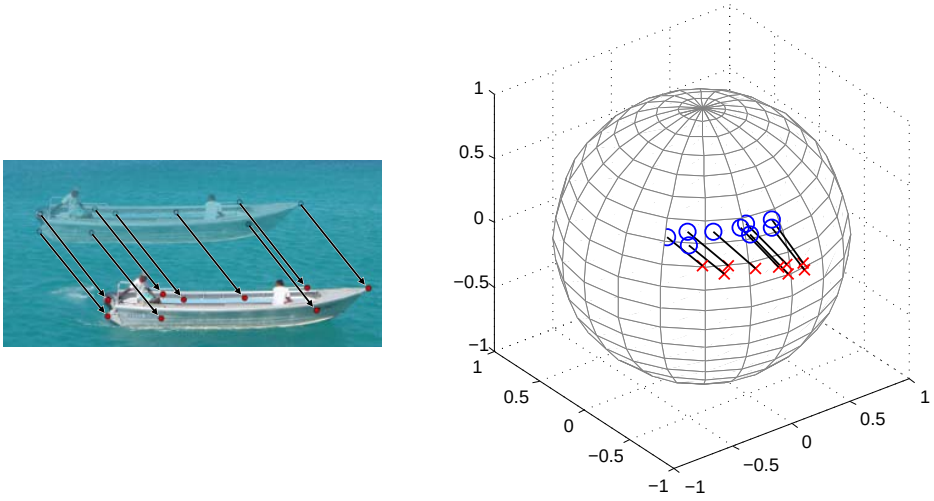
From this, it can be seen that $\Delta\mathbf{T}$ is comprised of different types of transformations. ${}^{G}_{I}\mathbf{T}_{t}$ and ${}^{I}_{G}\mathbf{T}_{t+1}$ are measured by the IMU and denote the orientation of $\{I\}$ with respect to $\{G\}$ at the respective time steps. The matrix product, $({}^{I}_{G}\mathbf{T}_{t+1}{}^{G}_{I}\mathbf{T}_{t})$, subsequently expresses the change of orientation of $\{I\}_{t+1}$ with respect to $\{I\}_{t}$ between time step t and $t+1$. The transformation between the camera and inertial sensor, ${}^{C}_{I}\mathbf{T}$, and its inverse, ${}^{I}_{C}\mathbf{T}$, are denoted as constant. However, it was established earlier in Section 3.1 that depending on the hardware configuration these transformations comprise calibration and synchronisation parameters that cannot be estimated precisely and could also vary over time. This means that correction term $\Delta\mathbf{T}$ is in fact time dependent. Hence, it has to be split into two correction terms, one per time step, $\Delta\mathbf{T}_{t}$ and $\Delta\mathbf{T}_{t+1}$. The correction term from Equation (3.20) then becomes

$$ \Delta\mathbf{T} = \Delta\mathbf{T}_{t+1}\ {}^{C}_{I}\mathbf{T}\ {}^{I}_{G}\mathbf{T}_{t+1}\ {}^{G}_{I}\mathbf{T}_{t}\ {}^{I}_{C}\mathbf{T}\ \Delta\mathbf{T}_{t}, \tag{3.21} $$

yielding the the full transition,

$$ {}^{C}_{V}\mathbf{T}_{t+1}\ {}^{V}\boldsymbol{f}_{n,t+1} = \Delta\mathbf{T}_{t+1}\ {}^{C}_{I}\mathbf{T}\ {}^{I}_{G}\mathbf{T}_{t+1}\ {}^{G}_{I}\mathbf{T}_{t}\ {}^{I}_{C}\mathbf{T}\ \Delta\mathbf{T}_{t}\ {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}. \tag{3.22} $$

In other words, the proposed approach is to match features between two consecutive frames (tracking), then utilise the available knowledge of how the features moved and how the assembly rotated in order to estimate the correction needed to provide a stabilised view of that target. Assuming the orientation of the virtual camera, $\{V\}$, with respect to $\{C\}$ remains the same between the time steps t and $t+1$, then, given camera ego-motion, the features in the virtual camera have to shift. Figure 3.6 shows the movement of features in a static virtual camera over time.



(a) Virtual camera (b) Sphere

Figure 3.6: Image registration and feature matching with subsequent mapping onto unit sphere.

If the orientation of the virtual camera is kept static over time, i.e. ${}^{V}_{C}\mathbf{T}_{t+1} = {}^{V}_{C}\mathbf{T}_{t}$, this requires introducing feature movement of $\Delta^{V}\boldsymbol{f}_{n,t+1}$ between time steps t and $t+1$ and subsequently allows rewriting Equation (3.19) as

$$ {}^{C}_{V}\mathbf{T}_{t}\left[{}^{V}\boldsymbol{f}_{n,t} + \Delta^{V}\boldsymbol{f}_{n,t+1}\right] = \Delta\mathbf{T}_{t+1}\ {}^{C}_{I}\mathbf{T}\ {}^{I}_{G}\mathbf{T}_{t+1}\ {}^{G}_{I}\mathbf{T}_{t}\ {}^{I}_{C}\mathbf{T}\ \Delta\mathbf{T}_{t}\ {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}. \tag{3.23} $$

Note that the correcting factor, $\Delta\mathbf{T}_{t+1}$, represents the synchronisation and calibration uncertainty. It will be broken down later in Equation (3.27).

Because

$$ {}^{V}\boldsymbol{f}_{n,t+1} = {}^{V}\boldsymbol{f}_{n,t} + \Delta^{V}\boldsymbol{f}_{n,t+1}, \tag{3.24} $$

Equation (3.23) can be rewritten as

$$ {}^{V}\boldsymbol{f}_{n,t+1} = {}^{V}\boldsymbol{f}_{n,t} + \underbrace{{}^{V}_{C}\mathbf{T}_{t}\ \Delta\mathbf{T}_{t+1}\ {}^{C}_{I}\mathbf{T}\ {}^{I}_{G}\mathbf{T}_{t+1}\ {}^{G}_{I}\mathbf{T}_{t}\ {}^{I}_{C}\mathbf{T}\ \Delta\mathbf{T}_{t}\ {}^{C}_{V}\mathbf{T}_{t}\ {}^{V}\boldsymbol{f}_{n,t}}_{\Delta^{V}\boldsymbol{f}_{n,t+1}}, \tag{3.25} $$

which, in fact is the standard kinematics equation.

This suggests that everything comes down to a tracking problem, i.e. the parameters of the virtual camera for time step $t+1$ can be computed by tracking the features from frame t to $t+1$.

3.2.6 Stabilised Feature-Based Object Tracking

So far, features have been discussed with respect to virtual camera coordinates, $\{V\}$. However, it is actually necessary to work in 3D global coordinates, $\{G\}$, when performing stabilised tracking. This is because tracking must be performed in a single frame of reference and there are in fact two frames of reference active during tracking based stabilisation: the inertial coordinate system, $\{I\}$, and the camera coordinate system, $\{C\}$, for the IMU and camera feature registration. Thus it is convenient to transform these into a common frame of reference, which is in fact the global coordinate system, $\{G\}$. Furthermore, this has several advantages. Firstly, global coordinates are already semi-stabilised due to the IMU cancelling out rotational disturbances. Secondly, it provides a continuous space, and the tracker subsequently does not face any boundary issues and no hand over-problems have to be addressed. If desired, the result (or even intermediate computations) can always be transferred back into virtual camera coordinates as described in Sections 3.2.1.1 and 3.2.1.2.

A particle filter is utilised on each available feature of the target object to keep dimensionality computationally tractable. If all N features of a given object are tracked by a single particle filter that would imply an N dimensional state space and it is well known that particle filters scale very poorly with dimensionality (Doucet et al., 2000), which would quickly lead to tracking failures. Instead each feature ${}^{G}\boldsymbol{f}_{n,t}$ is tracked independently and later the target object's movement is estimated by a least squares fitting of the motion of the individual features.

The position and velocity of each particle on the unit sphere is described by a state vector:

$$\boldsymbol{x}_t = \left(x,\, y,\, z,\, \frac{dx}{dt},\, \frac{dy}{dt},\, \frac{dz}{dt} \right)^T. \tag{3.26}$$

The state vector is expressed in global coordinates and corresponds to ${}^G\boldsymbol{f}_{n,t} = (x,\, y,\, z,\, 0)^T$ above. Observations, $\boldsymbol{y}_t$, are a mapping of ${}^V\boldsymbol{f}_{n,t}$ into global coordinate space following Equation (3.17) – such that they are in the same domain as $\boldsymbol{x}_t$.

The final goal of the stabilisation process is to estimate $\Delta\mathbf{T}_t$ per time step t. $\Delta\mathbf{T}_t$ is actually overlaid with uncertainty due to synchronisation and calibration issues as mentioned earlier in Section 3.1. It therefore cannot be modelled directly. Its effects on the position of the features of the virtual camera, on the other hand, *can* be modelled.

In a particle filter, there are two types of uncertainty: model and measurement uncertainty. Here, the measurement uncertainty is the error introduced by inaccurate measurements of the new position of the feature. In contrast, the model uncertainty is the transform uncertainty due to inaccurate calibration and synchronisation. In effect, model uncertainty in a particle filter defines how widely to search for a feature, and measurement uncertainty defines how strictly the particles must conform to the observed measurement.

Therefore the final position error of a feature between two time steps, t and $t+1$, is a combination of the feature error (measurement uncertainty), which is small and the transform error (model uncertainty), which is significant during a disturbance. The latter actually consists of two pieces: the calibration and the synchronisation errors as described in Section 3.1. As established, both of these errors cause a transformation error between the time steps, hence, they can therefore be modelled as

$$\Delta\mathbf{T}_{t+1} = \Delta\mathbf{T}_{t+1}^{Meas}\, \Delta\mathbf{T}_{t+1}^{Calib}\, \Delta\mathbf{T}_{t+1}^{Sync}. \tag{3.27}$$

A standard particle filter is utilised as defined in Chapter 2. The prediction step, $P(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)$, which transitions a particle, uses a model uncertainty defined as a zero-mean linear Gaussian with variance $\mathbf{C}$ on the state $\boldsymbol{x}_t = \left(x,\, y,\, z,\, \frac{dx}{dt},\, \frac{dy}{dt},\, \frac{dz}{dt}\right)^T$ and $\mathbf{C}$ is a diagonal covariance matrix since $(x,\, y,\, z)$ are assumed to be independent. The standard deviations of the state vector are chosen as $\boldsymbol{\sigma} = (0.01,\, 0.01,\, 0.01,\, 0.005,\, 0.005,\, 0.005)$ – the values are quite small since tracking is occurring on a unit sphere.

The update step of the particle filter uses an exponential measurement with the uncertainty based on the distance between a predicted particle, $\boldsymbol{x}_t^{(i)}$, and the observed new position

$\boldsymbol{y}_t$ of the feature. Specifically,

$$P(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}) = \lambda^{-1} \exp\left(\lambda \cdot \left\|\boldsymbol{y}_t - \boldsymbol{x}_t^{(i)}\right\|_2\right), \tag{3.28}$$

where $||_2$ is the l^2-norm, and $\lambda = 1$.

Tracking is then a matter of initialising particles around a given feature using the model uncertainty, then predicting the next position and updating the particle weights, w, according to Equation 2.39a in Chapter 2. The estimated position of the feature is then the weighted sum of the particles:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \cdot P(\boldsymbol{y}_t|\boldsymbol{x}_t = \boldsymbol{x}_t^{(i)}) \tag{3.29a}$$

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}. \tag{3.29b}$$

This predict-update cycle repeats for each frame.

The set of feature position estimates then is used to approximate the new position of the entire target so as to locate the virtual camera view around the target, as described in the next section.

3.2.7 Finding the Optimal Orientation of the Virtual Camera

Tracking provides information for time steps t and $t+1$ for every $n = 1, \ldots, N$ features; ${}^G\boldsymbol{f}_t$ and ${}^G\boldsymbol{f}_{t+1}$ are both available from the particle filter framework. The goal now is to compute a new orientation of the virtual camera for the next time step, ${}^C_V\mathbf{T}_{t+1}$, that minimises the reprojection error.

Thus, in $\{G\}$, it comes down to computing a $\Delta\mathbf{T}_{t+1}$ that minimises the cost function, $E(\cdot)$, of the reprojection of all features in subsequent time steps:

$$E(\Delta\mathbf{T}) = \min_{\Delta\mathbf{T}} \sum_{n=1}^{N} \left(\left[\Delta\mathbf{T}_{t+1} {}^G\boldsymbol{f}_{n,t}\right] - {}^G\boldsymbol{f}_{n,t+1}\right)^2. \tag{3.30}$$

It now becomes clear why tracking in $\{G\}$ instead of $\{C\}$ is advantageous for the proposed system: $\{G\}$ already incorporates the measurements from the IMU and therefore $\Delta\mathbf{T}_{t+1}$ is only a correction term to the already semi-stable image.

In fact, $\Delta\mathbf{T}_{t+1}$ is used to refine the *orientation* of the virtual camera. The optimisation

problem can therefore be simplified to finding the optimum rotation, $\Delta\mathbf{R}_{t+1}$, following

$$\Delta\mathbf{T}_{t+1} = \begin{pmatrix} \Delta\mathbf{R}_{t+1} & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.31}$$

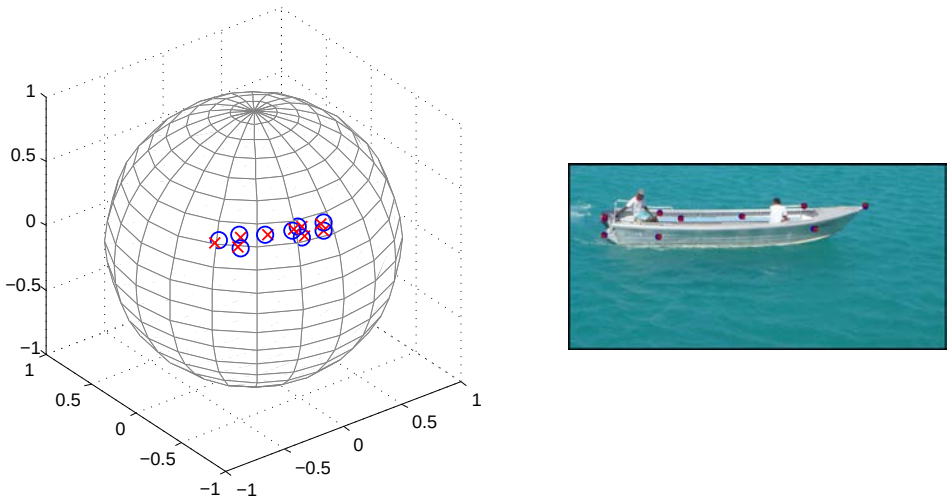
Because $\Delta\mathbf{R}_{t+1}$ is only a correction term and therefore small, it can be decomposed as the skew-symmetric matrix:

$$\Delta\mathbf{R}_{t+1} = \exp \begin{pmatrix} 0 & -\tilde{\omega}_3 & \tilde{\omega}_2 \\ \tilde{\omega}_3 & 0 & -\tilde{\omega}_1 \\ -\tilde{\omega}_2 & \tilde{\omega}_1 & 0 \end{pmatrix}. \tag{3.32}$$

Therefore, the optimisation problem in Equation 3.30 can be solved by finding the skew vector, $\tilde{\boldsymbol{\omega}} = (\tilde{\omega}_1, \tilde{\omega}_2, \tilde{\omega}_3)$ that minimises the reprojection error.

Typically more than three feature matches are found, allowing it to compensate for feature shifts introduced by translational camera ego-motion and making it robust to feature mismatching.

Figure 3.7 shows the reprojection as computed by the Newton optimisation process. For the parameter optimisation, the initial guess of the parameter can be selected as $\tilde{\boldsymbol{\omega}} = (0, 0, 0)$, which would yield $\Delta\mathbf{R}_{t+1} = \mathbf{I}_3$. In subsequent steps, the previously estimated parameters can be used.



(a) Sphere (b) Virtual camera

Figure 3.7: Feature reprojection on unit sphere and subsequent mapping into virtual camera image.

Finally, the orientation of the virtual camera can be updated using

$$ {}^{G}_{V}\mathbf{T}_{t+1} = \begin{pmatrix} \Delta\mathbf{R}_{t+1} & 0 \\ 0 & 1 \end{pmatrix} {}^{G}_{V}\mathbf{T}_{t}. \tag{3.33} $$

Reversing Equation (3.31) yields

$$ {}^{G}_{V}\mathbf{T}_{t+1} = \Delta\mathbf{T}_{t+1} \; {}^{G}_{V}\mathbf{T}_{t}, \tag{3.34} $$

which can be broken up as

$$ {}^{G}_{I}\mathbf{T}_{t+1} \; {}^{I}_{C}\mathbf{T} \; {}^{C}_{V}\mathbf{T}_{t+1} = \Delta\mathbf{T}_{t+1} \; {}^{G}_{I}\mathbf{T}_{t} \; {}^{I}_{C}\mathbf{T} \; {}^{C}_{V}\mathbf{T}_{t}. \tag{3.35} $$

Solving for ${}^{C}_{V}\mathbf{T}_{t+1}$ then yields the new orientation of the stabilised virtual camera as:

$$ {}^{C}_{V}\mathbf{T}_{t+1} = {}^{C}_{I}\mathbf{T} \; {}^{I}_{G}\mathbf{T}_{t+1} \; \Delta\mathbf{T}_{t+1} \; {}^{G}_{I}\mathbf{T}_{t} \; {}^{I}_{C}\mathbf{T} \; {}^{C}_{V}\mathbf{T}_{t}. \tag{3.36} $$

Note that this equation only has one correcting term while the initial stabilisation formulation in Equation (3.22) had two. There, the correction terms were stated with respect to $\{C\}$, which made it necessary to split them into t and $t+1$. However, in Equation (3.36) the correction term is given in $\{G\}$. Formally, this means that its primary function is to stabilise the image for movement of the features, i.e. to compensate for translational ego-motion of the camera. However, the calibration and synchronisation uncertainties are still part of this correction term (as discussed for Equation (3.27)).

3.3 System Hardware

This thesis utilises the Ladybug 2, an omnidirectional camera system manufactured by Point Grey Research. The camera system consists of six individual perspective cameras that are aligned around a single viewpoint. Each camera captures video at 1024×768 pixels per frame. The perspective cameras are synchronised using hardware triggers such that the conditions laid out in Section 3.1.2 are met and the capturing behaviour of the camera system actually conforms to Figure 3.3(a)). Furthermore, the perspective cameras are calibrated with respect to a joint coordinate system that has the shared viewpoint as origin – in fact $\{C\}$. The calibration transformations between the perspective coordinate systems $\{P_n\}$ and $\{C\}$ are provided by the manufacturer.

For inertial measurement the MTi, manufactured by Xsens is utilised. This IMU utilises an Extended Kalman Filter to fuse three-axis accelerometer, gyroscope, and magnetome-

Figure 3.8: Assembly of Ladybug 2 omnidirectional camera (red with black lenses) and MTi inertial sensor (orange).

ter measurements to compute drift-free attitude and heading information. The inertial coordinate system, $\{I\}$, is spanned at the centre of the MTi. The output of the sensor is given as a rotation with respect to the global coordinate system, $\{G\}$, spanned at the current location of the inertial sensor. While the sensor provides accelerometer measurements, which theoretically can be used to estimate translational movement, this is not facilitated in this thesis due to the reasons laid out in Section 2.3.2.

While the MTi is equipped with a hardware trigger mechanism, the Ladybug 2 camera does not possess such capabilities. In fact, as was laid out in Section 3.1, earlier in this Chapter, only a software trigger is available on the Ladybug 2 camera. The overall synchronisation between IMU and camera therefore follows the relationship as depicted in Figure 3.3(c). An approximate calibration between the devices was performed, as described in Section 3.1.1, yielding the approximate transformation between $\{C\}$ and $\{I\}$ as ${}^{C}_{I}\mathbf{T}$.

The assembly is depicted in Figure 3.8.

3.4 Experiments

Experiments were conducted to demonstrate the superiority of the proposed stabilised tracking approach by fusing the IMU and camera together in comparison to simply using either the IMU or camera registration alone. All experiments utilised the assembly described in Section 3.2 with relatively inaccurate calibration and synchronisation hence more sophisticated algorithms depending on tight calibration or synchronisation could not be considered. Experiments were carried out in lab conditions and a checkerboard was

chosen as a target object because it allows for precise error measurement. The assembly was calibrated several months before the actual measurements were recorded and it was taken apart and put back together a number of times in roughly but not precisely the same alignment. No re-calibration was performed after each assembly. Thus ${}^{C}_{I}\mathbf{T}$ could actually be considered a best guess rather than a true calibration because of the potential error. Nine sequences of 24 seconds each were recorded with full resolution omnidirectional video data at 25fps and inertial data sampled at 50Hz, with different videos testing different aspects of the stabilisation problem:

(I) three sequences with only rotational motion,

(II) three sequences with only translational motion, and

(III) three sequences with combined rotational and translational motion.

The sequences were recorded with the assembly held and moved about by hand to emulate real-world conditions. This, on the other hand, means that some noise is present in all sequences and particularly sequences (I) and (II) do not contain purely rotational and translational motion as (I) also contains some rotational and (II) some translational motion.

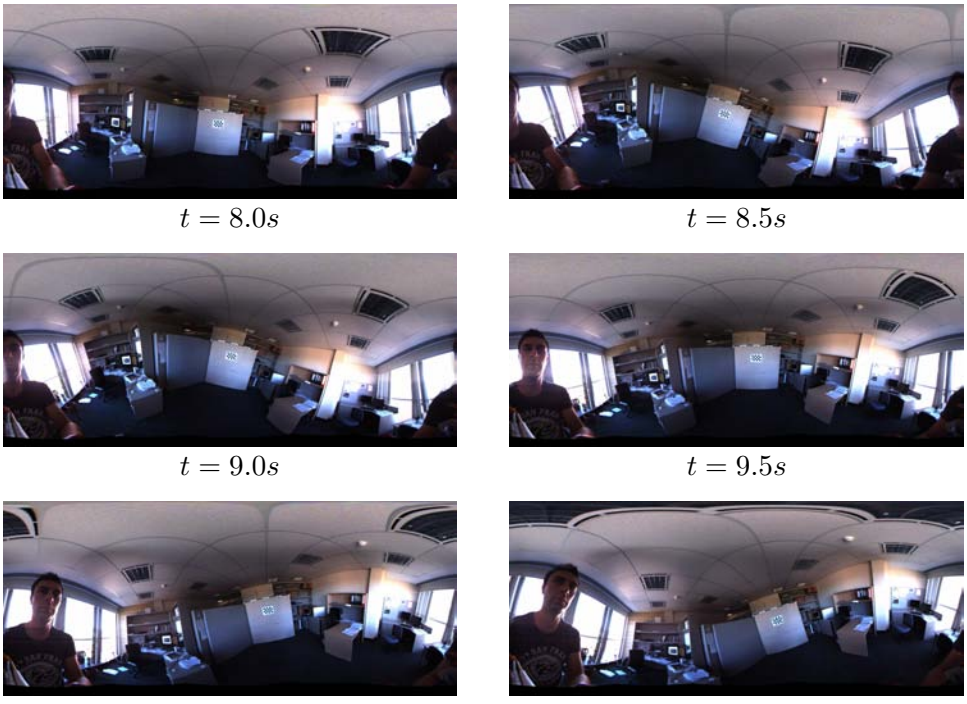


Figure 3.9: Omnidirectional video with rotational motion (Sequence (I)).

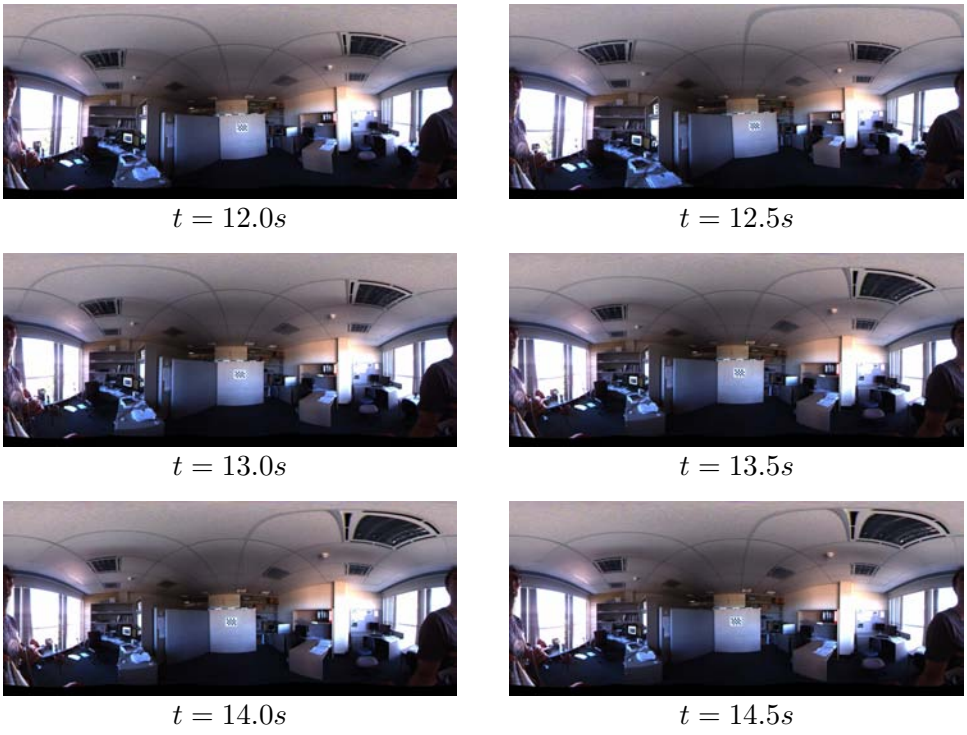


Figure 3.10: Omnidirectional video with translational motion (Sequence (II)).

One recording of each sequence was selected representatively (called (I), (II), and (III) from now on) and will be discussed in the following. However, the results from all sequences were used for the quantitative results that will be shown.

Figures 3.9–3.11 show $2.5s$ extracts from each of the sequences. Depicted are frames in $0.5s$ time steps from the full raw omnidirectional video. In Figure 3.12, ground truth for the three sequences is depicted over the full length of the recordings. The Figure shows the orientation of the assembly with respect to the roll, pitch, and yaw axis (x, y, z) and the position, which is measured in metres relative to the position from the beginning of the recording. The singularity in the roll and yaw channel at 17s in (I) of Figure 3.12 is due a full turn of the camera about the respective axis. This can also be seen in (III), where the roll axis is flipped upside down at 14s and 15s and later the yaw axis at 16s and 18s into the recording. In recordings where rotation is present ((I) and (III)) some noise in the position curve can be observed. This coincides with excessive changes in orientation. The reason for this is that the assembly is fairly bulky and a hand-held sequence might require e.g. re-grip or shift of hands if turned over.

To evaluate the performance of the proposed technique, the stabilisation error is computed as the Euclidean distance between the centre of the virtual camera and the centre of mass

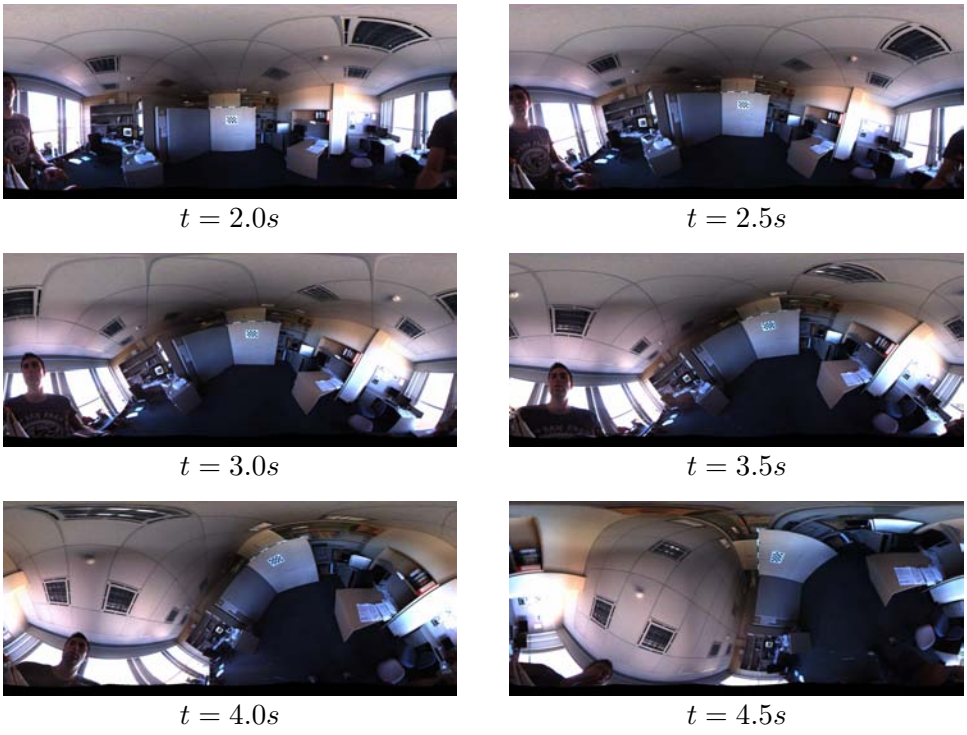


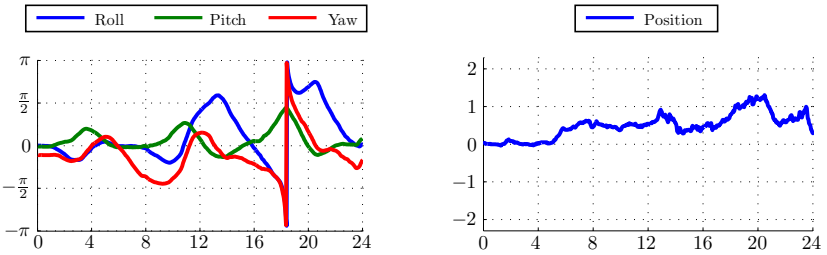
Figure 3.11: Omnidirectional video with combined motion (Sequence (III)).

of the checkerboard in the virtual camera. This method is plausible as the centre of the object and the centre of the virtual camera should coincide in a perfectly stabilised virtual camera.

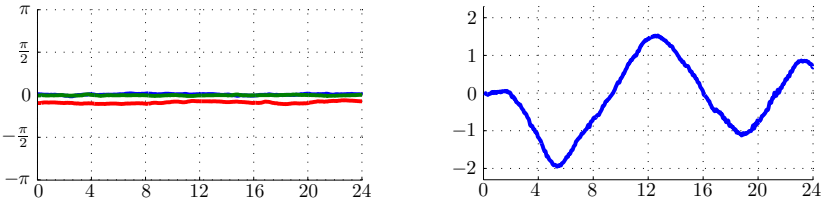
Note that this measurement does not take the rotational offset into account. This restriction is acceptable as the proposed algorithm is not prune to rotational error. At worst, the non IMU stabilised vision only approach would get rated better than it really is. The stabilisation error is subsequently mapped onto the unit sphere and converted into an angular representation. This allows the error to be stated independent from the resolution of the virtual camera. The error, ϵ, is computed as

$$\epsilon = \frac{\alpha}{h}\left[\left\|{}^{V}_{G}\mathbf{T}_t \frac{1}{N}\sum_{n=1}^{N} {}^{G}\boldsymbol{f}_{n,t}\right\|^2\right], \tag{3.37}$$

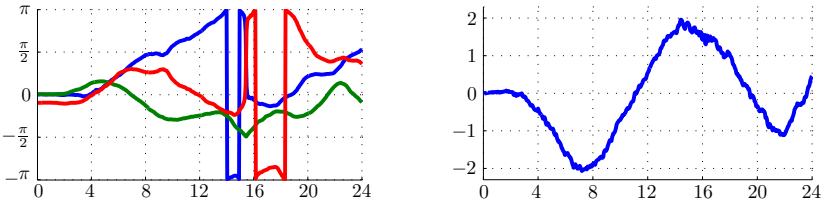
where α is the field of view of the virtual camera, h the vertical resolution, and ${}^{V}_{G}\mathbf{T}_t$ is the orientation of the virtual camera with respect to $\{G\}$ at the current time step, t.



(a) Rotational motion only (Sequence (I)).



(b) Translational motion only (Sequence (II)).



(c) Combined motion (Sequence (III)).

Figure 3.12: Ground truth for Sequences (I)–(III). The absolute orientation of the assembly in Euler Angles is shown on the left. The Figures on the right depict the translational offset with respect to the starting position in metres.

3.4.1 Results

The results of the experimental investigation are listed in Table 3.1. Here, the mean stabilisation error for all stabilisation techniques is computed for each set of sequences. Note that if the target object was unrecoverably lost, the error is computed up until that point – this happened for the vision only approach in several recordings that contain significant rotational ego-motion (all of sequences (I) and (III)).

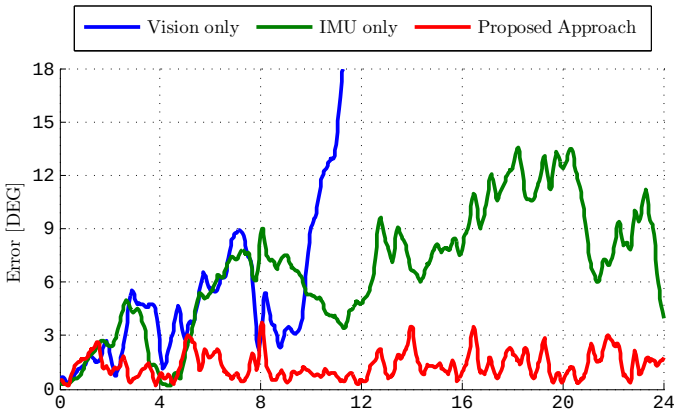
	(I) Rotation	(II) Translation	(III) Combined
IMU	6.65 ± 3.43	7.33 ± 4.64	8.71 ± 5.90
Vision	$5.04 \pm 3.78^{\dagger}$	1.32 ± 0.69	$2.58 \pm 2.41^{\dagger}$
Proposed Approach	1.31 ± 0.80	1.20 ± 0.63	1.31 ± 0.74

Table 3.1: Mean shift error in degrees. † Feature tracker of vision only approach unrecoverably lost the target, the mean error is computed up until the loss of the target.

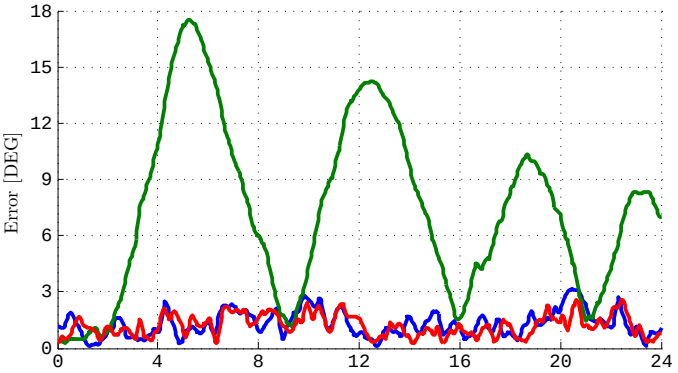
Figures 3.14–3.16 show virtual camera extracts as computed by the proposed and compared stabilisation algorithms for sequences (I)–(III). The frames correspond to the raw omnidirectional video shown earlier in Figures 3.9–3.11.

The results in Table 3.1 indicate that the IMU is a reliable source for stabilising ego-rotation. However, quick and rapid movements are not handled well by the IMU. The plot in Figure 3.13(a) shows that the error increases over time. This is directly correlated with the actual movement of the assembly (ground truth in Figure 3.12(a) shows an increase in rotational velocity over time) and can be explained by the lack of precise calibration and synchronisation. It is expected that the error will be higher with faster ego-motion because the Ladybug camera system has an average latency of $3.4fps$ (see Section 3.3), which adds up to an offset of up to $0.13s$, heavily affecting the performance in rapid ego-motion scenarios. Moreover, the IMU is not of much use for pure translational disturbances. The error plot in Figure 3.13(b) shows that the error is directly correlated with the translational ego-motion of the assembly as depicted in Figure 3.12(b). A similar error behaviour can be observed for the combined motion in Figure 3.13(c). As expected, the IMU is capable of detecting the rotational component of the ego-motion (to a certain extent) but completely unusable for the estimation of the translational component. The main error as shown in (c) is caused by the translational component of the ego-motion with some overlaid noise that is caused by the missing calibration and synchronisation between the devices.

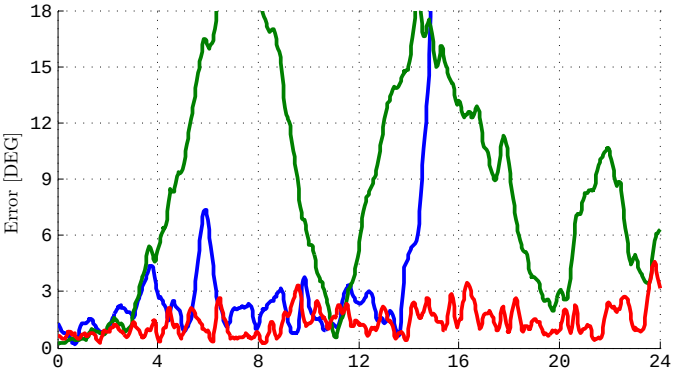
Relevant extracts of the IMU stabilised virtual camera for Sequences (I)–(III) are depicted



(a) Rotational motion only (Session (I))



(b) Translational motion (Sequence (II))



(c) Combined motion (Sequence (III))

Figure 3.13: Orientation error between the camera centre and the object's centroid.

in Figures 3.14(a)–3.16(a). 3.14(a) shows the stabilisation of rotational ego-motion with the checkerboard in approximately the centre of the virtual camera. The error is caused by missing calibration and synchronisation between camera and IMU. The translational movement in 3.15(a) remains undetected by the IMU, the necessary horizontal compensation is therefore missing, causing the checkerboard to move from the right to the left in the virtual camera. One can see that the virtual camera shown in 3.14(a) has the correct alignment but no compensation for the translational component of the ego-motion.

According to the mean error in Table 3.1, the vision approach outperforms the IMU approach with all three types of ego-motion. However, this is only partially true; while the feature tracker of the vision only approach can initially keep up with the IMU and track the rotational component of the ego-motion, it eventually loses track, and once the track is lost it is a permanent failure. Since the error is only computed up until the total loss of the target, the actual performance of the vision only approach is worse than the error implies. This behaviour can be observed for all sequences that contain rotational ego-motion – see error plots in Figure 3.13(a) and 3.13(c). However, the feature tracker of the vision only approach performs almost flawlessly when only translational movement is present. The plot in 3.13(b) shows the error is under 3° at all times. This is expected with movement where translation is dominant since the apparent motion of the scene is not large.

The frames in Figure 3.14(b) show the time instants just before the feature tracker loses track of the target. It can be seen how the error builds up over time and the checker board drifts off. 3.15(b) shows an almost perfect stabilisation with only minor errors. However, in Figure 3.16(b), with ego-rotation present again, the tracker still has the target in sight after 4.5 seconds but it shows a rotational discrepancy building up which will ultimately see it fail.

The proposed approach outperforms the other approaches in all types of ego-motion in all test sequences. The proposed stabilisation technique has a maximum mean error of 1.31°. More importantly, the mean error over the different types of disturbances is consistent. This shows that the feature tracker provides a high enough confidence to compensate for the lack of synchronisation and calibration of the assembly. Compared to the vision only approach that utilises the same feature tracker as the proposed approach, the proposed approach does not drift because the uncertainty of the particle filter can be held very small because the IMU already provides a semi-stabilised platform. This allows the search space to be to be narrowed significantly and subsequently improves tracking and ultimately stabilisation performance because of fewer misdetections.

Similarly, the rotation-only sequence depicted in Figure 3.14(c) shows a very stable virtual camera focused on the checker board. Only minimal deviations are observable. The IMU measures the ego-rotation and the feature tracker assists with refinements compensating for the lack of calibration and synchronisation. Therefore, a better result than the IMU only approach is expected. The sequence in Figure 3.15(c) shows a similar performance as for the vision only approach with the checkerboard stabilised in the centre of the virtual camera. In 3.16(c), the proposed approach is the only method providing a satisfactory stabilisation of the scene. As expected, the IMU estimates the rotational component of the ego-motion and the feature tracker refines the estimate and compensates for translational movement as well as calibration and synchronisation offsets.

3.5 Summary

This chapter proposed a stabilisation technique for omnidirectional cameras with an application to maritime surveillance. In the maritime domain, image stabilisation is an important aspect due to the challenging environmental conditions. Image stabilisation of an omnidirectional camera is especially demanding because of the instantaneous full spherical view. Issues with high resolution or parallax effects are more prominent in these camera systems because of the high field of view. Due to potentially rapid and excessive disturbances, a sensor fusion approach that utilises an inertial measurement unit in combination with an image registration was proposed. The IMU is able to reliably detect rotational disturbances while the image feature tracker provides a robust estimation of translational ego-motion.

Section 3.1 of this chapter discussed the need for calibration and synchronisation between the camera and inertial sensor. In particular, the missing synchronisation capabilities of the utilised Ladybug 2 camera and its implications to calibration were analysed and explained. Existing techniques were applied to provide an approximate calibration between the two sensors.

In Section 3.2, the unit sphere was selected as a frame of reference for the omnidirectional image. Virtual cameras that provide a limited field of view of the full spherical view were then introduced and the mappings between sphere and virtual cameras were derived. It was established that stabilisation of a virtual camera can actually be described as a tracking problem. A particle filter assisted image tracker was then utilised for feature tracking. Subsequently, optimisation techniques were utilised to fuse the estimates of the IMU and feature tracking that minimises the reprojection error between time steps. The

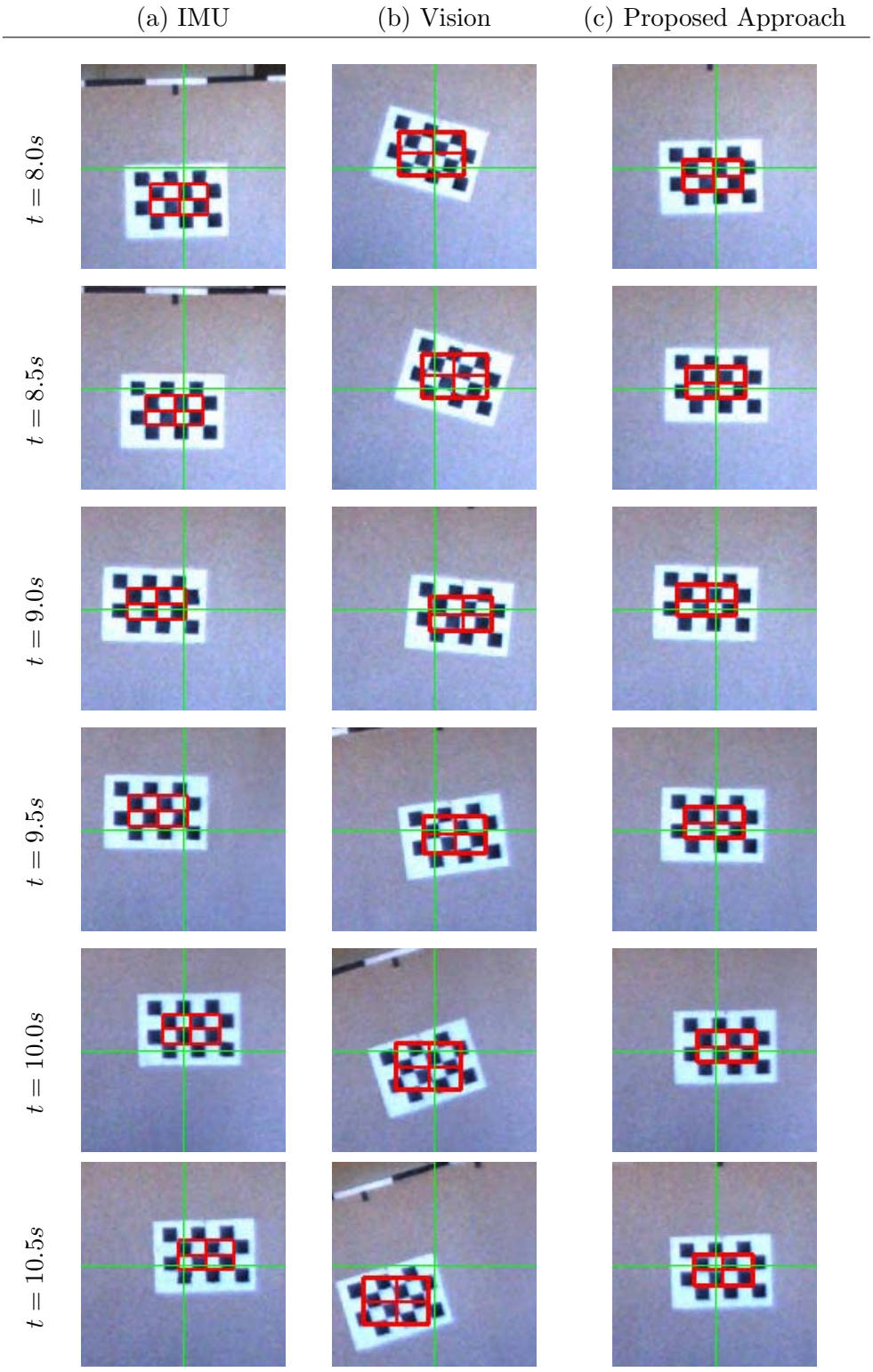


Figure 3.14: Omnidirectional video with rotational motion (Sequence (I)). The red cross shows the centre of the centroid, the green cross indicates the centre of the virtual camera.

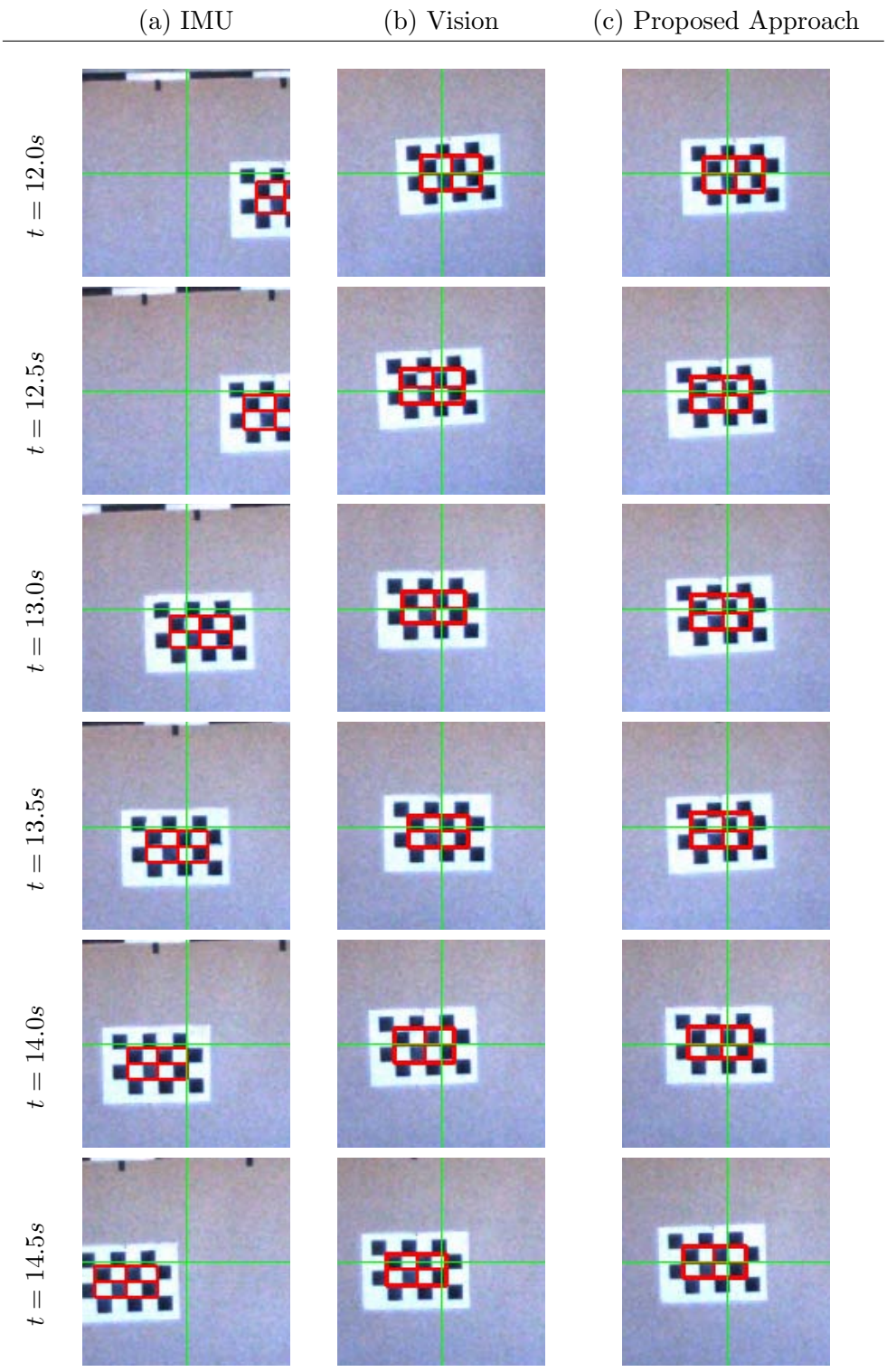


Figure 3.15: Omnidirectional video with translational motion (Sequence (II)). The red cross shows the centre of the centroid, the green cross indicates the centre of the virtual camera.

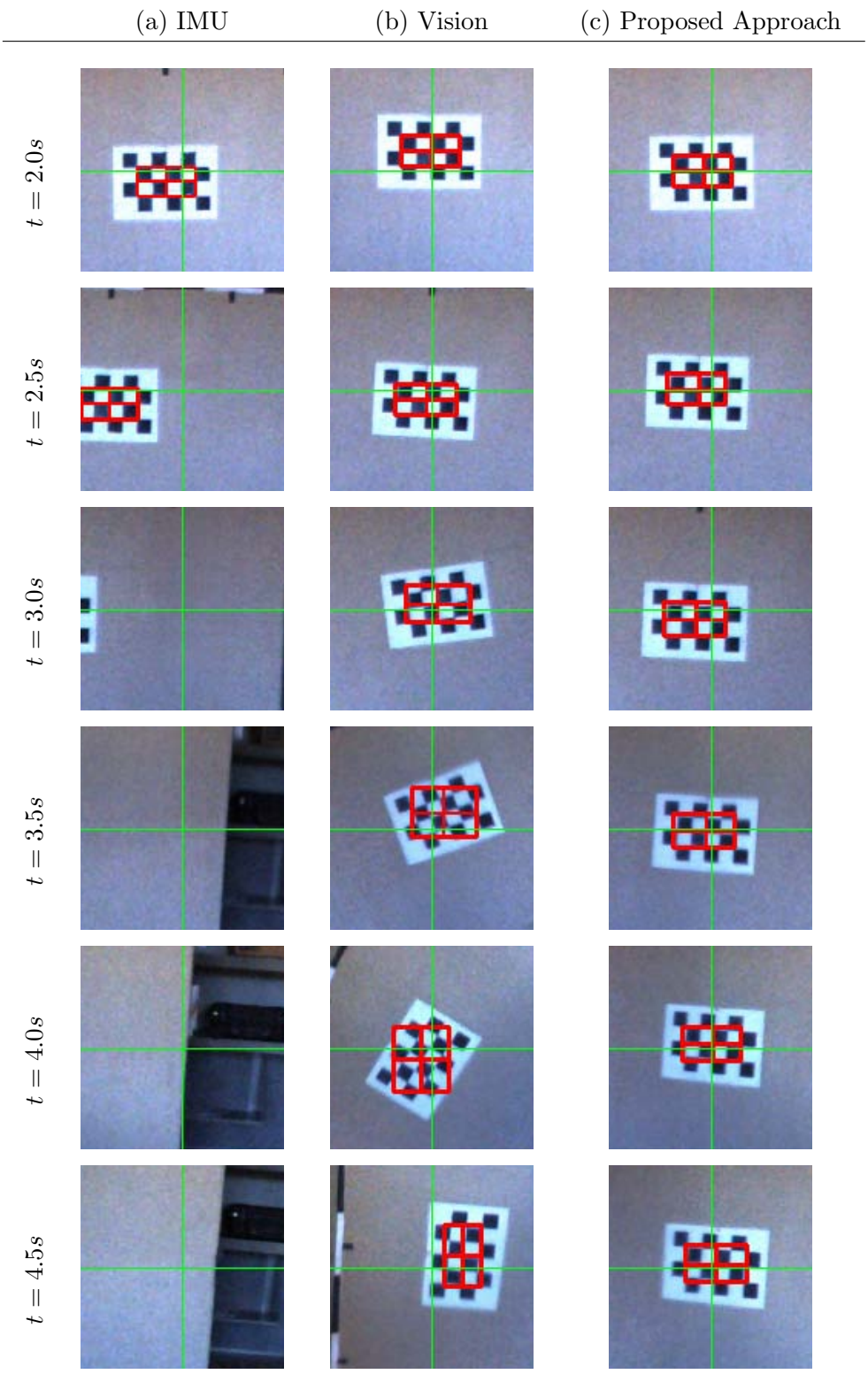


Figure 3.16: Omnidirectional video with combined motion (Sequence (III)). The red cross shows the centre of the centroid, the green cross indicates the centre of the virtual camera.

use of a probabilistic tracking technique allowed loosely calibrated and unsynchronised hardware as the measurement errors can be modelled using uncertainty of the particle filter. A further advantage of the proposed stabilisation approach is that it only requires an estimate for ${}^{C}_{I}\mathbf{T}$ – this allows for quick assembly and disassembly of the hardware without the need for precise alignment.

The chapter concluded with a series of experiments reported in Section 3.4. During the experiments the calibration and synchronisation offsets between both sensors was shown not to be an issue for the proposed approach. Even though they undoubtedly caused an increase in tracking inaccuracy, the particle filter assured that the tracking does not fail during fast rotations. The proposed approach has a consistent error no matter what type of motion occurs, indicating that the proposed approach will be robust under a broad variety of conditions and disturbances.

The experiments performed in this chapter were conducted for precise error measurement and therefore carried out in lab conditions. An application that deals with moving targets, which introduces challenges like the change of target appearance and parallax effects due to large translational offsets is described in Chapter 6, later in this thesis.

Chapter 4

Low-level Features for Maritime Visual Attention

In the previous chapter, an image stabilisation technique that focuses on target objects instead of stabilising entire scenes was developed. The technique enables omnidirectional vision systems to stabilise the view on targets in the presence of significant ego-motion. However, the proposed stabilisation process has to be initialised manually by selecting the target objects. This chapter addresses this shortcoming by developing a framework that automatically identifies regions of visual attention in maritime scenes. This chapter focuses on visual attention of static imagery, an adaption to omnidirectional cameras and video data is presented in Chapter 6.

Selective processing of regions in an image is beneficial not only for stabilisation of a view but also desirable when complex image processing algorithms are used. Mobile platforms are carefully optimised for minimum weight and power consumption to gain maximal mission time and operational range, maneuverability and navigability in shallow waters, easy access to narrow entrances and, in case of a surveillance platform, to make them hard to detect. The use of preprocessing stages that direct attention to regions where more complex image processing algorithms need to be performed and thereby ignore irrelevant areas can reduce the requirements for computational resources on the platform and help to achieve these goals. The alternative, to transmit image data to a base station and perform all image processing tasks offline, is infeasible as transmitting high resolution image data of an omnidirectional camera is limited not only by the range and bandwidth of the radio link, but might also compromise missions where radio silence is desired.

Visual attention can be defined as a problem of detecting parts of an image that stand out in relation to their surrounding regions or the entire image. The proposed framework is positioned as a detector for visual attention tuned for maritime imagery. It consists of multiple low-level features and feature detector cues, independently selected and assessed to respond to specific attributes of maritime objects. Again, emphasis is put on the target domain and the specific structure and appearance of maritime objects are taken

into account to improve performance. However, it is important to note that the *process* of feature construction and fusion is general and could easily be adapted to developing features for other domains.

For the presented framework, the term *Maritime Visual Attention* will be employed since it best describes the intent of the system. The framework outputs a map of the scene, indicating the probability of a region that might contain maritime objects as foreground and thus require further investigation/processing of the surveillance platform.

The proposed framework is intended to be used as an early processing stage or a prelude to higher level processing such as object detection, therefore it has the following requirements:

1. **Highlighting Objects.** The framework should highlight regions that contain maritime objects. Correspondingly, it should suppress areas that contain only background.
2. **Detection of multiple or non-dominant objects.** Saliency detectors concentrate on finding the most dominant object in an image. Scenes from within a harbour or in coastal proximity may have multiple target objects in sight and candidate regions might not always be dominant. Therefore, the framework needs to be able to detect non-dominant objects and if more than one target is present, the framework should not weigh the dominant over the non-dominant.
3. **Robustness to noise.** Maritime scenes potentially contain noise clutter like waves, sunlight glare, etc. The framework should be robust to noise and treat it as background.
4. **Tuned to the domain.** While a generic detector is ubiquitously deployable, the proposed system is intended for use in maritime environments. Therefore, pre-existing knowledge about the general type of target object (not the class but the type) or a model of the background promises to improve the performance of the framework.
5. **Recall performance.** Because the framework is mission critical, it should emphasise recall over precision to ensure potential target objects are not missed. However, recall should not be an exclusive aim. A manageable false alarm rate is still desirable.

This chapter is organised as follows: The following sections describe the proposed design for the maritime visual attention framework. In Section 4.1 Gaussian pyramids are introduced that ensure scale invariance of the approach. Then three different locality cues are

introduced that are used to evaluate low level visual features with respect to local, global, and centre-surround (from Achanta and Süsstrunk (2010)) regions in Section 4.2. Then, a number of low level features are extracted from the input image as described in Section 4.3. These features include responses from an edge detector, frequency components of the image, textural measurements, and distinctiveness in colour. All aforementioned cues result in a probability map per cue and feature. The maps are eventually combined using a Naïve Bayes classifier in Section 4.4. The proposed framework is depicted in Figure 4.1. Section 4.5 is devoted to the experimental evaluation and quantitative comparison to related approaches. The chapter concludes with a summary given in Section 4.6.

Preliminaries

In the following, the RGB coloured input image, $\mathbf{J}$, of height h and width w, is divided in $M \times N$ blocks of $b \times b$ with $b = 8$ being the block size. The group of pixels belonging to the block indexed by (i, j) is defined by the set, B, as

$$B_{ij} = \big\{\big((i-1)\cdot b + k\big) \times \big((j-1)\cdot b + l\big) \;\big|\; k = 1, \ldots, b \text{ and } l = 1, \ldots, b\big\}, \qquad (4.1)$$

with the block indices $i = 1, \ldots, M$ and $j = 1, \ldots, N$.

As described in the following sections, response maps, $\mathbf{Y}$, are computed for every low-level feature, $\mathbf{F}$, and locality cue, $f(\cdot)$, i.e. $f(\mathbf{F}) \rightarrow \mathbf{Y}$.

4.1 Scale Invariance

Target objects in maritime imagery vary in size due to the different physical size of the actual object or the distance to the camera. However, some feature detectors prefer objects at a certain scale (for example kernel based detectors), so scaling effects have to be taken into account when evaluating visual attention in an image (Itti et al., 1998; Liu et al., 2007; Alexe et al., 2010). Hence a Gaussian resolution pyramid is utilised to provide scale-independence for feature analysis. The pyramidal representation of an image (Burt, 1981; Ogden et al., 1985; Lindeberg, 1994; De Bonet, 1997) is created by successive low-pass filtering and sub-sampling. For low-pass filtering the use of a Gaussian kernel is recommended because no new structures are introduced in the sub-sampled, coarser, image (Lindeberg, 1994).

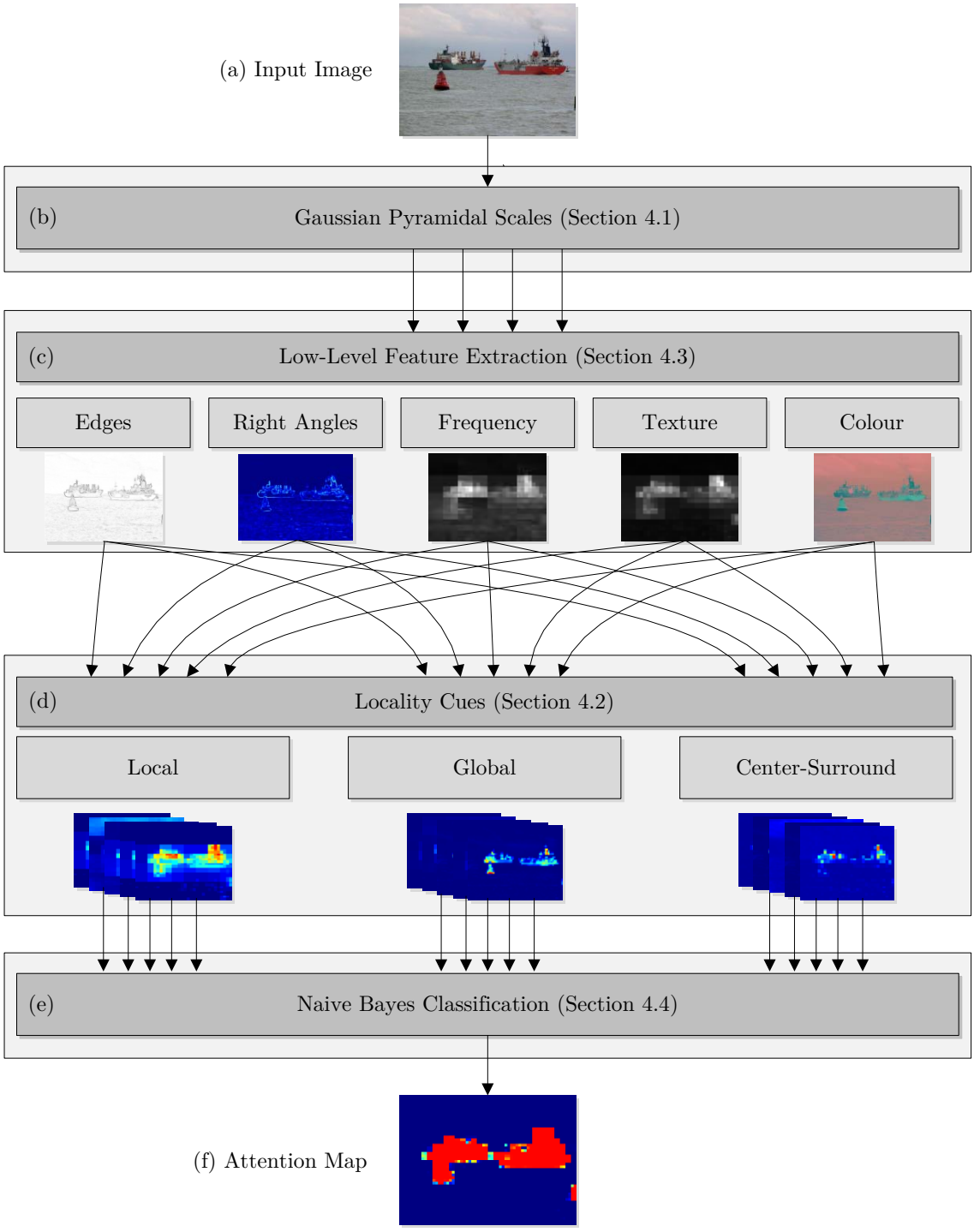


Figure 4.1: **Maritime Visual Attention Framework.** From the input image (a), a number of pyramidal scales are created (b). Low-level features are then extracted from every scale (c) and evaluated using three different locality cues resulting in a probability map per scale, feature, and locality cue (d). All maps are then combined using a Naïve Bayes approach (e), resulting in the final attention map (f).

(a) Original Image (b) Gaussian Pyramids

Figure 4.2: **Gaussian Pyramidal Scales.** An image of scale $\theta = 0$ (original size) is shown in together with four levels of the Gaussian pyramid, $\theta = 1, \ldots, 4$.

The multivariate Gaussian is defined as

$$\frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{\|\boldsymbol{\Sigma}\|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \tag{4.2}$$

With $\boldsymbol{x} = (x, y)$, the two dimensional Gaussian, $G(x, y)$, can therefore be defined by setting $k = 2$. Because x and y are independent, $\boldsymbol{\Sigma} = \mathbf{I}_2$ and therefore $\sqrt{\|\boldsymbol{\Sigma}\|} = 1$. This yields the bivariate function:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{4.3}$$

with σ as the standard deviation of the Gaussian.

In a Gaussian pyramid, the input image is halved in height and width with every level of the pyramid. It is therefore reasonable to approximate the Gaussian function with a discretised Gaussian convolution kernel, $\mathbf{G}$, of size 5×5 and $\sigma = 1.0$:

$$\mathbf{G} = \frac{1}{273} \begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix}. \tag{4.4}$$

Low-pass filtering of the RGB colour image is then performed by independently convoluting each channel, $c \in \{R, G, B\}$, with the Gaussian kernel

$$\hat{\mathbf{J}}_c = \mathbf{G} * \mathbf{J}_c, \tag{4.5}$$

where $*$ is the convolution operator.

From the image of level θ, the coarser image of level $\theta+1$ is then computed by sub-sampling the low-pass filtered image as

$$\mathbf{J}_c^{\theta+1}(x,y) = \hat{\mathbf{J}}_c^{\theta}(2x,2y) \tag{4.6}$$

where $\mathbf{J}_c^{\theta}$ is the image at level θ of the pyramid and $\mathbf{J}_c^0 = \mathbf{J}_c$ and $c \in \{R,G,B\}$.

Substituting Equation (4.5) with (4.6) defines the Gaussian pyramid as

$$\mathbf{J}_c^{\theta+1}(x,y) = [\mathbf{G} * \mathbf{J}_c^{\theta}](2x,2y). \tag{4.7}$$

A sample image with four Gaussian pyramidal levels is depicted in Figure 4.2. The maximum number of scales created in the pyramid depends on the size of the original image. As the size of a pyramidal level is halved in height and width respectively for each scale and the proposed framework makes use of block based measurements, a minimum pyramidal size of four blocks is used in this thesis. This yields the set of scales, Θ, as

$$\Theta = \Big\{ 0, 1, \ldots, \arg\max_{\theta} \Big[\min(w \cdot 2^{-\theta}, h \cdot 2^{-\theta}) \overset{(!)}{\geq} 4 \cdot b \Big] \Big\}, \tag{4.8}$$

for all scales $\theta \in \Theta$ and channels $c \in \{R,G,B\}$.

The size of an image $\mathbf{J}$ is given as $h \times w$, the size of the pyramidal level θ of the image, $\mathbf{J}^{\theta}$ is subsequently given as $h^{\theta} \times w^{\theta}$ with $h^{\theta} = \frac{h}{2^{\theta}}$ and $w^{\theta} = \frac{w}{2^{\theta}}$. Figure 4.2 depicts four pyramidal scales of a sample image.

4.1.1 Across-Scale Summation

Features will be extracted from each scale in the Gaussian pyramid independently. However, to provide scale-independent feature analysis, one must combine the various scales together into a single unified and scale-independent feature map. The approach uses summation across two pyramidal levels of an image $\mathbf{I}^{\theta} + \mathbf{I}^{\theta+1}$, where θ is the finer and $\theta+1$ the coarser level in the pyramid. The summation is performed by expanding the coarser image and a subsequent pixel-by-pixel summation:

$$\mathbf{I}_{xy}^{\theta} + \mathbf{I}_{\hat{x}\hat{y}}^{\theta+1}, \qquad \text{with } \hat{x} = \left\lceil \frac{x}{2} \right\rceil \text{ and } \hat{y} = \left\lceil \frac{y}{2} \right\rceil, \tag{4.9}$$

where $\lceil \cdot \rceil$ denotes the *ceil*-function.

Later in this chapter, summations across all scales of the Gaussian pyramid are performed for feature response maps. This can be efficiently done by repeated expansion and summation beginning at the coarsest scale. For a response map, indicated as $\mathbf{Y}$, this operation is denoted by the $\oplus$ operator:

$$\bigoplus_{\theta \in \Theta} \mathbf{Y} := \sum_{\theta \in \Theta} \mathbf{Y}^{\theta}_{\hat{i}\hat{j}} \qquad \text{with } \hat{i} = \left\lceil \frac{i}{2^{\theta}} \right\rceil \text{ and } \hat{j} = \left\lceil \frac{j}{2^{\theta}} \right\rceil. \tag{4.10}$$

4.2 Locality Cues

(a) Local Cue (b) Global Cue (c) Centre-Surround Cue

Figure 4.3: **Locality Cues.** The figure shows the regions (green) that are considered when evaluating a feature in an image block (red) by the local, global, and centre-surround detector cue. Note that the local cue shown in (a) is actually a density measure, which means that only the 8×8 block itself is considered. The centre-surround region in (c) varies with the spatial location of the reference block, whilst in (b) always the entire image is considered.

A detector cue is a function, $f(\cdot)$, that maps a low-level feature, $\mathbf{F}^{\theta}$, of pyramidal level θ to a response map (essentially a probability map), i.e. $\mathbf{Y} = f(\mathbf{F}^{\theta})$, where $\mathbf{F}^{\theta}$ can be a feature of any kind. The locality cues differ through the use of a different distance metric and the respective region used to compute the probability map. Figure 4.3 shows the regions considered for each of the cues for a sample image. The following three independent locality cues are used to evaluate each of the low-level features and are presented in this section:

Local Cue. The local cue, $f^{L}(\cdot)$, computes a density measure of each feature, emphasising the part of the image with the highest concentration of the respective feature.

Global Cue. The global cue, $f^{G}(\cdot)$, computes the piece-wise difference of a feature for each block of the image.

Centre-Surround Cue. The centre-surround cue, $f^S(\cdot)$, computes the difference of a feature to a surrounding region and is able to detect regional distinctiveness.

4.2.1 Local Cue

Visual attention is not concerned with the recognition or identification of objects or object classes but more with the general detection of regions of interest with the objective that these regions are indicative of objects of interest. Low-level features therefore do not need to be critically evaluated for accuracy and precise location but can be qualitatively assessed. The presence of a feature alone can be a sufficient indication for visual attention at the respective position or the surrounding region. Figure 4.4 shows a maritime scene featuring a sailing boat. Imagine an edge detector that is used to identify ships (and only

(a) Input Image (b) Edge Image (c) Edge Density (Local Cue)

Figure 4.4: **Local Cue.** An edge detector is applied to the input image resulting in the edge image. From this, the edge density is computed using block-wise integration. Note that the edge image shown in (b) and all edge images shown later in this thesis have been inverted and contrast has been adjusted for better visualisation.

ships); common practice is to match contours from the edge image with known shapes (Fonseca and Manjunath, 1996) or build and match descriptors from the edge histogram that represent the object class (Dalal and Triggs, 2005). Either way, the edge image (as the low-level feature) is used to create a high level descriptor and the object is detected (or not) based on descriptor matches. If the actual object class is of secondary interest – a low altitude aeroplane might be something one wants to detect in the image as well as a ship – the region of interest can be identified by the higher density of edges in this area. In the example, anything that would have a more complex shape than the waves and clouds in the image would cause a higher complexity and hence density in the edge image in this region, resulting in a higher value of visual attention and eventually highlight this region of the image.

The proposed local detector cue, $f^L(\cdot)$, is designed to do exactly this. It is applied to a low-level feature response map and computes the density of the feature in question

using block-wise integration. This allows highlighting of blocks that are dominated by the presence of a low-level feature. The low level detector cue, $f^L(\cdot)$ is subsequently defined as

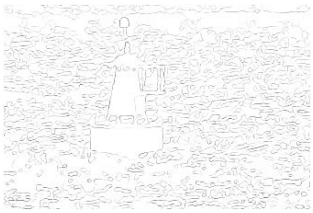
$$f^L_{ij}(\mathbf{F}^\theta) := \sum_{(x,y)\in B_{ij}} \mathbf{F}^\theta_{xy}, \tag{4.11}$$

where (i, j) are the block indices and (x, y) indicate the pixel positions within the feature map as defined in equation (4.1). Figure 4.3(a) shows a sample image with the block that is used for computing the local feature cue highlighted.

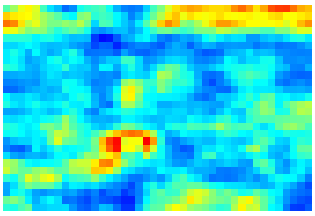
4.2.2 Global Cue

While a high response to a low-level feature in a region may suggest an area of visual attention, the exact opposite is possible as well: a high response density can be caused by a noisy background and the absence of a feature response might be the actual region of interest. Imagine an image depicting a rough sea with a high number of waves and a buoy of a single colour and low texture in the centre as in Figure 4.5. Due to the

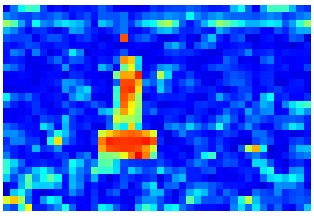
(a) Input Image



(b) Edge Image



(c) Edge Density (Local Cue)



(d) Edge Uniqueness (Global Cue)

Figure 4.5: **Global Cue.** Applying an edge detector to the input image, results in the edge image from which the edge density is computed using the local detector cue. The uniqueness of the edge feature is computed by the piece-wise distance of the edge density. Note that the points of high value at the base of the buoy in (c) and (d) are not at the very same position of the image. The local cue (c) highlights the strong border of the body, while the global cue (d) highlights the absense of edges within the body.

number of waves, the edge image will be highly pronounced in the vicinity of waves and

the buoy will have a well defined contour. However, the buoy itself will have no edges present because of its smooth shape. Applying the local detector cue on the edge image will thus highlight the background and suppress the buoy. Whether it is the presence or absence of a feature, common for both is that the region of interest is different from the image; in other words the region of interest has a high uniqueness compared to the rest of the image. To compute the factor of uniqueness of a block of the image with respect to a given feature, the difference of the feature response between the block and the rest of the image is evaluated.

The global detector cue, $f^G(\cdot)$, is subsequently defined as the sum of the squared distances between each block to perform a piece-wise comparison of each block with the rest of the image to identify regions within the image that are unique with respect to the feature:

$$f_{ij}^G(\mathbf{F}^\theta) := \sum_{k=1}^{M^\theta} \sum_{l=1}^{N^\theta} \left\| \sum_{(x,y)\in B_{kl}} \mathbf{F}_{xy}^\theta - \sum_{(x,y)\in B_{ij}} \mathbf{F}_{xy}^\theta \right\|^2, \tag{4.12}$$

where (i, j) and (k, l) are block indices within the feature map. Figure 4.3(b) shows a block together with the global region within a sample image.

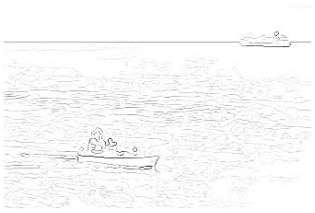
A global measurement has been previously used to find unique objects in an image: Liu et al. (2007) use a global measurement to identify salient objects by comparing the spatial variance with respect to the spatial distribution of a colour. Achanta et al. (2009) find salient regions in an image by comparing the CIELAB vector of every pixel with the image mean vector.

4.2.3 Centre-Surround Cue

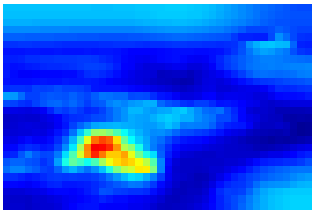
Features that are highly distinctive or features that have a unique presence (or absence) in an image can be detected by the local or global detector cue respectively. However, both measurements imply that at least one region of interest exists within the image – the region with the highest feature density or the highest feature uniqueness respectively.

Figure 4.6 shows an image that depicts a relatively calm sea with two ships: a pilot boat in the front and a cruise ship on the horizon of the image. The edge image reveals the silhouettes of both ships. When applying the local detector cue, the pilot boat is clearly highlighted due to the higher density of edges in this area. However, the cruise ship in the distance, even though it has a certain structure and response in the edge image, is marked as background. This is due to the much higher density of edges of the pilot boat, which

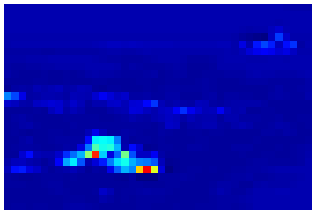
(a) Input Image



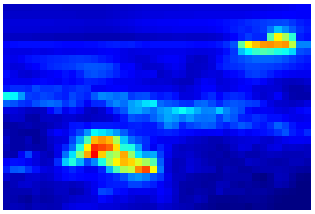
(b) Edge Image



(c) Local Cue



(d) Global Cue



(e) Centre-Surround Cue

Figure 4.6: **Centre-Surround Cue.** A surrounding region is extracted for each block of the edge image, which is then used for comparison in the centre-surround cue.

suppresses the local maximum of edge density in vicinity of the cruise ship. The global detector cue yields a similar result. It emphasises the most unique part of an image with respect to the feature, which in this case is within the proximity of the pilot boat.

Comparing a region with a surrounding sub window instead of the entire image is more likely to overcome this as it allows comparing parts independently. For the cruise ship this would mean that only the edge density within the proximity of the ship is used for comparison and the density within the area of the pilot boat would not bias the response. Itti et al. (1998) suggested creating multiple scales of an image and compared a pixel with the same pixel in a larger scale as the surrounding region. A centre-surround window is also used by Liu et al. (2007). The authors compared a reference region with its annulus where the spatial position of the centre matches the centre of the reference region and the area of the annulus is the same area as the area of the reference region. Achanta and Süsstrunk (2010) proposed the use of a maximum symmetric centre-surrounding sub window, which acts as a band pass filter for the spatial frequency. They showed that adjusting the low cut-off frequency depending on the spatial location improves detection of saliency for pixels that are far from object borders and that varying the window size depending on the spatial location outperforms the aforementioned techniques. Furthermore, due to the use of a maximum symmetric centre-surrounding sub window, no annulus needs to be evaluated. It is therefore used for the proposed centre-surround detector cue.

For a map of size $M^\theta \times N^\theta$ from the pyramidal level θ, the surrounding window of a

reference block (i, j) is spanned by the maximum symmetric distance, m_i^θ and n_j^θ:

$$m_i^\theta = \min(i, M^\theta - i) \text{ and } n_j^\theta = \min(j, N^\theta - j). \tag{4.13}$$

The set of blocks belonging to the surrounding window for a reference block (i, j) is subsequently defined as

$$S_{ij}^\theta = \left\{ \left((i - m_i^\theta + 1, \ldots, i + m_i^\theta - 1) \times (j - n_j^\theta + 1, \ldots, j + n_j^\theta - 1) \right) \mid m_i^\theta,\ n_j^\theta \right\}, \tag{4.14}$$

Then, the mean, $\overline{\mathbf{F}}^\theta$, of a feature, $\mathbf{F}^\theta$, in the maximum symmetric surrounding window of (i, j) can be computed as

$$\overline{\mathbf{F}}_{ij}^\theta = \underbrace{\left((2m_i^\theta + 1)(2n_j^\theta + 1) \right)}_{A = \#(S_{ij}^\theta)}^{-1} \sum_{(x,y) \in S_{ij}^\theta} \mathbf{F}_{xy}^\theta, \tag{4.15}$$

where A is the area of the maximum symmetric surrounding window, that is also the number of blocks in S_{ij}^θ. Note that (i, j) are block indices and $\overline{\mathbf{F}}_{ij}^\theta$ represents a block within the feature map of pyramidal level θ, not a pixel. The size of $\overline{\mathbf{F}}^\theta$ is thus given as $M^\theta \times N^\theta$. The Euclidean distance between the features at (i, j) and the mean of its maximum symmetric surrounding window then yields the centre-surround detector cue for a feature, $\mathbf{F}^\theta$,

$$f_{ij}^S(\mathbf{F}^\theta) := \left\| \overline{\mathbf{F}}_{ij}^\theta - \sum_{(x,y) \in B_{ij}} \mathbf{F}_{xy}^\theta \right\|^2. \tag{4.16}$$

A block (i, j) is shown in Figure 4.3(c) together with the matching centre-surround window in which the green region is defined by the limits of the image and is symmetric in width and height about the block of interest.

4.3 Low-Level Features

This section introduces the low-level features used within the proposed visual attention framework. The idea is to design a number of features that each respond to specific attributes of maritime objects. Preliminary observations showed that different features respond to different objects or parts of objects. While the response of a single feature alone cannot detect maritime objects, a combination of features might reveal their presence. The low-level features described in the following are used because they are simple and easy to compute and do not require a specific format for the input image. They are independently assessed and visualised using heatmaps. All features discussed in this section are used as

possible candidates in this chapter – later in this thesis this approach will be extended and a machine learning approach will be utilised to select the most relevant (see Chapter 5).

The following low-level features are presented in this section:

Edges. The density of edges is used as a measure of overall structure in parts of the image.

Right Angles. A right angle feature sensitive to orthogonal edges is used to emphasise regions that contain man-made structures.

Frequency. The density of high frequency components is used to identify "noisy" regions in the image.

Texture. Texture is used to detect irregularities in areas.

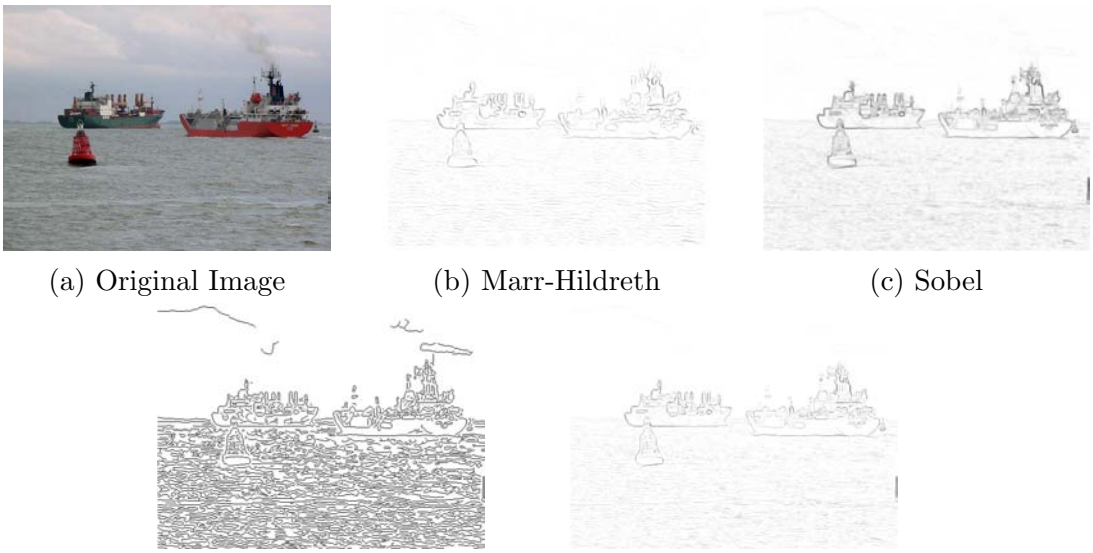
Colour. Colour is used to identify regions with a unique colour compared to their surroundings or the rest of the image.

For performance visualisation, the response of each of the low-level features using the three locality cues (local, global, and centre-surround) are shown on a number of test images. From the dataset, these images are manually selected to represent a wide range of performance for each of the features and cues and with good and poor detection response with respect to maritime objects. Here, a good detection response does not necessarily mean that all objects in the image have to be detected with a near perfect performance, but only that corresponding regions are highlighted and the background is suppressed. In fact, as the features are designed to be used only in combination, a good detection response is used in terms of a qualitative measure and a high recall is favoured over a high precision for the detector.

Test images that yield a poor detection response with a feature or locality cue are also evaluated using a different feature that yields a good detection response – note the cross-reference under these images. This illustrates the different performance of each feature for different images and the need to use more than one feature in combination. The performance of the features are analysed and discussed at the end of each subsection.

4.3.1 Edge Based Features

In computer vision, edge detectors are used to estimate structural components and separate areas of similar intensity of an image by highlighting the separating gradient between the areas. The presence of edges in a region of an image may indicate structured components and thus is an important feature in the proposed framework. A number of edge detectors have been proposed in the past (see Figure 4.7), the most notable being a combination of the convolution of image intensity by the second-order Gaussian and zero-crossings as proposed by Marr and Hildreth (1980) or image kernel filters like the *Sobel*-Operator (Ballard and Brown, 1982).



(a) Original Image (b) Marr-Hildreth (c) Sobel

(d) Canny (e) Canny weighted with Sobel

Figure 4.7: **Edge Detectors.** Response of different edge detectors to a test image.

The Canny edge detector (Canny, 1986) smoothes the intensity image with a two dimensional Gaussian and computes the image gradient by applying the Sobel detector in horizontal and vertical directions. Following this, a hysteresis threshold is applied to accept strong and discard weak edges. The Canny edge detector has become the *de facto* standard because of its insensitivity to noise and low error rate, outperforming the aforementioned techniques. It is therefore used in this thesis with a subsequent multiplication of the edge image with the magnitude of the image gradient, yielding an edge image that is weighted by the strength of the edges. To compute the edge image, $\mathbf{E}^\theta$, of the pyramidal level θ, first the intensity image, $\mathbf{I}^\theta$, needs to be computed from the RGB image, $\mathbf{J}^\theta$. Using the weighting factors suggested by Fairchild (2005) yields

$$\mathbf{I}^\theta = 0.2985 \cdot \mathbf{J}_R^\theta + 0.5870 \cdot \mathbf{J}_G^\theta + 0.1140 \cdot \mathbf{J}_B^\theta. \tag{4.17}$$

The image gradients, $\mathbf{G}_x^\theta$ and $\mathbf{G}_y^\theta$, of each pyramidal level $\theta \in \Theta$ are computed from the intensity image using the Sobel operator in the vertical and horizontal directions:

$$\mathbf{G}_x^\theta = (1,2,1)^T * \big((1,0,-1) * \mathbf{I}^\theta\big) \tag{4.18a}$$

and

$$\mathbf{G}_y^\theta = (1,0,-1)^T * \big((1,2,1) * \mathbf{I}^\theta\big). \tag{4.18b}$$

The edge image for the pyramidal levels $\theta \in \Theta$ is then computed as

$$\mathbf{E}^\theta = \mathit{Canny}(\mathbf{I}^\theta) \cdot \sqrt{\big(\mathbf{G}_x^\theta\big)^2 + \big(\mathbf{G}_y^\theta\big)^2}, \tag{4.19}$$

yielding the set of edge images $\mathcal{E} = \big\{\mathbf{E}^\theta \mid \theta \in \Theta\big\}$.

As mentioned before, a dense presence of edges might indicate a structured content within the corresponding region and thus suggest the presence of man-made objects. This feature is related to the edge density measure suggested for object detection by Alexe et al. (2010). However, Alexe et al. compare the edge density of a window in relation to its surrounding region to predict whether the window contains an entire object with a closed boundary – a significant limitation given the presence of noise and texture in maritime scenes. The proposed feature is used to predict the presence of any kind of structure. It is evaluated using three different locality cues at multiple scales. An edge density measure is computed using the local locality cue as

$$\mathbf{Y}_E^L(\mathbf{I}) := f_{ij}^L(\mathcal{E}) = \bigoplus_{\theta\in\Theta} \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{E}_{xy}^\theta. \tag{4.20a}$$

An evaluation based purely on edge density might highlight a noisy background if it has a higher density of edges than the foreground object. In a highly structured image, the focus of attention may therefore be on the area with low edge density. Hence, the dissimilarity in edge density is used as another measure. It is computed using the edge image as an input for the global locality cue, highlighting the region in the image that is most distinctive compared to the rest of the image as

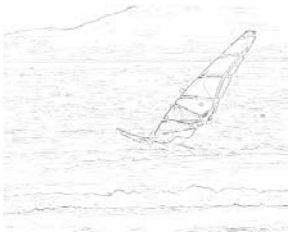
$$\mathbf{Y}_E^G(\mathbf{I}) := f_{ij}^G(\mathcal{E}) = \bigoplus_{\theta\in\Theta} \sum_{k=1}^{M} \sum_{l=1}^{N} \Big\| \sum_{\substack{(x,y)\\ \in B_{kl}}} \mathbf{E}_{xy}^\theta - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{E}_{xy}^\theta \Big\|^2. \tag{4.20b}$$

The result of the global measurement, however, depends on the edge density of the rest of the image and number of targets. As established in Section 4.2.3, the comparison with the surrounding window will highlight the target regardless. The regional edge feature is computed using the centre-surround cue as

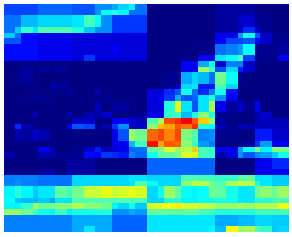
$$\mathbf{Y}_E^S(\mathbf{I}) := f_{ij}^S(\mathcal{E}) = \bigoplus_{\theta\in\Theta} \left\| \overline{\mathbf{E}}_{ij}^{\theta} - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{E}_{xy}^{\theta} \right\|^2, \tag{4.20c}$$

where $\overline{\mathbf{E}}_{ij}^{\theta}$ is the mean edge density of the window surrounding (i, j) of pyramidal level θ as described in Section 4.2.3 and computed following Equation (4.15), where $\overline{\mathbf{F}}^{\theta}$ is $\overline{\mathbf{E}}^{\theta}$.

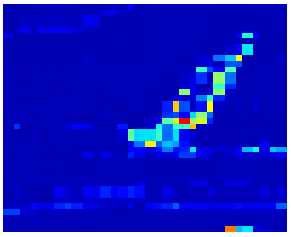
(a) Original image



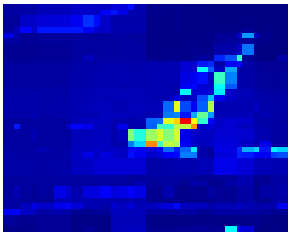
(b) Edge image



(c) Local edge feature



(d) Global edge feature



(e) Centre-surround edge feature

Figure 4.8: **Edge feature.** Evaluation of the edge feature by the local, global, and centre-surround locality cues.

Figure 4.8 shows the responses of the edge feature for the local, global, and centre-surround locality cues.

Preliminary Analysis

Figure 4.9 shows the response of the local, global, and centre-surround cues on the edge feature on a number of test images. The locality cues give a good response for the test images in (a)–(f), indicating the presence of an object in the respective region of the image. For the test images in (b) and (d), the detector over segments, in (b) parts of the bottom of the image are highlighted due to the highly structured waves in this part of the image,

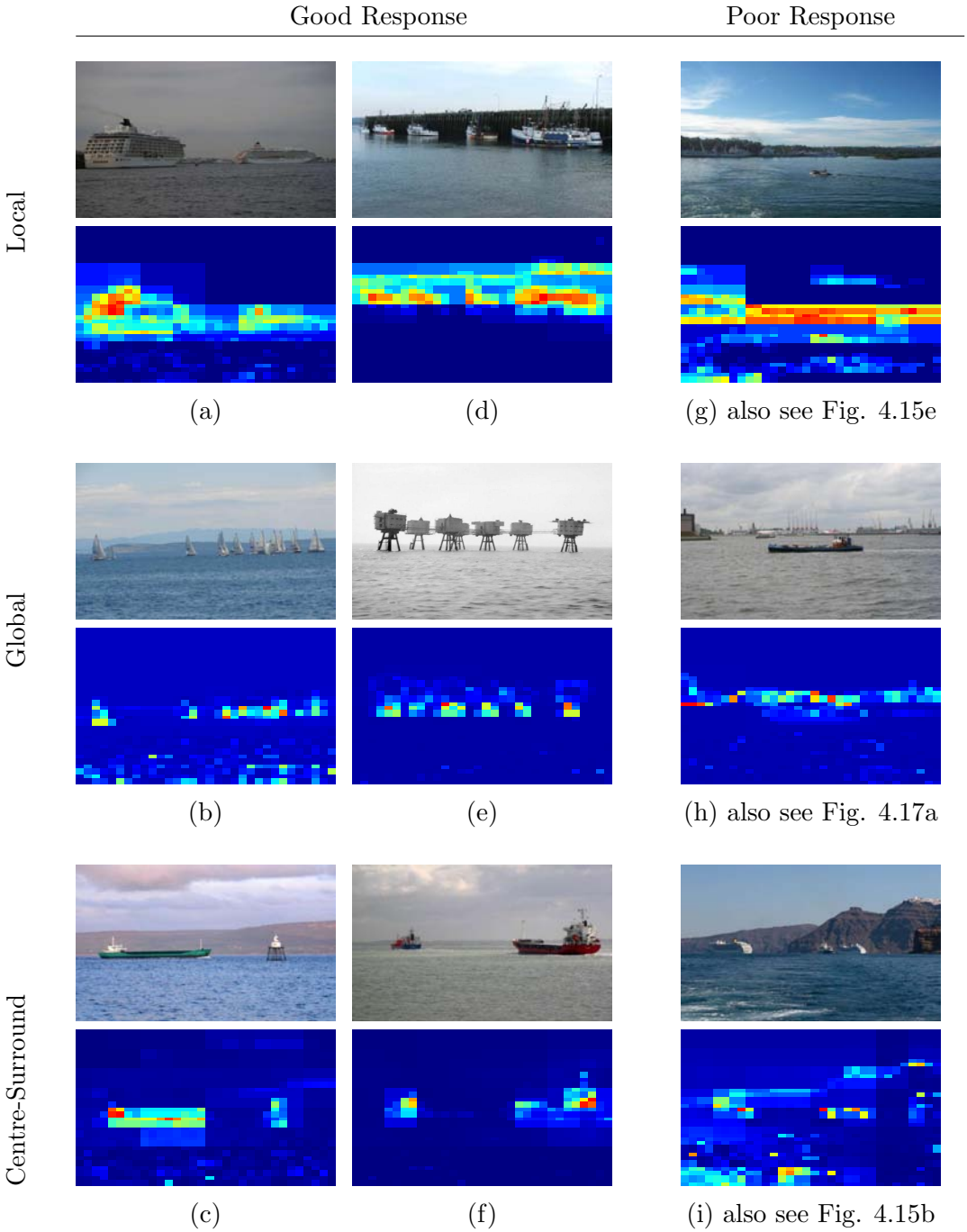


Figure 4.9: **Responses to edge feature.** The local, global, and centre-surround detector cues are used to evaluate the edge feature on a number of test images. The heatmaps in (a)–(f) reveal a good response to the test images with respect to maritime objects while heatmaps (g)–(i) expose a poor performance when using the edge feature. The cross references for these heatmaps refer to features yielding a good performance on the respective test images.

while in (d) the oversegmentation is due to the high presence of edges on the pier.

The test images in (a), (b), and (d) have also been evaluated using other features yielding a poor performance: (a) has been evaluated using the frequency feature with the centre-surround cue (Figure 4.15(i)) but due to a lack of disparity in frequency components, only parts of the two ships were highlighted. However, as shown in Figure 4.9(a), the density of edges proved to be sufficient to detect the target objects. Both, (b) and (d), yielded poor performance using the colour feature in combination with the local and the centre-surround locality cues – as shown in Figure 4.19(g) and (h) respectively. The colour of the sailing ships in 4.19(g) did not show enough distinctiveness from the image mean to detect the objects. However, the number of edges on the objects allowed detection using the global edge cue as shown in Figure 4.9(b).

With a higher density of edges on the objects than on the surrounding background region, the test image in (d) is correctly segmented using the local edge cue, contrary to the centre-surround colour feature, where multiple parts of the pier get highlighted as foreground objects. Even though the test images depicted in (c), (e), and (f) show a noisy sea and also some background (c) and textured cloud coverage ((c) and (f)), the contour of the target objects produces well defined edges that provide enough uniqueness for the edge feature to detect the objects.

However, the test images in (g)–(i) yield a poor response of the locality cues based on the edge feature, the reason for the poor response is the spread out dominance of edges in the background of the images. Note that the test images (g) and (i) are also evaluated using the frequency feature in Figure 4.15, while (h) is tested using the colour feature in Figure 4.19.

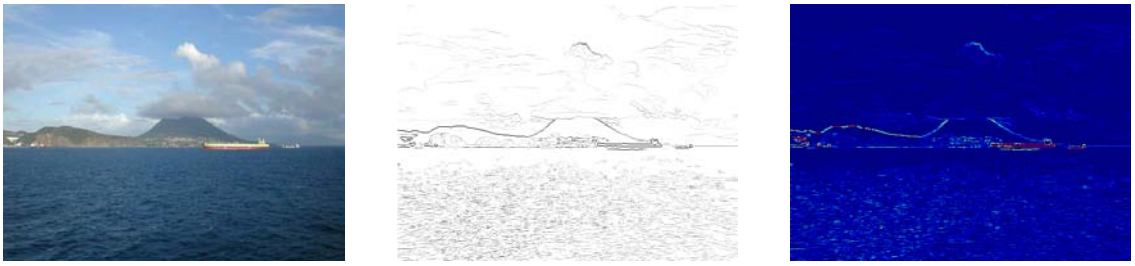
Right Angles

The presence of edges is a first indicator for structure or texture. However, in coastal regions, images are likely to contain background that will respond to the edge detector, oversegment, and thus affect detection accuracy of actual target objects. A right angle filter is created as another low level feature, as right angles are more dominant in (man-made) structures than in natural scenes. A kernel that is sensitive to horizontal and vertical lines is defined and convolved with the edge image resulting in the right angle

feature:

$$\mathbf{R}^{\theta} = \frac{1}{16} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 2 & 4 & 2 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} * \mathbf{E}^{\theta}, \tag{4.21}$$

where $\mathbf{E}^{\theta}$ is the edge image of pyramidal level θ as computed in Equation (4.19), yielding the set of right angle responses $\mathcal{R} = \{\mathbf{R}^{\theta} \mid \theta \in \Theta\}$.



(a) Original Image (b) Edge Image (c) Right Angle Response

Figure 4.10: **Right Angle Detector.** The response of the right angle detector highlights the parts of the edge image that have right angles and mostly horizontal or vertical lines.

The Canny edge detector results in an edge image with fine lines, i.e. edges are marked with only one pixel in width making it ideal to be used with this filter. Because the imagery originates from a stabilised recording platform, a kernel filter is a sufficient detector as the stabilisation process ensures horizontal and vertical lines in the scene are properly aligned in the image. However, the 5×5 kernel does allow some inaccuracy of orientation. Figure 4.10 shows a sample image and the edge image as computed by the Canny edge detector together with the response of the proposed right angle filter. One can see that the contour of the target ship is correctly identified by the Canny edge detector (Figure 4.10(a)), however, the edges in the background have similar weighting as the target. Figure 4.10(c) shows the result of the subsequently applied right angle detector. Here, the horizontal components of the target ship are weighted higher than the rest of the image. It should be noted that the detector also responds to the contours of the hills due to the "pixel-stepping" of the diagonal. This effect is visible most at high resolution

(e.g. pyramidal level $\theta = 0$) and small kernel size.

Using the local locality cue, the density of the right angle feature is computed as

$$\mathbf{Y}_R^L(\mathbf{I}) := f_{ij}^L(\mathcal{R}) = \bigoplus_{\theta\in\Theta} \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{R}_{xy}^{\theta}, \tag{4.22a}$$

as well as a dissimilarity measure based on the global locality cue as

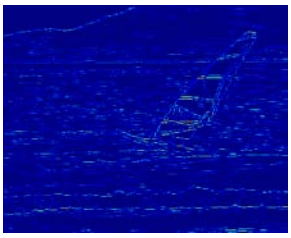
$$\mathbf{Y}_R^G(\mathbf{I}) := f_{ij}^G(\mathcal{R}) = \bigoplus_{\theta\in\Theta} \sum_{k=1}^{M}\sum_{l=1}^{N} \left\| \sum_{\substack{(x,y)\\ \in B_{kl}}} \mathbf{R}_{xy}^{\theta} - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{R}_{xy}^{\theta} \right\|^2. \tag{4.22b}$$

The centre-surround cue for the right angle feature is subsequently defined as

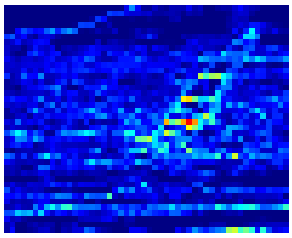
$$\mathbf{Y}_R^S(\mathbf{I}) := f_{ij}^S(\mathcal{R}) = \bigoplus_{\theta\in\Theta} \left\| \overline{\mathbf{R}}_{ij}^{\theta} - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{R}_{xy}^{\theta} \right\|^2, \tag{4.22c}$$

where $\overline{\mathbf{R}}^{\theta}$ is the mean edge density of the surrounding window of pyramidal level θ as described in Section 4.2.3 and computed using Equation (4.15) where $\overline{\mathbf{F}}^{\theta}$ is $\overline{\mathbf{R}}^{\theta}$.

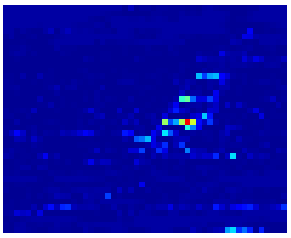
(a) Original image



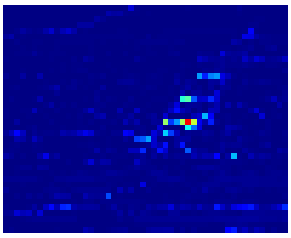
(b) Right angle response



(c) Local right angle feature



(d) Global right angle feature



(e) Centre-surround right angle feature

Figure 4.11: **Right angle feature.** Evaluation of the right angle feature by the local, global, and centre-surround locality cues.

The responses of the right angle feature for the local, global, and centre-surround locality cues are shown in Figure 4.11.

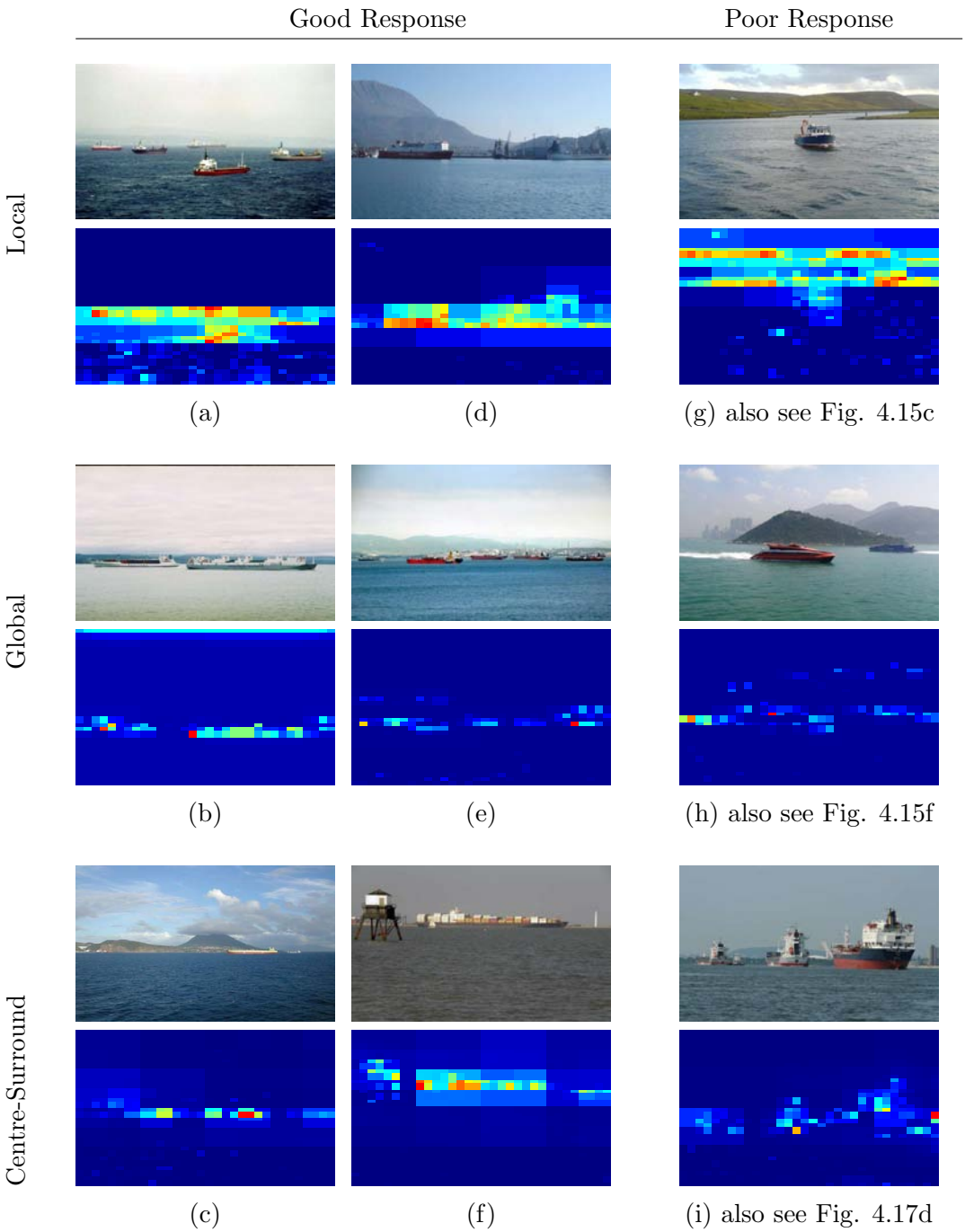


Figure 4.12: **Responses to right angle feature.** The local, global, and centre-surround detector cues are used to evaluate the *right angle* feature on a number of test images. The heatmaps in (a)–(f) reveal a good response to the test images with respect to maritime objects while heatmaps (g)–(i) expose a poor performance when using the right angle feature. The cross references for these heatmaps refer to features yielding a good performance on the respective test images.

Preliminary Analysis

The response of the three locality cues for the right angle feature is shown in Figure 4.12. All test images contain a high density of edges, however, convolving with the right angle kernel, yields an emphasis of the man-made structural components of the images. This works well for the test images in (a)–(f), where the feature shows a good response with respect to maritime objects.

The test image in (a) has a good recall performance, however the response shows that some oversegmentation is present. This test image has also been assessed using the colour feature in Figure 4.19(h), where it yielded a poor performance as piece-wise distance measurement caused a highlighting of the dark parts in the left and right bottom of the image instead of the maritime objects. The images in Figures 4.12(c), (d), and (e) all have a structured background that is visible in the edge image. However, the subsequently applied right angle kernel is able to distinguish between the contour of the hills and clouds and the small ship that consists of mostly vertical elements.

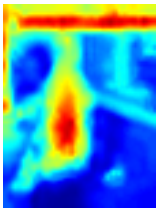
However, a poor performance of the right angle feature is shown for the test images in (g)–(i). In (g), the strong edges in the background (separation between hills and sky) and the lack of right angles on the target object lead to a misdetection. For the test images in (h) and (i) the detector produces mostly noise. In both images, a sharp peak can be observed; in (h) the peak is located on the ship's horizontal ornamental strip, in (i), the peak is located on an building close to the right image border. All three images are also evaluated using different features, where the segmentation yields better results – Figure 4.15(c) and (f) show the responses of the frequency feature for the test images in Figures 4.12(g) and (h) respectively, while (i) is evaluated using the textural feature in Figure 4.17(d).

4.3.2 Frequency based Features

The edge feature proposed in section 4.3.1 is used to detect boundaries between areas with different intensity, that might suggest boundaries of an object. Localised abrupt changes in intensity cannot be reliably detected with this feature. Consider a target object in front of a highly structured background as shown in Figure 4.13, where a tall ship is about to pass through a bridge in London's Upper Pool. The tall ship is highly structured, however the background, especially the upper part of the bridge is as well. In fact, due to the high contrast between the upper part of the bridge and the background in this region, it has

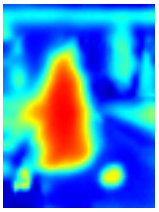
(a) Input image

(b) Canny weighted with Sobel



(c) Local edge feature

(d) High-pass filtered image



(e) Local frequency feature

Figure 4.13: **Comparison of local edge and frequency feature.** The local edge feature is computed using the Sobel weighted Canny edge detector. The high frequency components of the input image are shown together with the local frequency feature.

well pronounced edges, resulting in a high weighting of the Sobel weighted Canny edge detector operator as shown in (b). The result of the local edge feature (density) is depicted in (c). A closer inspection of the structure of the tall ship reveals that it is caused by the rigging of the ship.

A high frequency feature is proposed to detect these *noisy* regions within an object. However, while this means areas of sea with highly pronounced waves might be highlighted by this detector, areas of sky will be suppressed since it is typical that low frequencies dominate in those parts of the image.

The high frequency components of the input image can be computed as

$$\mathbf{D}^{\theta} = \nabla^2 \mathbf{I}^{\theta}, \tag{4.23}$$

where $\mathbf{I}^{\theta}$ is the intensity image of pyramidal level θ and ∇^2 is the Laplacian, yielding the set of high frequency responses $\mathcal{D} = \left\{ \mathbf{D}^{\theta} \mid \theta \in \Theta \right\}$.

From this, the local locality cue that estimates the density of high frequency components for each block (i, j) of the image can be computed as

$$\mathbf{Y}_F^L(\mathbf{I}) := f_{ij}^L(\mathcal{D}) = \bigoplus_{\theta \in \Theta} \sum_{\substack{(x,y) \\ \in B_{ij}}} \mathbf{D}_{xy}^{\theta}. \tag{4.24a}$$

The global locality cue, estimating the dissimilarity of high frequency density between

each block of the image is subsequently defined as the piece-wise distance,

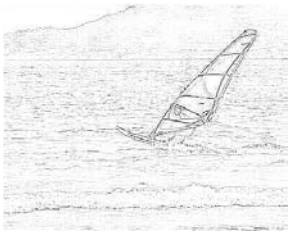
$$\mathbf{Y}_F^G(\mathbf{I}) := f_{ij}^G(\mathcal{D}) = \bigoplus_{\theta\in\Theta} \sum_{k=1}^{M^\theta} \sum_{l=1}^{N^\theta} \Big\| \sum_{\substack{(x,y)\\ \in B_{kl}}} \mathbf{D}_{xy}^\theta - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{D}_{xy}^\theta \Big\|^2, \tag{4.24b}$$

leaving the centre-surround cue as

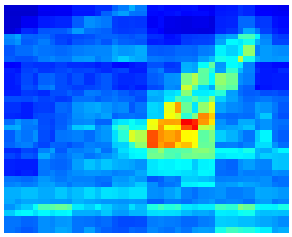
$$\mathbf{Y}_F^S(\mathbf{I}) := f_{ij}^S(\mathcal{D}) = \bigoplus_{\theta\in\Theta} \Big\| \overline{\mathbf{D}}_{ij}^\theta - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{D}_{xy}^\theta \Big\|^2, \tag{4.24c}$$

where $\overline{\mathbf{D}}_i^\theta j$ is the mean edge density of the window surrounding (i, j) of pyramidal level θ as described in Section 4.2.3 and computed using Equation (4.15) where $\overline{\mathbf{F}}^\theta$ is $\overline{\mathbf{D}}^\theta$. Figure 4.15 shows the response of this detector. Figure 4.14 shows the responses of the

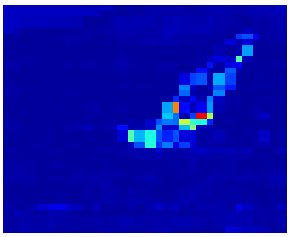
(a) Original image



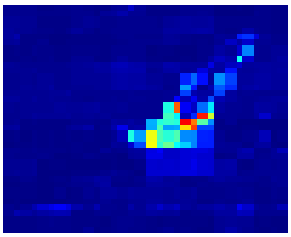
(b) Laplacian image



(c) Local frequency feature



(d) Global frequency feature



(e) Centre-surround frequency feature

Figure 4.14: **Frequency feature.** Evaluation of the frequency feature by the local, global, and centre-surround locality cues.

frequency feature for the local, global, and centre-surround locality cue respectively.

Preliminary Analysis

The responses to the local, global, and centre-surround locality cues of the frequency feature are shown in Figure 4.15. The feature shows a good response for the test images in (a)–(f), highlighting the maritime objects. These test images are also evaluated using

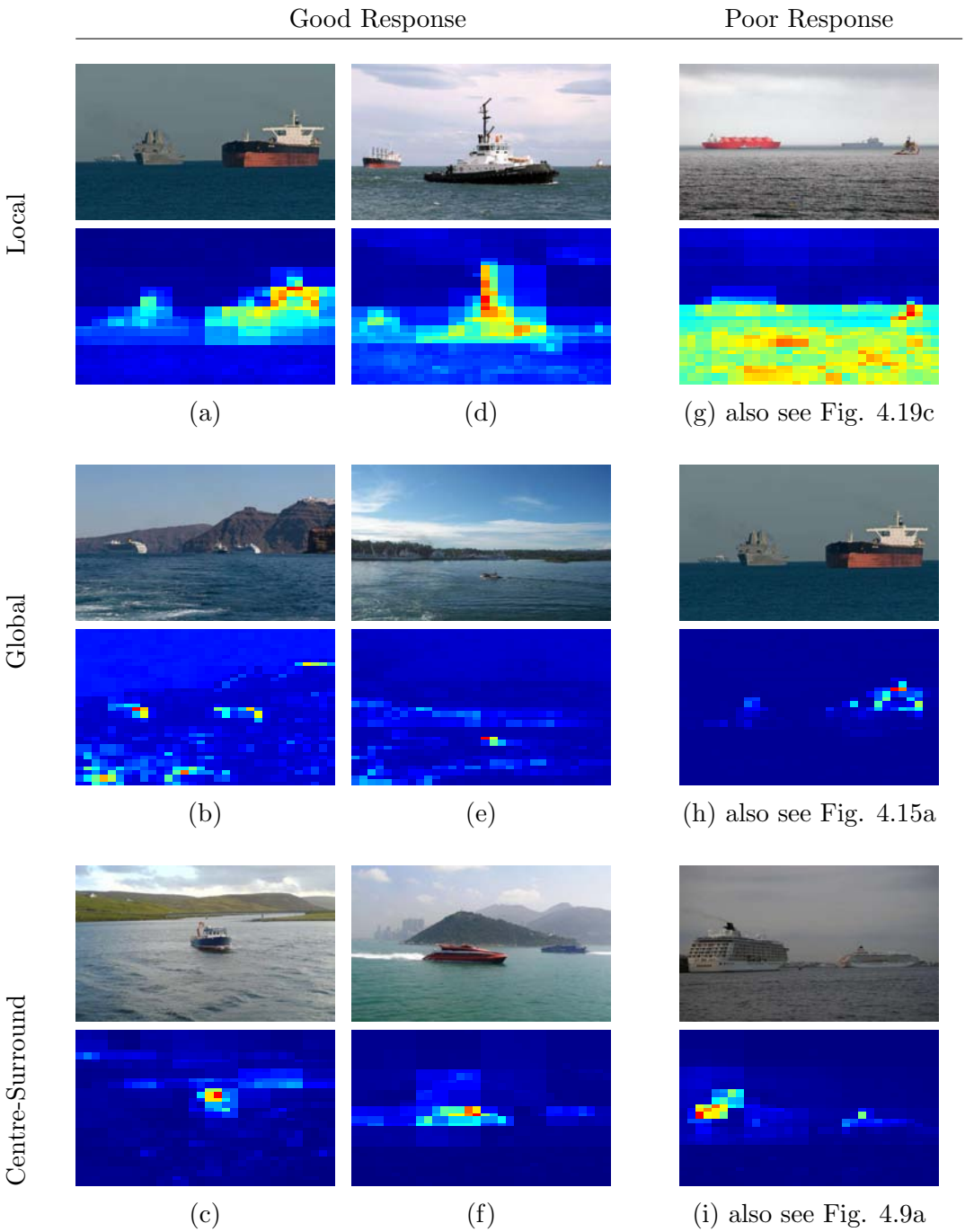


Figure 4.15: **Responses to frequency feature.** The local, global, and centre-surround detector cues are used to evaluate the *frequency* feature on a number of test images. The heatmaps in (a)–(f) reveal a good response to the test images with respect to maritime objects while heatmaps (g)–(i) expose a poor performance when using the frequency feature. The cross references for these heatmaps refer to features or other detector cues yielding a good performance on the respective test images.

other features: the image in (d) is evaluated using the global textural feature as shown in Figure 4.17(h), where the mast of the ship has a significantly higher response than the rest of the object(s) so that these get suppressed in the response map. However, the density of high frequency components in both targets allow a reliable detection using the local cue of the frequency feature as depicted in (d). The images in (b) and (e) are both tested using the edge feature as shown in Figure 4.9(i) and (g) respectively. While the global frequency feature yields a reliable detection of the maritime objects in both images, the edge feature was not able to pick up the objects due to a high number of edges in the background. Although the global locality cue produces some noise and false positives on the waves in the bottom part and on top of the hills in the top-right part of the test image in Figure 4.15(b) due to the high frequency components in these areas, the overall performance of the feature is satisfactory. The ship in (e) is detected with a high accuracy due to the mostly low frequencies in the image.

Likewise, the centre-surround cue of the high frequency feature is able to pick up the maritime object in (c) and (f) due to the difference of the frequency components on the objects and their surrounding region. Both images were tested using the right angle feature as shown in Figure 4.12(g) and (h) respectively. However, the right angle feature yielded a poorer response due to the absence of strong vertical and horizontal edges in the images.

Note that Figure 4.15(a) and (h) depict the same test image. It is evaluated using the same frequency feature but using different locality cues. In (h), the global locality cue is used. Here, the high frequency component is compared to the rest of the image in a piece-wise manner, yielding a poor detection of the oil tanker's hull and failing to detect the indistinct ship to its left. This poor performance is due to only small and almost uniform differences in the frequencies such that the global locality cue could not identify a specific region of uniqueness. When only evaluating the density as with the local locality is enough to highlight both objects as shown in (a).

Furthermore, the local locality cue in Figure 4.15(g) yields a poor result, highlighting most of the waves in the bottom of the test image due to their highly structured appearance and thus high density of high frequency components in this area. The image is also tested using the colour feature in Figure 4.19(c) yielding a good result. Due to lack of disparity in frequency components within the surrounding region of the two ships in Figure 4.15(i), only parts of the objects are highlighted yielding a poor overall performance; the edge feature is also used to evaluate this image, shown in Figure 4.9(a) with satisfactory performance.

4.3.3 Textural Features

Texture is a feature that can describe the appearance of areas in an image. Texture can define the structure of an object in terms of "patterns" of pixel values and thus enables identification of nuances and irregularities. Haralick et al. (1973) use second order statistics for texture analysis by estimating the relationship between pairs of pixels within the image. The authors use the grey level co-occurrence matrix (GLCM) to record the number of occurrences of a specific pixel pair. The GLCM is a square matrix of size $N_g \times N_g$, where N_g is the number of grey levels (intensity levels) in the image. In this thesis a quantisation of 32 grey levels is used, i.e. $N_g = 32$. The matrix is normalised and considered as an array of probabilities of the pair of grey levels occurring at a specific position in the image.

For each block (i, j) of the intensity image, $\mathbf{I}^\theta$, of pyramidal level θ, the spatial probability of the respective pixel pair in direction of vector δ is computed for the grey levels $c = 1, \ldots, N_g$ and $d = 1, \ldots, N_g$ for the first and second pixel respectively as

$$p^\theta_{\delta ij}(c, d) = \sum_{(x,y)\in B_{ij}} \begin{cases} 1, \text{ if } \mathbf{I}^\theta(x, y) = c - 1 \text{ and } \mathbf{I}^\theta(x + \delta_x, y + \delta_y) = d - 1, \\ 0, \text{ otherwise.} \end{cases} \tag{4.25}$$

This yields the GLCM for each block (i, j) and direction δ,

$$\begin{pmatrix} p^\theta_{\delta ij}(1,1) & p^\theta_{\delta ij}(1,2) & \ldots & p^\theta_{\delta ij}(1,N_g) \\ p^\theta_{\delta ij}(2,1) & p^\theta_{\delta ij}(2,2) & \ldots & p^\theta_{\delta ij}(2,N_g) \\ \vdots & \vdots & \ddots & \vdots \\ p^\theta_{\delta ij}(N_g,1) & p^\theta_{\delta ij}(N_g,2) & \ldots & p^\theta_{\delta ij}(N_g,N_g) \end{pmatrix}. \tag{4.26}$$

Following Haralick et al. (1973), the GLCM for each block, (i, j), and pyramidal level, θ, is computed for four directions, $\delta = \left\{(-1, 0)^T, (-1, 1)^T, (0, 1)^T, (1, 1)^T\right\}$, which equals orientations of 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, and $\frac{3\pi}{4}$ respectively. A total of 14 textural features extracted from the GLCM are proposed by Haralick et al. (1973), however, only four are commonly used:

- The *Local Contrast* of a block (i, j) describes the relative grey level (intensity) difference between pixels and their neighbours in direction δ. High changes in intensity in this direction will be picked up by a high local contrast. The local contrast is

computed as

$$ {}^{C}\mathbf{C}^{\theta}_{\delta ij} = \sum_{(c,d)} (c-d)^2 p^{\theta}_{\delta ij}(c,d). \tag{4.27a} $$

- The *Homogeneity* of a block (i,j) is a similarity measure utilising the GLCM distribution. A homogeneous block is a block with very little change in intensity, i.e. the GLCM is close to being a diagonal matrix. The homogeneity feature is thus computed as

$$ {}^{H}\mathbf{C}^{\theta}_{\delta ij} = 1 - \sum_{(c,d)} \frac{p^{\theta}_{\delta ij}(c,d)}{1+(c-d)^2}. \tag{4.27b} $$

- The *Energy* of a block (i,j) is a measure of the entropy in the block and computed by estimating the spread of the distribution in the GLCM. The energy of a block is high for a constant image. The feature is therefore computed as

$$ {}^{E}\mathbf{C}^{\theta}_{\delta ij} = 1 - \sum_{(c,d)} p^{\theta}_{\delta ij}(c,d)^2. \tag{4.27c} $$

- *Correlation* within a block (i,j) shows the correlation of intensity of a pixel in direction δ compared to the reference pixel. Computing the correlation results in a value between $[-1,1]$ for maximum negative or positive correlation respectively. Therefore the absolute value is used as

$$ {}^{X}\mathbf{C}^{\theta}_{\delta ij} = \Big| \sum_{c,d} \frac{1}{\sigma_{cij}\sigma_{dij}} (c-\mu_{cij})(d-\mu_{dij}) p^{\theta}_{\delta ij}(c,d) \Big|, \tag{4.27d} $$

where

$$ \mu_{cij} = \sum_{c} c \sum_{d} p^{\theta}_{\delta ij}(c,d), \tag{4.27e} $$

$$ \mu_{dij} = \sum_{d} d \sum_{c} p^{\theta}_{\delta ij}(c,d), \tag{4.27f} $$

$$ \sigma_{cij} = \sum_{c} (c-\mu_{cij})^2 \sum_{d} p^{\theta}_{\delta ij}(c,d), \tag{4.27g} $$

$$ \sigma_{dij} = \sum_{d} (d-\mu_{dij})^2 \sum_{c} p^{\theta}_{\delta ij}(c,d). \tag{4.27h} $$

The textural feature is computed for each pyramidal level, θ as a linear combination of the local contrast, ${}^{C}\mathbf{C}^{\theta}_{\delta ij}$, homogeneity, ${}^{H}\mathbf{C}^{\theta}_{\delta ij}$, energy, ${}^{E}\mathbf{C}^{\theta}_{\delta ij}$, and correlation, ${}^{X}\mathbf{C}^{\theta}_{\delta ij}$, within

a block of the intensity image for all orientations, δ,

$$\mathbf{Z}_{ij}^{\theta} = \sum_{\delta} {}^{C}\mathbf{C}_{\delta ij}^{\theta} + {}^{H}\mathbf{C}_{\delta ij}^{\theta} + {}^{E}\mathbf{C}_{\delta ij}^{\theta} + {}^{X}\mathbf{C}_{\delta ij}^{\theta}, \tag{4.28}$$

yielding the set $\mathcal{Z} = \left\{\mathbf{Z}^{\theta} \mid \theta \in \Theta\right\}$ containing the textural features of all Gaussian pyramidal levels.

The local textural feature is computed by applying the local locality cue to the linear combination of textural features as

$$\mathbf{Y}_{T}^{L}(\mathbf{I}) := f_{ij}^{L}(\mathcal{Z}) = \bigoplus_{\theta\in\Theta} \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{Z}_{xy}^{\theta}. \tag{4.29a}$$

Utilising the global locality cue yields the global textural feature as

$$\mathbf{Y}_{T}^{G}(\mathbf{I}) := f_{ij}^{G}(\mathcal{Z}) = \bigoplus_{\theta\in\Theta} \sum_{k=1}^{M^{\theta}} \sum_{l=1}^{N^{\theta}} \left\| \sum_{\substack{(x,y)\\ \in B_{kl}}} \mathbf{Z}_{xy}^{\theta} - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{Z}_{xy}^{\theta} \right\|^{2}. \tag{4.29b}$$

The centre-surround textural feature is subsequently defined as

$$\mathbf{Y}_{T}^{S}(\mathbf{I}) := f_{ij}^{S}(\mathcal{Z}) = \bigoplus_{\theta\in\Theta} \left\| \overline{\mathbf{Z}}_{ij}^{\theta} - \sum_{\substack{(x,y)\\ \in B_{ij}}} \mathbf{Z}_{xy}^{\theta} \right\|^{2}, \tag{4.29c}$$

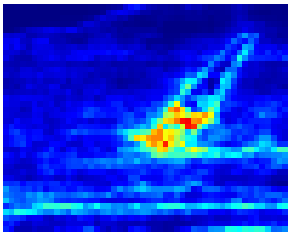
where $\overline{\mathbf{Z}}_{ij}^{\theta}$ is the mean of the textural feature of the window surrounding (i, j) of pyramidal level θ as described in Section 4.2.3 and computed following Equation (4.15), where $\overline{\mathbf{F}}^{\theta}$ is $\overline{\mathbf{Z}}^{\theta}$. The response of this detector is depicted in Figure 4.17.

The responses of the textural feature for the local, global, and centre-surround locality cues are shown in Figure 4.16.

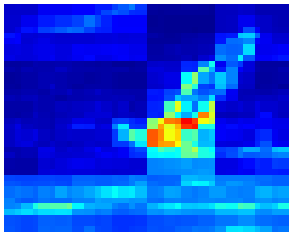
Preliminary Analysis

The response maps for the local, global, and centre-surround cues of the textural feature are shown in Figure 4.17. Good responses are shown for the test images in (a)–(f). Here, all maritime objects are detected and little noise and only few false positives are produced. The images shown in (a) and (d) have also been tested using the edge and right angle feature; see Figures 4.9(h) and 4.12(i) respectively. The former image has a large density of edges in the background, not providing enough uniqueness to the maritime object

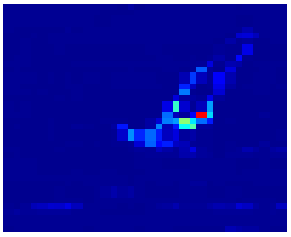
(a) Original image



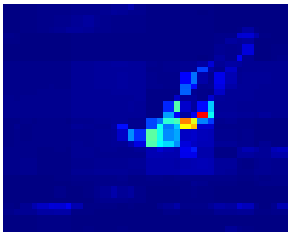
(b) Textural response



(c) Local textural feature



(d) Global textural feature



(e) Centre-surround textural feature

Figure 4.16: **Textural feature.** Evaluation of the textural feature by the local, global, and centre-surround locality cues.

when evaluating using the edge feature. On the other hand, Figure 4.12(i) shows a poor response of the centre-surround cue on the right angle feature, resulting mostly in noise. The reason being a high response of the detector at a rectangular object at the image border, dominating the response map. However, both test images yield satisfactory results when evaluated using the local textural feature as shown in Figures 4.17(a) and (d). The contrast between the maritime objects and surrounding sea and sky regions is sufficient to identify the targets in both images. This is also true for the test images in (b), (c), and (f), where target objects are identified due to piece-wise differences in contrast between each block of the image (b) or surrounding regions – (c) and (f) respectively.

The local and global locality cues are used to evaluate the same image using the textural feature in (e) and (g). While the density measurement used for the local locality cue yields a poor response, highlighting all foreground parts of the image due to the high contrast on both the target object as well as the harbour background (g), a good response comes from the global locality cue that is able to identify the actual maritime object in the image (e). This is due to the piece-wise approach of the global locality cue: the piece-wise difference reveals that due to the texture and sharp contour between the object and the background the contrast within an object region is higher than on the structured background.

For the test images in (h) and (i) the textural feature also yields a poor performance, failing to correctly highlight the maritime objects. In (h), the contrast in the area of the

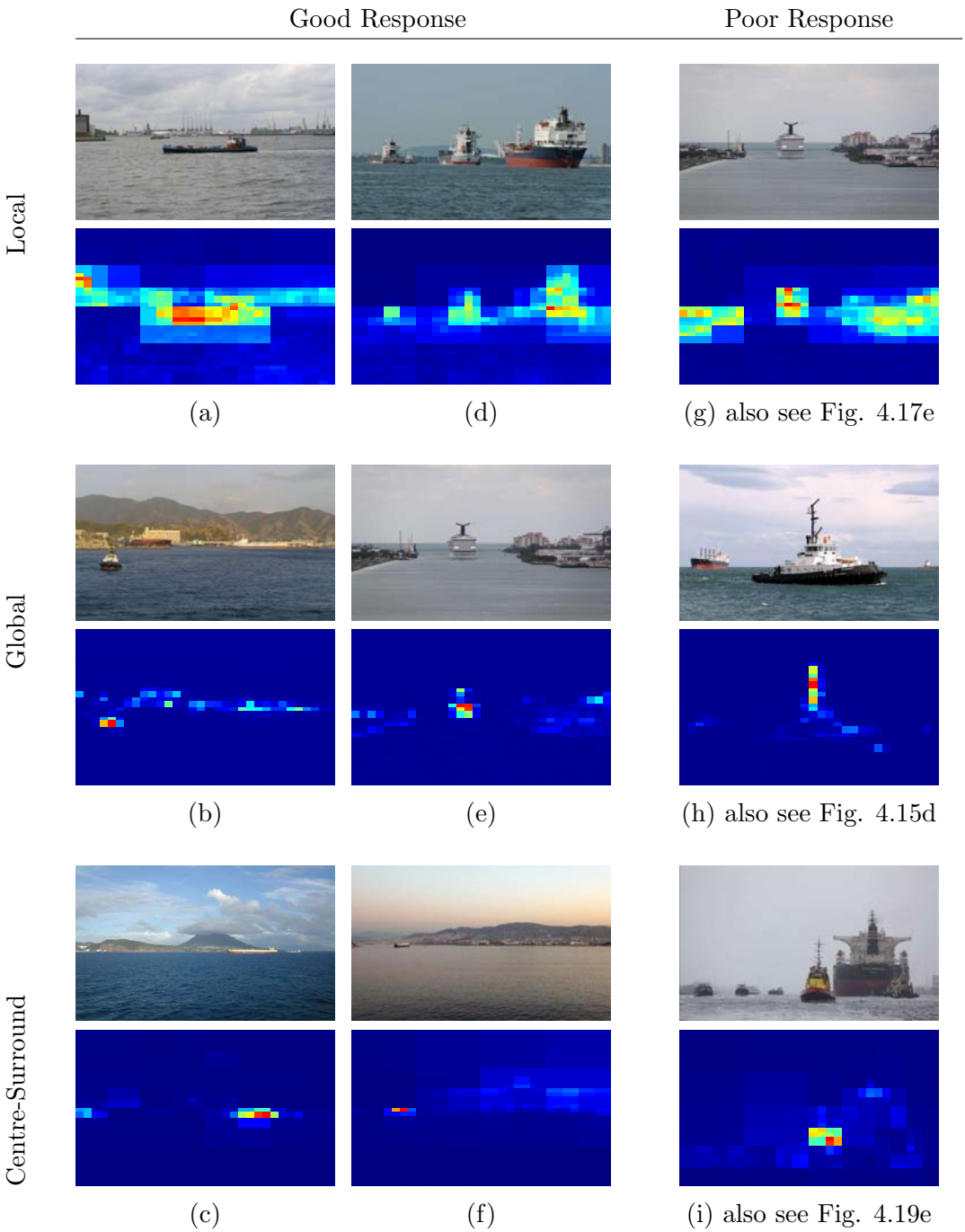


Figure 4.17: **Responses to textural feature.** The local, global, and centre-surround detector cues are used to evaluate the *textural* feature on a number of test images. The heatmaps in (a)–(f) reveal a good response to the test images with respect to maritime objects while heatmaps (g)–(i) expose a poor performance when using the textural feature. The cross references for these heatmaps refer to features yielding a good performance on the respective test images.

mast of the ship in the centre of the image is significantly higher than in the rest of the image, which causes the global cue to highlight only this part of the image as this is the most unique part of the feature. The area with the highest contrast for the test image in (i) is the part between the big ship on the right and the tug boat just to the left of it. When compared to the surrounding region that consists of mostly plain coloured areas with no abrupt changes, even more emphasis is put on this area. The images in (h) and (i) are also tested using the local frequency detector (Figure 4.15(d)) and global colour feature (Figure 4.19(e)) respectively.

4.3.4 Colour

Colour can be an effective feature to separate the object of interest out from the background (for example a bright yellow dingy surrounded by blue water). In maritime environments, colour can be ineffective when the colour contrast between the object and the surrounding is low due to low light condition or when dealing with camouflaged objects. However, not all objects are expected to be camouflaged, and indeed normal shipping vessels tend to stand out in stark contrast to the surrounding water specifically to reduce the chance of accidental collisions. Generic objects are defined not to have a specific colour and this is also true for maritime objects. The assumption of a ship being painted grey, or a buoy painted red cannot be made. However, instead of focusing on a specific colour or colour distribution of the target object, differences in colour are more likely to indicate the presence of objects. Maritime scenes sometimes consists of large areas with similar colours, e.g. sky but also large buildings or natural scenes that dominate the background. This is in fact the fundamental observation that Achanta et al. (2009) and Achanta and Süsstrunk (2010) exploit to perform saliency detection, to great effect. In this case, a target object can be identified through its difference of colour compared to the rest of the image.

Computing colour difference is preferably done in CIELAB space, where the perceptive colour difference corresponds to the Euclidean distance between the two colour vectors – see Chapter 2. This allows the use of local, global, and centre-surround locality cues without the need to standardise the channels.

Let $\mathcal{J} = \left\{ \mathbf{J}^{\theta} \mid \theta \in \Theta \right\}$ represent the set of all pyramidal levels, θ, of the image in CIELAB space, then $\mathbf{J}^{\theta}_{xy}$ is a three dimensional vector containing the L*, a*, and b* channels of pyramidal level θ at pixel (x, y). As colour does not have a density property, the distance between the image mean and the current block is used for the local locality cue. For each block of the image (i, j), the Euclidean distance between the image mean and the mean

CIELAB vector of the block is computed using the local locality cue as

$$\mathbf{Y}_C^L(\mathbf{I}) := f_{ij}^L(\mathcal{J}) = \bigoplus_{\theta\in\Theta} \Big\| \overline{\mathbf{J}}^{\theta} - \sum_{(x,y)\in B_{ij}} \frac{1}{b^2}\,\mathbf{J}_{xy}^{\theta} \Big\|, \tag{4.30a}$$

where b is the block size and $\overline{\mathbf{J}}^{\theta}$ is the CIELAB mean of the pyramidal level θ and is computed as

$$\overline{\mathbf{J}}^{\theta} = \frac{1}{h^{\theta}\cdot w^{\theta}} \sum_{x,y} \mathbf{J}_{xy}^{\theta}. \tag{4.30b}$$

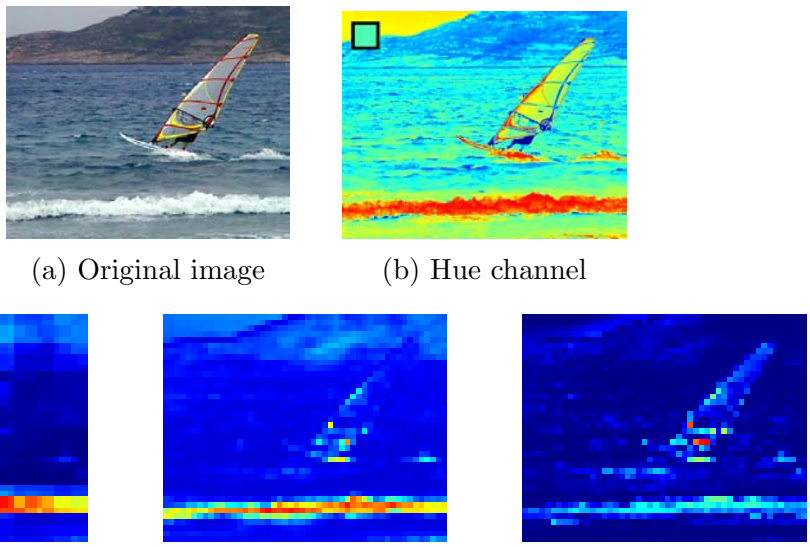
In a colour image with multiple dominant regions of the same colour and a small size target, the mean CIELAB vector of the image will be roughly half way between these colours since CIELAB is designed to model perceived distances as linear distances. If a maritime object is of approximately this colour, it will not be highlighted in the response map, as the distance between the mean colour and the target colour will be roughly zero. Moreover, both of the dominating background regions will get highlighted as there is a distance between their colour and the image mean colour. Computing the sum of the squared distances between image blocks, as suggested by the global locality cue, can identify regions of unique colour:

$$\mathbf{Y}_C^G(\mathbf{I}) := f_{ij}^G(\mathcal{J}) = \bigoplus_{\theta\in\Theta} \sum_{k=1}^{M^{\theta}} \sum_{l=1}^{N^{\theta}} \Big\| \sum_{(x,y)\in B_{kl}} \mathbf{J}_{xy}^{\theta} - \sum_{(x,y)\in B_{ij}} \mathbf{J}_{xy}^{\theta} \Big\|^2. \tag{4.31}$$

In images with large objects, objects of similar colour, or complex backgrounds, comparing against the image mean colour highlights the background as it is more unique than the actual objects. Achanta and Süsstrunk (2010) showed that using the maximum centre-surround windows as defined by the centre-surround locality cue can overcome this issue. The centre-surround colour feature is subsequently defined as

$$\mathbf{Y}_C^S(\mathbf{I}) := f_{ij}^S(\mathcal{J}) = \bigoplus_{\theta\in\Theta} \Big\| \overline{\mathbf{J}}_{ij}^{\theta} - \sum_{(x,y)\in S_{ij}} -\frac{1}{b^2}\,\mathbf{J}_{xy}^{\theta} \Big\|^2, \tag{4.32}$$

where $\overline{\mathbf{J}}_{ij}^{\theta}$ is the CIELAB mean of the window surrounding (i,j) of pyramidal level θ as described in Section 4.2.3 and computed using Equation (4.15), where $\overline{\mathbf{F}}^{\theta}$ is $\overline{\mathbf{J}}^{\theta}$. Figure 4.18 shows the responses of the colour feature for the local, global, and centre-surround cues.



(a) Original image (b) Hue channel

(c) Local colour feature (d) Global colour feature (e) Centre-surround colour feature

Figure 4.18: **Colour feature.** Evaluation of the colour feature by the local, global, and centre-surround locality cues. Image (b) shows the hue channel of the CIELAB image, the small box in the top left shows the mean hue of the image.

Preliminary Analysis

The response of the local, global, and centre-surround locality cues for the colour feature is shown in Figure 4.19. The feature yields a good response for the test images in (a)–(f) and a poor response for the images in (g)–(i) respectively.

The local locality cue is used to compute the difference of the mean colour of a block to the mean colour of the image. The mean colour of the test image in (a) is rather dark, yielding a highlighting of bright parts in the image, including the ship in the foreground. However, some false positives are also detected as they have a similar colour difference. The same is true for the images in (b) and (d), where the maritime objects are identified by the sum of the piece-wise distances of colour between the blocks, (b), or the difference in colour to the image mean (d). In (b), the global locality cue also misdetects a part of the coastal area in the image due to the high difference in colour compared with the other blocks (sea and sky) in the image. The test image in (e) has been previously evaluated using the textural feature and the response of the centre-surround locality cue is shown in Figure 4.17(i). There, the area between the two ships on the right side of the image had been highlighted as it stood out with high contrast compared to the surrounding region, while the actual maritime objects in the image have been neglected. Using the global colour locality cue, however, all maritime objects have been correctly identified, as the difference in colour between

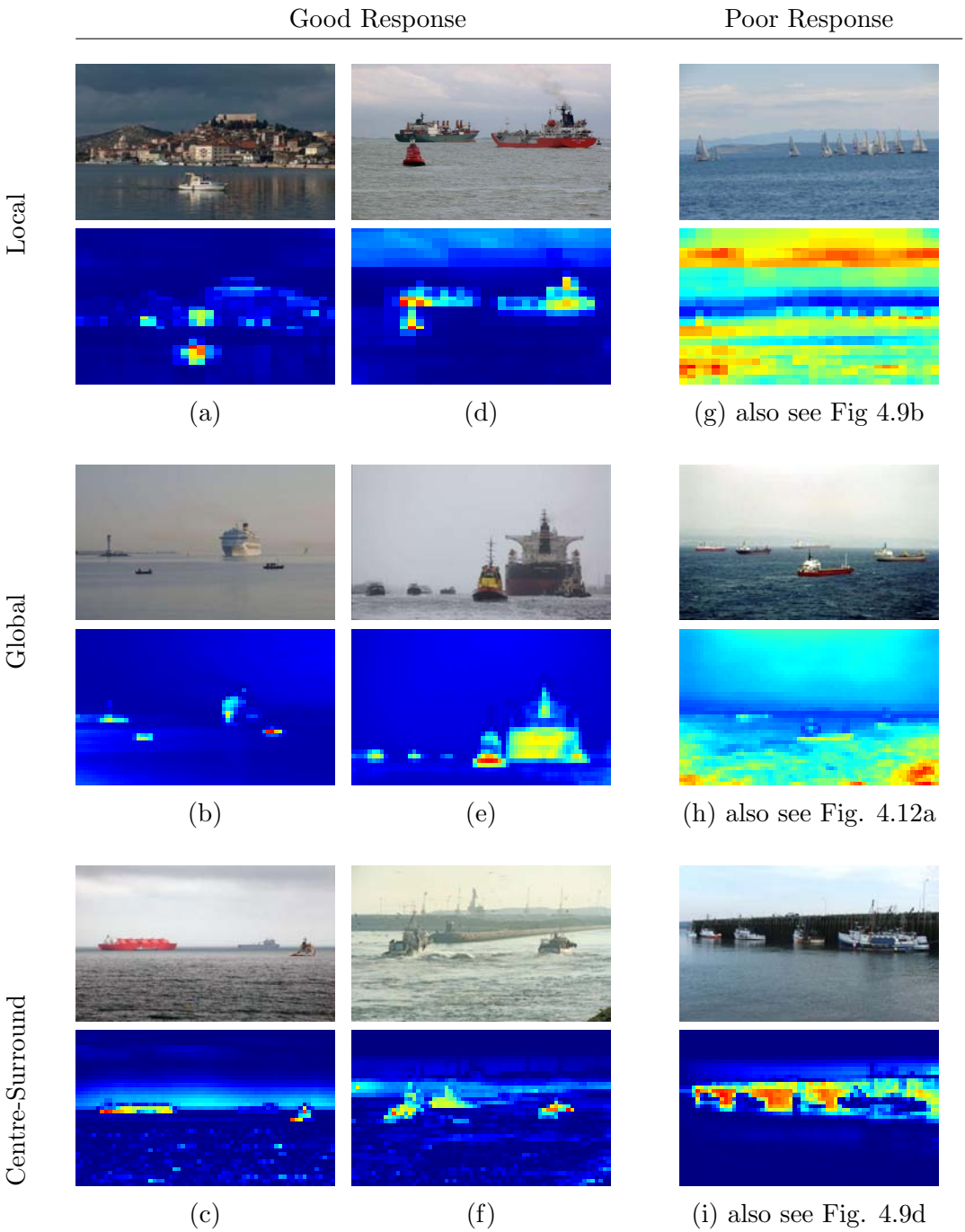


Figure 4.19: **Responses to colour feature.** The local, global, and centre-surround detector cues are used to evaluate the colour feature on a number of test images. The heatmaps in (a)–(f) reveal a good response to the test images with respect to maritime objects while heatmaps (g)–(i) expose a poor performance when using the colour feature. The cross references for these heatmaps refer to features or detector cues yielding a good performance on the respective test images.

the blocks containing the target objects and the background was sufficient. Beneficial for this was that the two big ships on the right had a different colour as the piece-wise comparison of image blocks favours blocks that are highly distinctive compared to others. Thus, weighting on both ships was increased as they not only differ from the global colour but also from each other.

The test image depicted in Figure 4.19(c) shows a good detection performance for the two ships in the foreground due to the colour difference to their surrounding area. When evaluated with the local frequency locality cue the waves in the foreground were highlighted, yielding a poor performance of the feature detector. The test image in (f) contains two maritime objects that do not differ much in colour from the ships. However, as their surrounding areas are mostly water that is distinctive in colour, the centre-surround locality cue is able to highlight the objects.

A poor response of the detector is shown for the test images in (g)–(i). In (g) the colour of the sailing ships is closer to the image mean than the actual background, meaning the detector highlights the background. The ships are detected using the global edge feature in Figure 4.9(b) as they have a higher edge density than the rest of the image.

Even though the ships in Figure 4.19(h) seem to be distinctive in colour to a human observer, the colour feature using the global locality cue has very poor performance, failing to highlight the presence of the ships in the image. As the global cue computes the difference in colour between an image block and the entire image, areas that have a high difference are highlighted. However, in this image, the colour of the sky and sea are very different, resulting in a mean that lies somewhere in between. This yields to a high distinctiveness of the entire image with respect to the image mean and results in a high response for the entire image. However, as one can see in Figure 4.12(a), the ships can be detected by evaluating the density of right angles present in the image.

The centre-surround cue used to evaluate the test image in Figure 4.19(i) fails to identify the moored ships and highlights the pier instead. This is due to the high difference of the dark pier compared to the surrounding areas, which are dominated by bright sky above and the ships right on the pier. The ships are detected by the edge feature in Figure 4.9(d) due to the higher density of edges on the target objects.

4.4 Classification

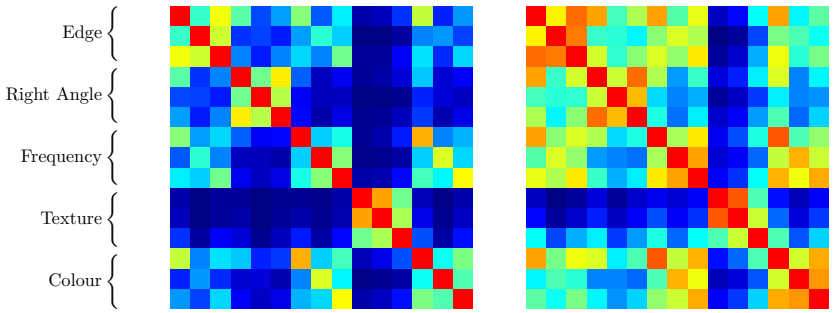
Predicting if a block of an image contains a maritime object is a problem of binary classification: a classifier is employed to compute the probability of a block containing either an object or background. A Naïve Bayes classifier has been selected for this task because it allows for probabilistic inputs; see Section 2.6 for a detailed discussion. A necessary and sufficient condition for the use of Naïve Bayes is that the input variables (features) must be conditionally independent given the class, a requirement that can be verified by assessing the correlation between features.

The correlation between two variables, A and B, can be computed as

$$\rho(A, B) = \frac{E[(A - \mu_A) \cdot (B - \mu_B)]}{\sigma_A \sigma_B}, \tag{4.33}$$

where $E[\cdot]$ is the expected value and σ is the standard deviation of the distribution.

The correlation matrix, ρ, for a set of variables, $Y = \{Y_1, Y_2, \ldots, Y_n\}$, is a $n \times n$ symmetric matrix where the matrix entries, ρ_{ij}, are the result of $\rho(Y_i, Y_j)$. Figure 4.20 shows correlation matrices between all previously introduced low-level features and locality cues as input variables on two datasets – the datasets will be formally introduced in Section 4.5.1 and 4.5.2 later in this chapter. The matrices show the correlation between the entire



(a) *MSRA* (b) *shipspotting*

Figure 4.20: Correlation of all features evaluated on the *MSRA* and *shipspotting* dataset. For each of the five low-level features, the local, global, and centre-surround locality cues are shown (top to bottom).

set of features and locality cues, $\mathbf{Y} = \{\mathbf{Y}_E^L, \mathbf{Y}_E^G, \mathbf{Y}_E^S, \mathbf{Y}_R^L, \mathbf{Y}_R^G, \ldots, \mathbf{Y}_C^S\}$, in a heatmap representation.

For the first dataset (Figure 4.20(a)), almost no correlation is observed between the features, even within the locality cue variants of a single feature. The second dataset (Fig-

ure 4.20(b)) also has little correlation between different features, although it exhibits some correlation between locality cues of the same feature. Although this implies some dependence, the effect of assuming independence is not severe – at worst the Naïve Bayes classifier will merely underperform since the dependencies were not considered. Furthermore, the correlations are mainly between localities of the *same* feature and these dependencies are not actually informative for classification purposes. Another finding of the correlation analysis is that the textural feature (with any locality cue) is the most unique feature in the set. See Section 5.2 for a discussion about the importance and influence of different features and locality cues.

Based on the findings, all low-level features and locality cues are combined using the Naïve Bayes approach. The resulting network is depicted in Figure 4.21. The input features are separately normalised to $\mathbf{Y} \rightarrow 0 \ldots 1$ and treated as probability maps. Training the Naïve Bayes classifier is, as discussed in Chapter 2, a matter of counting the occurrence of each feature given the known ground-truth class and normalising the resultant histogram of feature values to produce a set of conditional probability tables (one per feature) as well as the prior probability table of the classes. These probability tables form the parameters of the Naïve Bayes classifier. Then when a test image is provided, features are extracted and for each image block the probability of a maritime object, $P\left(X = object \,|\, Y_E^L, \ldots, Y_C^S\right)$, is calculated using the learned parameters. If this probability exceeds a given threshold then the block is classified as a maritime object – the threshold is varied to produce a precision/recall curve to analyse the sensitivity and performance of the system.

The Bayesian classifier is trained and evaluated for each dataset using a 10 fold cross-validation on the respective datasets.

4.5 Experiments

The proposed approach for visual attention has been evaluated on two different datasets and compared against current state of the art saliency detectors. This section names the work to which the proposed approach is compared to, introduces the datasets that are used for evaluation, and gives details about the experiments conducted. The section closes with a detailed discussion of the experimental results.

The proposed approach is compared to the following existing work:

- Achanta and Süsstrunk (2010) because it is amongst the most recent saliency detec-

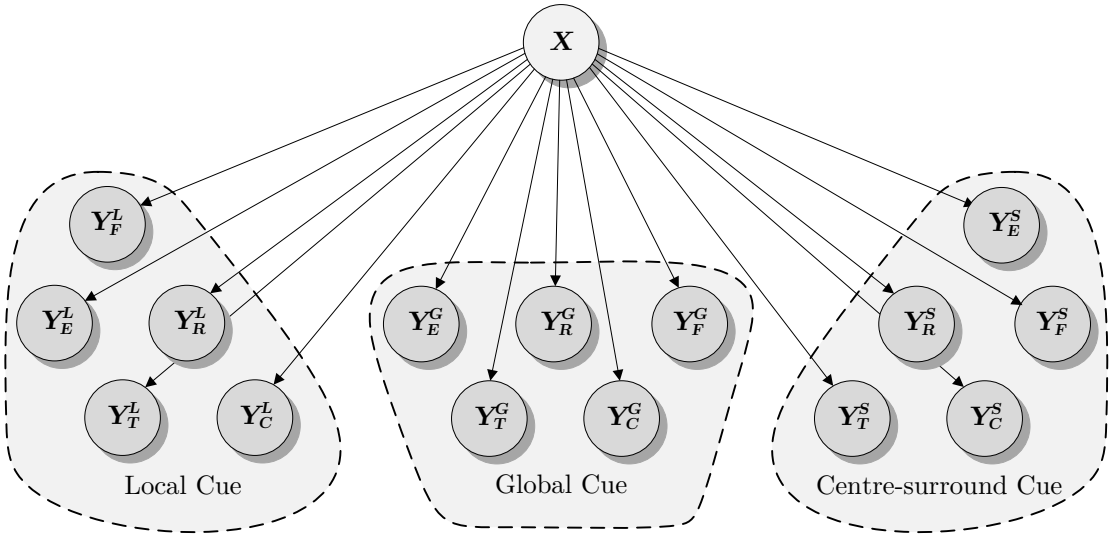


Figure 4.21: Bayesian network of the classifier.

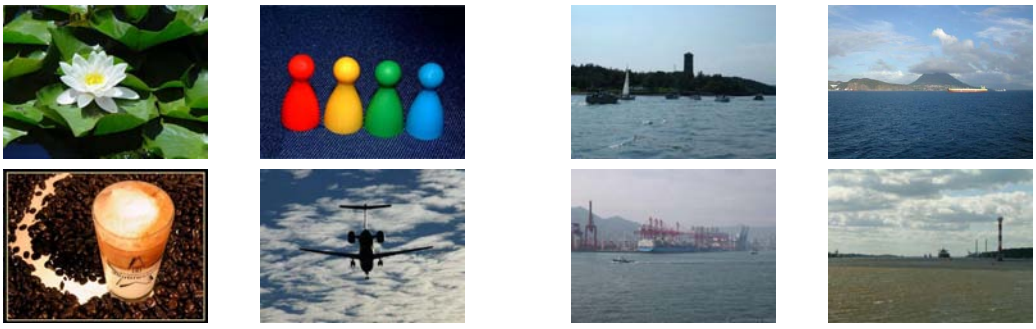
tors and has been shown to be highly effective. The authors demonstrated that their proposed method outperforms the works of Itti et al. (1998), Harel et al. (2007), and Hou and Zhang (2007).

- Rosin (2009) due to its simple and parameterless approach which outperforms Itti et al. (1998) and Ma and Zhang (2003), and can keep up with Liu et al. (2007). Although Rosin recommends performing erosion to reduce the overfitting produced by the algorithm it is evaluated based on the raw results to avoid introducing an additional parameter that must be optimised. In any event, Rosin showed that such an erosion would only improve performance by less than 10%.
- Alexe et al. (2010) because their *objectness* measure can be used to approach the problem of visual attention in a unconventional way. The authors showed that their approach outperforms Itti et al. (1998) and Hou and Zhang (2007).

All of the above mentioned are discussed in detail in Chapter 2.

Experiments for the maritime visual attention framework are performed on two different datasets (Figure 4.22). The *Salient Object Database (MSRA)* is the community standard test set, consisting of a variety of object classes and backgrounds. The proposed approach is tested against this dataset to evaluate its performance for the detection of generic objects. Additionally, experiments on domain specific imagery are desirable. However, a dataset with a focus on maritime scenes was not publicly available. Therefore a test set consisting of real-world maritime imagery has been compiled and published as the

shipspotting dataset (Albrecht et al., 2011). The proposed approach for maritime visual attention is compared to the above mentioned approaches on both datasets. For this, each image is evaluated according to the classification criterion introduced in Section 2.8 and the results are shown in precision/recall plots.



(a) *MSRA* (b) shipspotting

Figure 4.22: Sample images from *MSRA* and *shipspotting* datasets.

4.5.1 MSRA – Salient Object Database

The *Salient Object Database* (Liu et al., 2007), referred to as *MSRA* in the remainder of this thesis, is a generic object data set consisting of a total of 25 000 images. In each image, the dominant salient object has been annotated using a bounding box. Achanta et al. (2009) took the tremendous effort to annotate a subset of 1 000 images at a pixel level, outlining the shape of the salient object in each image. This thesis agrees with the statement of those authors that comparing the shape of the object instead of a bounding box is more realistic and allows for a better evaluation of classification accuracy. Therefore, their subset is used for evaluation in this thesis.

4.5.1.1 Results and Discussion

The precision/recall plot shown in Figure 4.23 indicates that Achanta and Süsstrunk (2010) and Alexe et al. (2010) outperform Rosin (2009) and the proposed approach on *MSRA*. This is not altogether surprising since the dataset contains general images of both human and natural scenes rather than specifically maritime images that the features were selected for. A clear indication about which of the former two is the best algorithm for this dataset cannot be given as the curves intersect and thus weighting towards precision or recall is dependent on the field of application. Figure 4.24 shows the response to a number of

sample images from *MSRA* evaluated by the detectors of Alexe et al. (2010), Achanta and Süsstrunk (2010), and Rosin (2009) compared to ground truth and the proposed approach.

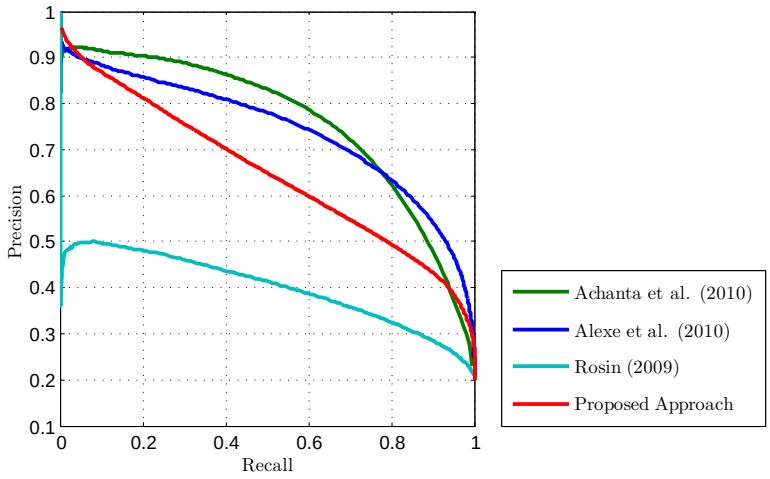


Figure 4.23: Precision/Recall plot comparing the performance of the four evaluated algorithms on the *MSRA* dataset.

The adapted *objectness* measure as proposed by Alexe et al. (2010) provides a good indication for the presence of a salient object. In general, the centre part of each object is detected with a satisfactory performance. However, the borders of the objects are mostly feathered, not providing a sharp segmentation between object and background. This is caused by the weighted window approach – as mentioned, Alexe et al. (2010) only compute the score of a rectangular window containing an object anywhere within the window. While a large window is weighted with a high score because it contains an object, the actual object boundary is "fringy" due to the rectangular shape and the weight is uniformly distributed within the window – yielding low precision. Nevertheless, the images in Figure 4.24(a)–(c) yield good performance with the objects correctly identified. In (d), the bucket is weighted more important than the ape wearing it. This result actually is debatable because it is dependent on the definition of saliency. One could argue that the bucket, not the ape, is actually the most salient object in the image and Alexe et al. (2010) therefore detect the correct object. This again shows the ambiguity of the definition of saliency, where detection accuracy can be unintentionally affected by human interpretation in the ground truthing process. The objects in (e) and (f) are highlighted mostly in the centre of the respective objects with the extremities of the player and the pike of the building missing in the saliency maps because of their small shape. Overall, Alexe et al. (2010) provide a good recall with respect to ground truth, detecting all objects. The precision of the approach is acceptable, it is mostly limited by the inferior rectangular shape of the sampled windows.

Achanta and Süsstrunk (2010) detect the object in Figure 4.24(a) with almost perfect accuracy. Their algorithm does not get distracted by the shadows in the corners of the image due to the low frequency spatial cut-off in these areas. In (b), on the other hand, the cut-off causes the highlighting of the blue flowers in the background because they are dominant in their respective region, while the position of the actual salient object yields a comparison of the colour difference towards the entire image. However, this difference is not large enough, causing the blue flowers to dominate the saliency map. The colour difference of the bird's body in (c) picks up the saliency object by Achanta and Süsstrunk (2010), but it can be seen that the head and the wing tips do not stand out in colour by much, which causes these parts to be missed by the approach. The ape's bucket and parts of the ape's body in (d) are highlighted by Achanta and Süsstrunk (2010) but the dominant regions are the ape's arms. Interestingly, the grass areas on the left side are highlighted as well – this behaviour should be avoided with the low cut-off, but the position of the grass seems to be at an unfortunate position, such that the rocks in the top and bottom dominate the background of the window and cause the grass to be highlighted as salient. In (e), the jersey of the player has dominant colour differences on the shoulder and pants, which get highlighted. However, the player does not stand out due to the dark colours being similar to the background. This is an unavoidable drawback of a single feature approach, causing misdetection if the (only) feature detector fails. In (f), Achanta and Süsstrunk (2010) again shows an almost perfect response, where it highlights the salient object including the delicate pike on the building. The approach proposed by Achanta and Süsstrunk (2010) is purely based on perceived colour difference and emphasises this difference. The content of *MSRA*, where most salient objects stand out in colour strengthens the performance of this approach. The spatial cut-off frequency furthermore emphasises objects in the centre of the image.

The precision/recall plot of the approach proposed by Rosin (2009) suggests that his method detects most of the salient objects in *MSRA* but oversegments the objects at the same time. As a matter of fact, the object in Figure 4.24(a) is correctly detected because the strong edges of the object with respect to the background yield a good response of the detector. However, the object is overfitted – this is also the case in (b), where blossum and caulis are highlighted and overfitted. A small region in the bottom left of the image is highlighted due to the presence of text in this area resulting in a response to the edge detector. In (c), a similar response is given, where the bird is detected but oversegmented. Rosin (2009) highlights the bottom part of the ape's bucket in (d) with a high confidence due to the strong edges present at the bucket's thread and contrast towards the background. The ape itself is not highlighted due to the homogeneous texture of the coat and subsequent low edge count. However, the separating region between left arm and body, creates a peak in the saliency map due to the very strong edge towards the visible

background in this area. Finally, the images in (e) and (f) show good performance of the detector. With the edge density measurement actually outperforming Alexe et al. (2010) and Achanta and Süsstrunk (2010) that fail to highlight the extremities and torso respectively. Overall, the approach proposed by Rosin (2009) shows an acceptable performance given its simplicity. The recall performance actually outperforms all other approaches but this stands against poor precision due to oversegmentation. The author suggests addressing the oversegmentation by eroding the results with a disk structure. However, this was not performed in this comparison as it would introduce an additional parameter that has to be optimised and is difficult to justify given the expected performance increase (Rosin, 2009). Furthermore, small objects or delicate part of objects like the pike in (f) could not be detected if the map would be systematically eroded.

Based on the precision/recall plot, the proposed approach promises a similar detection quality to Rosin's edge density measure with better separation of the salient objects towards the background. On initial inspection what is most striking about the heatmaps produced by the proposed approach is that there is uniform strong response over the objects, dropping off quickly at the borders. In contrast, the compared approaches tend to have highly variable response within a single object. Achanta and Süsstrunk (2010) propose the use of graph cuts to extract the object's shape by using the spatial consistency between nearby strong and weak responses. However, this requires a subsequent higher level processing stage to usefully segment the object from the background. With the proposed approach the contrast between foreground and background is already very distinct. Interestingly, this distinctiveness is highly uniform across the object even though blocks are processed independently from their spatial neighbours. This implies that the approach is able to correctly identify visual attention at a small scale (blocks) and still provides a good representation of the object at the macro scale. The object in Figure 4.24(a) is correctly detected with the entire object uniformly highlighted. A small oversegmentation causes the object to appear coarser and larger than it actually is. The proposed visual attention approach identifies the shape of the blossom in (b) almost correctly, yet the delicate contour of the object is oversegmented. However, the proposed approach outperforms all other compared detectors on this image in both precision and recall. The bird in (c) is detected but massively oversegmented. The proposed approach fails to detect the left wingtip of the bird – facing the same issue as the other approaches. Some responses to waves are present in the produced saliency map as well, causing false positive measurements. (d) shows the saliency map of the ape and its bucket. Here, the proposed approach fails to detect the shape of the target object while the saliency map reveals that the object is massively oversegmented. Additionally, a number of false positives are detected in the background around the actual object. The player in (e) is detected with a good accuracy in both precision and recall. The proposed approach detects the entire player

with the exception of his lower leg, which is a significant improvement on the detection of the torso by Alexe et al. (2010), the detection of only parts of the player's jersey by Achanta and Süsstrunk (2010), or by the oversegmented result of Rosin (2009). Finally, the object in (f) is correctly detected and uniformly highlighted. The proposed approach shows an acceptable overall performance on *MSRA* with few misdetections in challenging images and some oversegmentation. The Bayesian classifier allows objects to be uniformly highlighted.

4.5.2 Shipspotting Dataset – Maritime Objects

The *shipspotting* Website (http://www.shipspotting.com, last accessed 2011-11-14) is a community Website of hobby photographers that are interested in ships and maritime scenes. Photos are categorised and images from the category *Harbour Overview Images* are suitable as testing images for the proposed framework. One hundred images that contain the desired scenery and represent the viewpoint of a mobile maritime platform were selected. All images were downscaled to a maximum size of 512 pixels in either width or height while preserving the aspect ratio. Annotations at pixel level were published by Albrecht et al. (2011). Here, care was taken to follow an exact definition of saliency. As the purpose of the proposed framework is the identification of areas of visual attention that are of relevance for maritime surveillance or pose a possible hazard to the platform, the following two criteria were used:

1. Object had to be of maritime nature and on the surface of the water.
2. Object is not a fixed landmark that would appear on a map or satellite image.

This, for example, excludes cranes in a cargo harbour as well as lighthouses but will include buoys or floating platforms.

4.5.2.1 Results and Discussion

The precision/recall plot shown in Figure 4.25 demonstrates that the proposed approach outperforms Achanta and Süsstrunk (2010), Alexe et al. (2010), and Rosin (2009) in both precision and recall on the *shipspotting* dataset. Compared to *MSRA*, the precision of the detector proposed by Achanta and Süsstrunk (2010) dropped drastically, while the recall performance remained constant. This is likely explained by the much lower emphasis

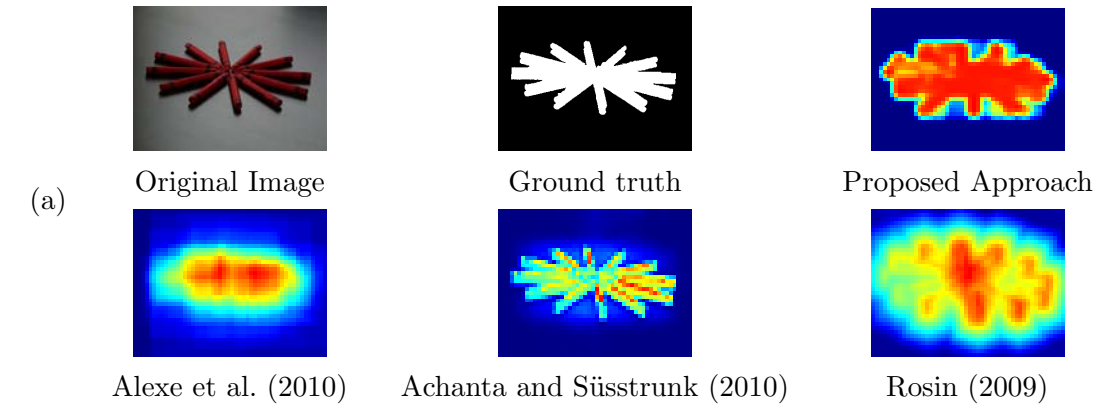
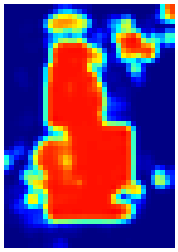


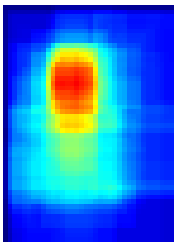
Figure 4.24: Results for *MSRA* database (continued on next page).

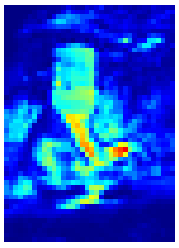
(d)

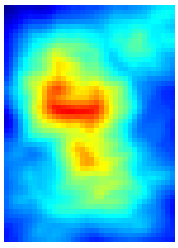

Original Image


Ground truth


Proposed Approach


Alexe et al. (2010)

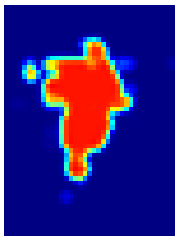

Achanta and Süsstrunk (2010)

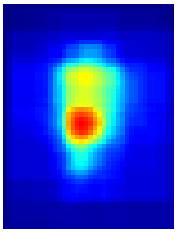

Rosin (2009)

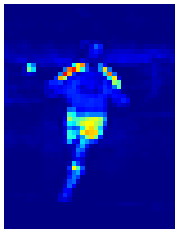
(e)

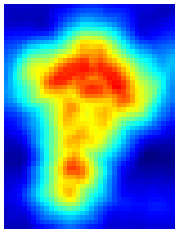

Original Image


Ground truth


Proposed Approach


Alexe et al. (2010)

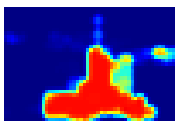

Achanta and Süsstrunk (2010)

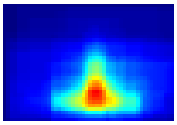

Rosin (2009)

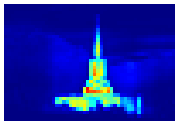
(f)


Original Image


Ground truth


Proposed Approach


Alexe et al. (2010)


Achanta and Süsstrunk (2010)

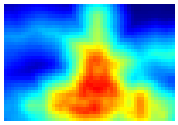

Rosin (2009)

Figure 4.24: Results for *MSRA* database (continued from previous page).

on colour as a distinguishing attribute of maritime objects. A similar change is notable for Alexe et al. (2010), where the precision declined more than recall performance. Due to the randomly sampled window approach, their method is more vulnerable to reduced object sizes that are part of the *shipspotting* dataset. The approach proposed by Rosin

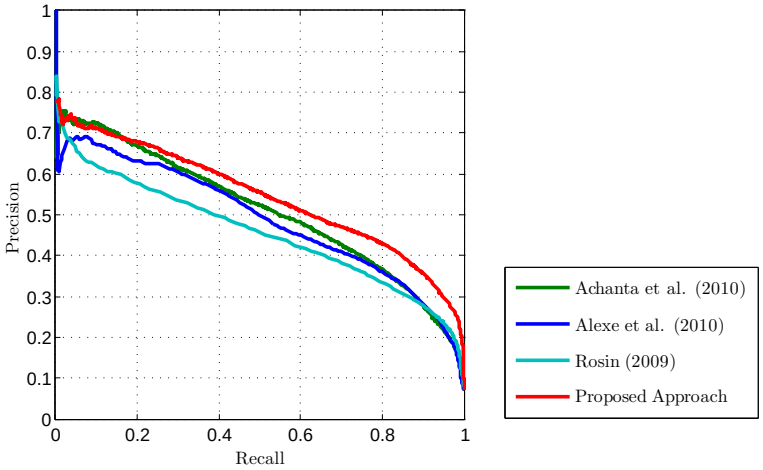


Figure 4.25: Precision/Recall plot comparing the performance of the four evaluated algorithms on the *shipspotting* dataset. The presented algorithms outperforms other algorithms in both precision and recall performance.

(2009) improved performance in terms of both precision and recall when evaluated on the *shipspotting* dataset.

Overall, the performance of all detectors except Rosin (2009) dropped. Of these three, the proposed approach handled the change of the dataset best with the smallest change in precision and recall due to the low-level features specifically designed to cater for the maritime scenes. Figure 4.26 shows the response to a number of sample images from *shipspotting*, evaluated using the approaches of Alexe et al. (2010), Achanta and Süsstrunk (2010), and Rosin (2009) compared to ground truth and the proposed approach.

The *objectness* measurement proposed by Alexe et al. (2010) results in a partial detection of the maritime construction on the test image shown in Figure 4.26(a). The resulting saliency map shows a high weighting on the centre-piece of the construction, while the stilts and truss segments are only peripherally highlighted. The detector fails to identify the second target on the left side. The image in (b) contains a number of sailing boats in front of a challenging background. Instead of separating the sailing boats, Alexe et al. (2010) highlight the entire region with an emphasis on the ships on the right side of the image due to the more complex shapes in this area. The tugboat in (c) is the most salient object in this image for their algorithm. While the big ship on the right is partially

highlighted, the three ships on the left are completely discarded by this detector. Overall, Alexe et al. (2010) are able to detect dominant objects in the dataset. Smaller objects are combined into one or missed completely. Again, precision suffers from the window approach that only provides uniformly distributed weights for each window.

Achanta and Süsstrunk (2010) are able to detect the maritime construction in Figure 4.26(a) with a good response in precision for the centre-piece, the stilts, and the truss segment. Additionally, parts of the ship on the left side are detected as well, due to colour-dissimilarity of these areas compared to the surrounding region. The minimal colour difference between the sailing boats and the background in (b) causes the detector to miss the target objects. The position of the boats in the vertical centre of the image causes the centre-surround window that is utilised to extract the surrounding region to include almost all of the grey sky. This corresponds to the colour of the sails resulting in a low colour difference and subsequent misdetection of the boats. On the other hand, the colour difference approach enables Achanta and Süsstrunk (2010) to detect almost all target objects in (c), where the tug boat in the centre is emphasised because of the big difference in colour. However, the third ship from the left is hardly visible as the surrounding window at this position includes all other ships of similar colour, resulting in a low dissimilarity. In contrast to the *objectness* measure, Achanta and Süsstrunk (2010) have no issues with small objects if they are distinctive in their respective regions. However, objects that are not, are likely to be missed either partially or completely.

The edge density based detector proposed by Rosin (2009) is able to identify all objects in the test images of Figure 4.26. In (a), the maritime structure in the centre of the image is detected and the entire ship on the left, which was missed by Alexe et al. (2010) and Achanta and Süsstrunk (2010), is detected. However, both targets are significantly overfitted, with the background partly highlighted. In (b), Rosin (2009) detects a false positive in the bottom left corner of the image. Apart from that, the saling boats are correctly identified due to their distinct edge-separation towards the background. The mostly uni-coloured background in (c) favours Rosin's approach as it causes distinct edges between the background and objects. The large ship on the right and the tugboat are highly structured causing a high edge count and subsequently dominate the resulting saliency map. However, the map is overlaid with false positives. As expected from the high recall rate, the approach proposed by Rosin (2009) is able to direct attention to all objects in the sample images. However, the objects are significantly overfitted and large regions of false positives are detected.

The proposed approach detects the maritime platform and the ship in Figure 4.26(a). While the platform is oversegmented slightly, the detected areas are uniformly highlighted,

suggesting equal importance has been given to each part of the platform and ship. In (b), the detection of the sailing ships is similar to Alexe et al. (2010) and Rosin (2009). While Alexe et al. only detect the flock of ships on the right, Rosin and the proposed approach also identify the sailing ship in the left part of the image. In comparison to Rosin, the proposed approach provides a better segmentation, uniformly highlighting the detected regions with well-defined borders. However, a high number of false positives are detected in the bottom part of the image, where sea is present, significantly affecting the precision of the detector in this image. All ships in (c) are detected by the proposed approach, even though the detector oversegments the ships on the left towards the sea and sky background significantly. Overall, the proposed approach is able to detect all maritime objects in the sample images. While some objects are segmented with good precision, others are overfitted and a number of false positives are created. This is mostly in regions with a maritime background – i.e. regions of sea, which will be addressed in the next Chapter.

4.6 Summary

The target-centric image stabilisation process presented in Chapter 3 had to be initialised manually by selecting the target object. This shortcoming has been addressed in this chapter and a visual attention framework has been proposed that can be used to automatically identify and direct visual attention to areas of interest in maritime imagery.

The presented framework provides a method to fuse various low-level features and distance measurements (locality cues) in a Bayesian network and compute the probability of visual attention in a region of the image. The Bayesian formulation allows for the fusion of multiple features such that the weaknesses of some features can be successfully offset by the strengths of others. Thus the brittleness of using a single feature (such as Achanta and Süsstrunk (2010) and Rosin (2009)) can be compensated for by fusing multiple features that complement each other.

The low-level features can be of any kind, however, the presented features have been selected with the maritime domain in mind and to respond to characteristics of maritime objects. The selected features were introduced and positive and negative responses to sample images were shown and discussed for each feature and cue.

Experimental evaluation showed that the proposed framework gives good response with respect to accurate ground truth images. The framework is tested on the community

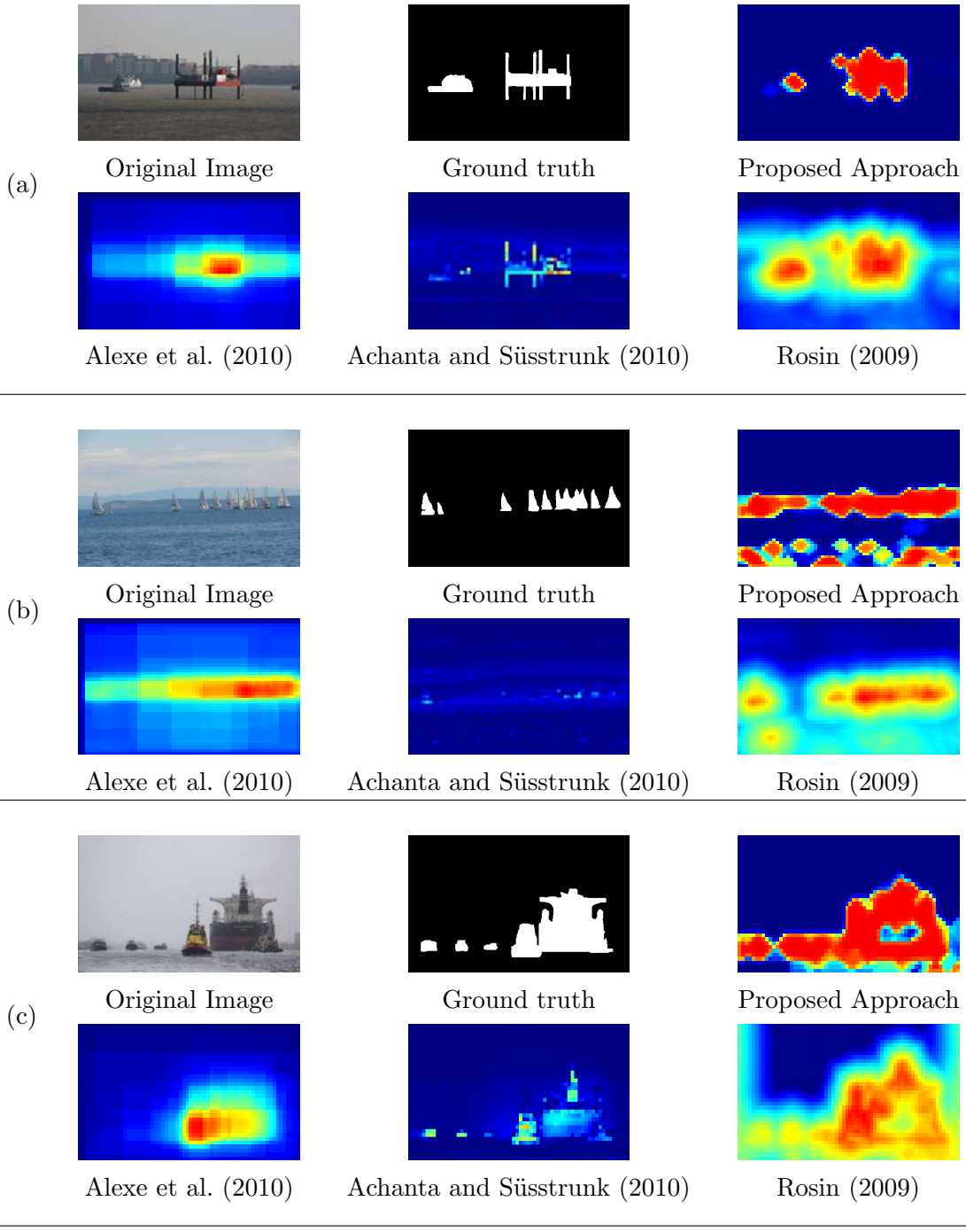


Figure 4.26: Results from *shipspotting* database.

standard *MSRA* dataset as well as on a domain specific dataset consisting of maritime imagery. A comparison with three different state-of-the-art detectors showed that the approach outperforms existing techniques in the described environment.

Chapter 5

Segmentation and Feature Selection for Maritime Visual Attention

The previous chapter introduced a framework that allowed fusion of domain specific low-level features using three different locality cues. The purpose of the framework is to be able to identify potential areas of interest in maritime imagery. It showed acceptable performance on a standard dataset and outperformed state-of-the-art detectors on domain specific imagery. Initial manual evaluation of the features ensured that they responded well to maritime objects.

This chapter seeks to further improve upon the performance of the proposed framework by introducing features extracted from more sophisticated, classifier-based detectors. An assistive technique that is often employed to improve classification performance is the prior segmentation into foreground and background. This chapter addresses a domain specific background segmentation method for maritime applications via classification of sea, sky and "other". Although the actual recognition of the low-level characteristics is essential to the detection of maritime objects, *a prior* segmentation into potential target and non-target regions helps reduce the search space, limits the number of false positives detected and thus increases classification accuracy. Furthermore, a feature selection process is integrated into the visual attention framework that allows concentrating on the most relevant features before fusion and therefore reduce computational cost for analysing irrelevant features.

The chapter is organised as follows: a maritime specific background segmentation method that uses colour and gradient orientation is proposed and evaluated in Section 5.1. The features proposed in the previous chapter together with the background segmentation are run through a feature selection process as described in Section 5.2 and eventually combined using a Bayesian Network. Experiments are performed and the proposed framework is quantitatively evaluated in Section 5.3. The chapter concludes with a summary in

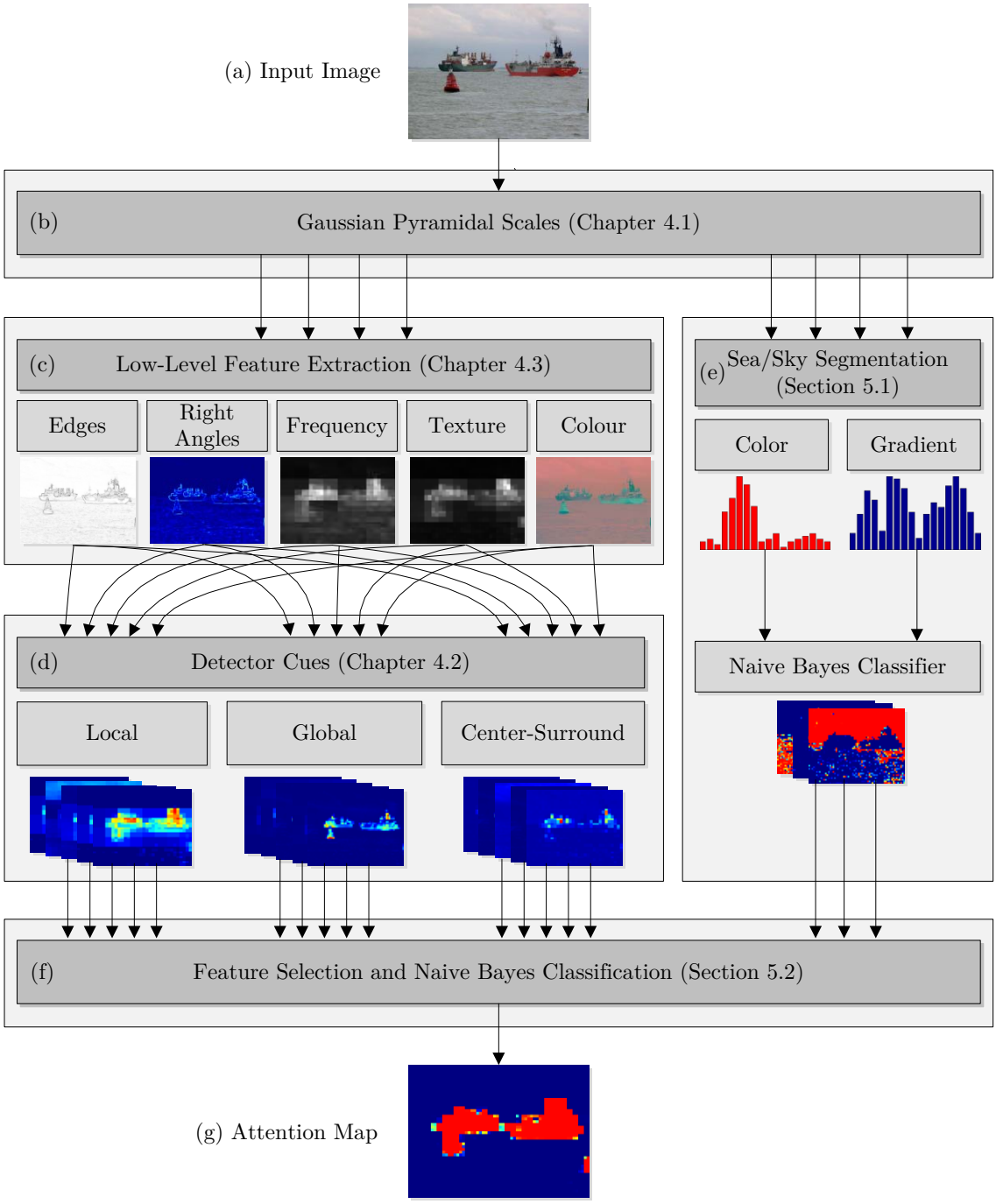


Figure 5.1: **Maritime Visual Attention Framework.** From the input image (a), a number of pyramidal scales are created (b). Low-level features are then extracted from every scale (c) and evaluated using three different locality cues resulting in a probability map per scale, feature, and locality cue (d). Simultaneously a sky/sea segmentation based on colour and orientation of gradients of the input image is performed (e), also resulting in probability maps for each of the classes. All maps are then combined using a Naïve Bayes approach (f), resulting in the final attention map (g).

Section 5.4.

5.1 Maritime Background Segmentation

Image segmentation is a technique that is used in computer vision to partition an image into regions with similar appearance or context. Compared to the pixel-wise representation of the image, the segmented representation allows a more abstract description of the image content (Felzenszwalb and Huttenlocher, 2004; Zhang et al., 2008). While the previous chapter introduced measurements that described the appearance of maritime objects based on low-level features (pixel level), this section proposes a segmentation of the image into potential target and non-target regions (foreground and background respectively) based on *a-priori* scene knowledge. In maritime scenarios, the two dominant non-target regions are *sea* and *sky*. The desired segmentation into the two non-target background regions *sea* and *sky* as well as the potential target *foreground* region is shown in Figure 5.2. This section

(a) Input image (b) Desired segmentation

Figure 5.2: Sample image and desired segmentation into regions of *sky* (red), *sea* (green), and *foreground* (blue).

proposes the use of a Bayes classifier that is used to detect the two classes of background and segments the image accordingly. The classifier is trained once on a learning set and the learned parameters are saved so that the classifier can be integrated into the proposed visual attention framework without the need for retraining.

A number of sample tiles depicting regions of *sea*, *sky*, and random *foreground* are depicted in Figure 5.3. The tiles were extracted from images from the *PASCAL Visual Object Classes (VOC) Challenge* (Everingham et al., 2010). This thesis argues that the colours of both *sea* and *sky* are distinctive within the image and can thus be used for colour based classification. Furthermore, the image gradient of waves, even though they are randomly occurring and of arbitrary shape, have dominant directions. Based on these assumptions, a descriptor is created that consists of a histogram of colour and orientation of the local gradient. While assumptions can be made for the background in maritime

(a) *Sky* (b) *Sea* (c) *Foreground*

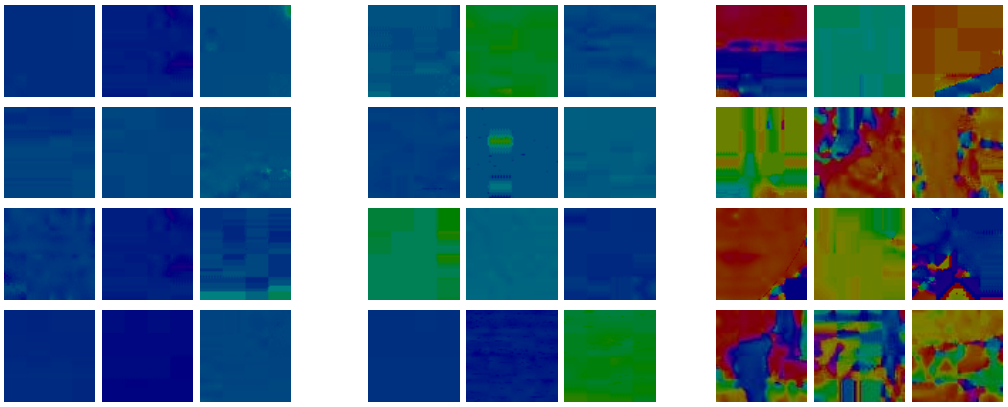
Figure 5.3: Sample images of classes *sky*, *sea*, and *foreground*.

scenes, this is not true for foreground. The purpose of the visual attention framework is to detect any maritime object and a detailed description would exclude objects with unknown appearance. The pursued approach is therefore to find an acceptable description of background and detect potential target regions (*foreground*) as everything which is not background. In the following, the colour attributes of *background* regions containing portions of *sea* and *sky* are described and the unique shape characteristics of the classes are explored. Both properties are combined in a descriptor that is computed for each block of the image. A Bayes classifier is then utilised to compute probability maps for each of the three classes. The maps are eventually fed into the maritime visual attention framework as additional feature cues for feature selection and final classification as depicted in the framework overview in Figure 5.1.

5.1.1 Colour

Amongst others, the colour feature was used in the previous section to identify potential regions of interest by computing the Euclidean distance in CIELAB space and unique regions by the perceived difference in colour. While the CIELAB model is ideally suited for computing colour differences as the outcome of the Euclidean distance is scalar, describing a specific colour requires at least two channels – plus an additional channel for luminance, if desired. As established in Section 2.7, the HSV colour model also uses two channels to encode colour information – plus an additional channel for luminance, if desired. One channel (hue) is used to hold the base colour, while another is used to encode the relative brightness of the primary colour (saturation). HSV is a cylindrical colour model with hue represented as the phase around the vertical axis ranging from red (0°) through green (120°), blue (240°), and back to red (360°). As will be demonstrated, the assumption that

sea and *sky* independently consist of nuances of a unique base colour can be made. Evaluating only the phase of the hue-channel and neglecting saturation allows the description of the colour of *sea* and *sky* to be based purely on their base colour, allows tolerance for nuances, and reduces the dimensionality of the descriptor.



(a) *Sky* (b) *Sea* (c) *Foreground*

Figure 5.4: Hue channel of images of classes *Sky*, *Foreground*, and *Sea*.

5.1.1.1 Colour of Sky

The perceived colour of the sky during daylight depends on the solar spectrum and the wavelength dependence of the scattering (Bohren and Fraser, 1985; Smith, 2005). While the sun emits a wide spectrum of radiation, the earth's atmosphere functions as a filter and absorbs much of it. Yet, the atmosphere is not homogeneous and absorbs different wavelengths differently. Furthermore the position of the sun relative to the horizon determines the distance the rays have to travel through the atmosphere and how much they get absorbed. This is especially critical during dusk and dawn. The time of operation for the surveillance platform, for which the vision system is designed, is during daylight time; it is thus valid to assume the sky to appear in nuances of blue.

5.1.1.2 Colour of Sea

The absorption spectrum of water has a minimum at 410nm (violet–blue) and peaks above 700nm (red) (Braun and Smirnov, 1993; Pope and Fry, 1997). While water in small quantities, e.g. in a bottle or glass, is not substantially affected and appears to be clear,

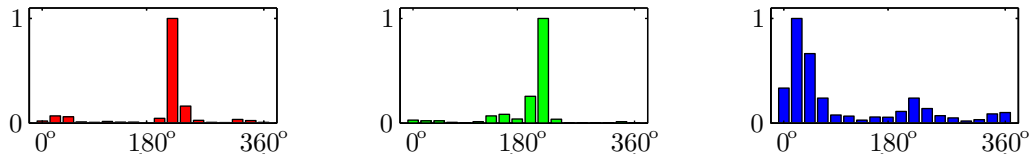
water in larger quantities, i.e. ocean, appears to be of blue colour. This again justifies the assumption that a primary colour can be used to identify areas of *sea* in images.

5.1.1.3 Colour of Foreground

Foreground as the potential target region, on the other hand, is a class that contains everything that is not of the aforementioned background classes. Therefore, the primary colour of random images not containing any parts of *sea* or *sky* must be evaluated as a negative class. This approach – to select a large set of images that represent a negative class – is quite commonly used, such as in highly successful face detection algorithm of Viola and Jones (2001). The idea is to choose a large variety of non-sea and non-sky images (or sub-images) that will effectively "map out" the space of images that are *not* sea or sky. Hence the set is not limited to maritime objects but contains images of any type of scene, object or part thereof.

5.1.1.4 Analysis

Figure 5.4 shows the hue channel for the sample images of each class: *sea*, *sky*, and *foreground*. The corresponding phase histogram is depicted in Figure 5.5. The histogram, denoted as $\boldsymbol{d}_{hue}$, is calculated for the sample images with 20° separation, resulting in 18 bins for the *hue*-channel. As can be seen from (a) and (b), the classes *sky* and *sea* have a dominant phase of the hue channel around 200°–240°, which corresponds to a primary colour of violet–blue, as expected. While a variety of colour components are present in class *foreground*, a dominance between approximately 0°–90° can be observed. The reason for this is that the majority of images from the class *foreground* contain natural scenes, for which Párraga et al. (1998) found that their spectrum has a dominance for wavelengths between $500 - 600nm$, which corresponds to a range between red and green or a phase of approximately $0° - 120°$ in the hue channel.



(a) *Sky* (b) *Sea* (c) *Foreground*

Figure 5.5: Hue histogram, $\boldsymbol{d}_{hue}$, visualising the distribution of the primary colour for classes *sky*, *sea*, and *foreground*.

5.1.2 Gradient

The edge feature described in the previous chapter makes use of the Canny edge detector to obtain the edge image from the intensity image. The orientation is of no interest for the edge feature, and the presence of edges is used only to identity possible target objects within the image. However, as described in Section 4.3.1, waves in an image respond well to an edge detector, as can be seen on a number of images (e.g. Figures 4.5(b), 4.6(b), 4.7(e), or 4.10(b)). What was described as noise and an unwanted characteristic previously, will be investigated as a possible feature of the background in this section.

The local image gradient is computed from the intensity image, $\mathbf{I}$, using the Sobel operator following Equation (4.18a) and (4.18b):

$$\mathbf{G}_x = (1, 2, 1)^T * \big((1, 0, -1) * \mathbf{I}\big) \tag{5.1a}$$

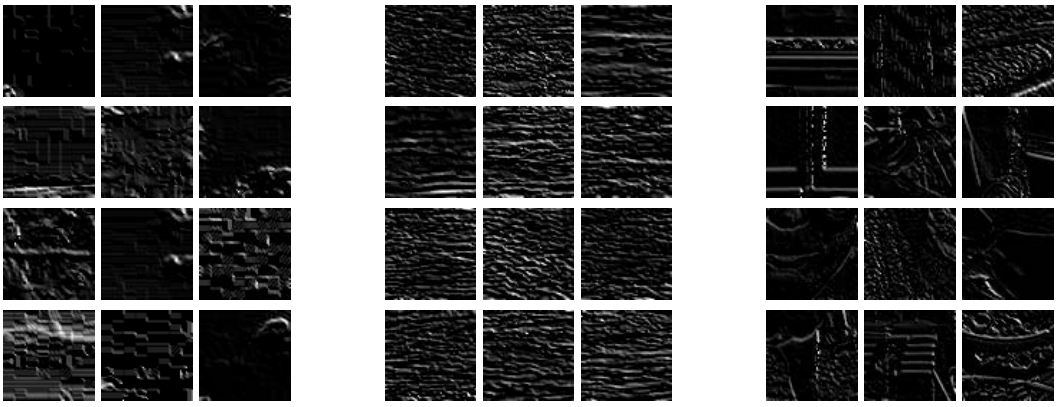
and

$$\mathbf{G}_y = (1, 0, -1)^T * \big((1, 2, 1) * \mathbf{I}\big). \tag{5.1b}$$

A map of the local gradient is then created by computing the orientation at pixel-level as

$$\phi_{xy} = \text{atan2}\left(\mathbf{G}_y, \mathbf{G}_x\right). \tag{5.1c}$$

The atan2 operator is used instead of $\tan^{-1}$ to compute the orientation because it maps to a full circle, $\phi_{xy} \to 0° \ldots 360°$, instead of $\phi_{xy} \to -90° \cdots + 90°$.



(a) *Sky* (b) *Sea* (c) *Foreground*

Figure 5.6: For the sample images from Figure 5.3, the weighted gradient is shown as (a)–(c) for classes *Sky*, *Foreground*, and *Sea* respectively.

Figure 5.6 shows the gradient for the sample images of each class: *sea*, *sky*, and *foreground*.

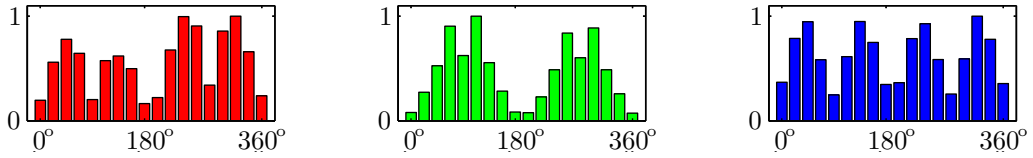
A detailed investigation was performed using a weighted vote histogram of orientations. For this, each sample is weighted by the magnitude of the local gradient as suggested by Lowe (2004) and Dalal and Triggs (2005) before being added to the appropriate bin of the histogram. This way, a gradient with a higher intensity is weighted higher than a gradient with a lower intensity. The magnitude is computed as

$$\psi_{xy} = \sqrt{(\mathbf{G}_x)^2 + (\mathbf{G}_y)^2}. \tag{5.2}$$

Lowe (2004) used 36 bins with a 10° separation to describe local features. However as the purpose here is not to create an identifying descriptor but to identify dominating orientations, a separation of 20° is used to allow some variation. This also improves the compactness of the descriptor size. The resulting weighted histogram is denoted as $\boldsymbol{d}_{hog}$.

5.1.2.1 Analysis

Figure 5.7 depicts the weighted histograms of gradients for the sample images of the three classes. For class *sea*, peaks at 90° and 270° (with some variation) are observable. This corresponds to a dominance of the gradient in the horizontal direction. The histogram



(a) *Sky* (b) *Sea* (c) *Foreground*

Figure 5.7: Gradient histogram, $\boldsymbol{d}_{hog}$

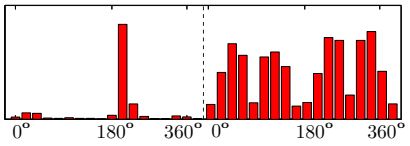
for *foreground* has four well defined peaks representing horizontal and vertical edges in the image. On closer inspection, it becomes clear that these directions are caused by the shape of the objects. This is in accordance with the findings of the right angle detector described in Section 4.3.1, which is based on the assumption that man-made objects have a tendency for vertical and horizontal edges. With *sky* the shape of the histogram is less clear cut and in fact does not follow a meaningful pattern. Note that the 180°-ambiguity is observable on the gradient histograms for classes *sea* and *foreground* (Figure 5.7(b) and (c)) while for class *sky*, the gradient histogram has a slightly higher magnitude for 180°...360° than for 0°...180° (Figure 5.7(a)) due to the subtile gradient of the sky. It is therefore important to utilise the full histogram for the descriptor.

5.1.3 Descriptor and Classification

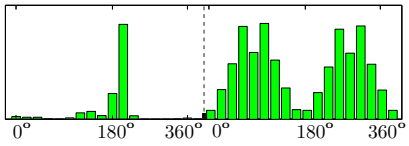
The hue histogram, $\boldsymbol{d}_{hue}$, and the weighted histogram of gradients, $\boldsymbol{d}_{hog}$, are separately normalised such that $\boldsymbol{d}_{hue} \to 0 \ldots 1$ and $\boldsymbol{d}_{hog} \to 0 \ldots 1$. They are then combined in a single descriptor with $2 \cdot 18 = 36$ dimensions as shown in Figure 5.8.

$$\boldsymbol{d} = \left(\frac{\boldsymbol{d}_{hue}}{\max \boldsymbol{d}_{hue}}, \frac{\boldsymbol{d}_{hog}}{\max \boldsymbol{d}_{hog}} \right). \tag{5.3}$$

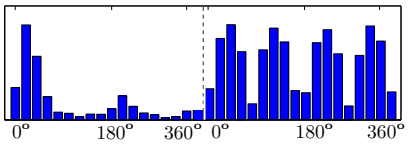
Images containing scenes of *sea* and *sky* as well as random images as negative training samples were manually extracted from the *PASCAL Visual Object Classes (VOC) Challenge* (Everingham et al., 2010). Images from the VOC dataset were chosen as a learning set so that the proposed target segmentation was trained with no relation to the global test data. The classes were balanced in terms of having equal numbers of samples for each class to avoid any bias and only parts of the images that contained scenes relevant to the classes were used.



(a) Class *sky*



(b) Class *sea*



(c) Class *foreground*

Figure 5.8: Descriptors for classes *sky*, *sea*, and *foreground*. The descriptors consist of the hue histogram (left) and orientation histogram (right).

Descriptors were then computed on a block basis for each image, resulting in a total of approximately 16,000 instances per class. To be consistent with the features introduced in the previous chapter, a block size of 8×8 was chosen. A correlation analysis (see also Section 4.4) of the descriptors computed for the test dataset (Figure 5.9) shows no obvious correlation between the dimensions of the colour descriptor, $\boldsymbol{d}_{hue}$. The descriptor based on the orientation of gradients, $\boldsymbol{d}_{hog}$, shows a minor widening of the main diagonal due to overlaps in adjacent orientations; as well as a diamond shaped correlation covering the bottom right 50% of the diagram which is due to the 180° phase equality. However, it is not significant enough to justify discarding these dimensions. Independence of dimensions

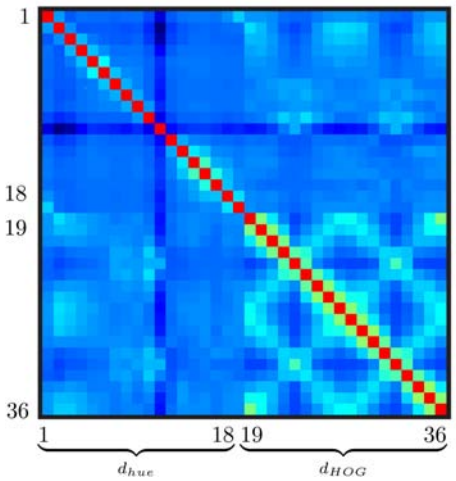


Figure 5.9: Correlation of the descriptor, $\boldsymbol{d} = (\boldsymbol{d}_{hue}, \boldsymbol{d}_{hog})$.

is advantageous because it allows the use of a Naïve Bayes approach for classification (see Section 2.6).

The structure of the Bayesian network used for the Naïve Bayes approach is depicted in Figure 5.10. Training and evaluation of the classifier is similar to that described in Section 4.4.

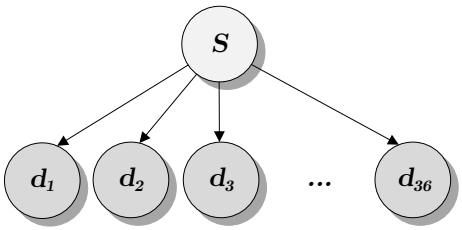


Figure 5.10: Baysian network of the sea/sky classifier.

Later in this chapter, the results of the sea/sky-classifier are fed into a feature selection process of the visual attention framework. Henceforth, the probabilities will be denoted as $\mathbf{Y}^{sky}$, $\mathbf{Y}^{sea}$, and $\mathbf{Y}^{fg}$:

$$\mathbf{Y}^{sea} = P(S{=}sea \mid d_1, d_2, \ldots, d_{36}) \tag{5.4a}$$

$$\mathbf{Y}^{sky} = P(S{=}sky \mid d_1, d_2, \ldots, d_{36}) \tag{5.4b}$$

$$\mathbf{Y}^{fg} = P(S{=}fg \mid d_1, d_2, \ldots, d_{36}) \tag{5.4c}$$

	Learning Set			*shipspotting*		
	Sea	*Sky*	*Fg*	*Sea*	*Sky*	*Fg*
Precision	0.906	0.780	0.842	0.852	0.738	0.691
Recall	0.806	0.923	0.786	0.551	0.964	0.620
F_1-Score	0.853	0.846	0.813	0.670	0.837	0.653
F_2-Score	0.824	0.890	0.797	0.593	0.908	0.633

Table 5.1: Performance of the sea/sky classifier.

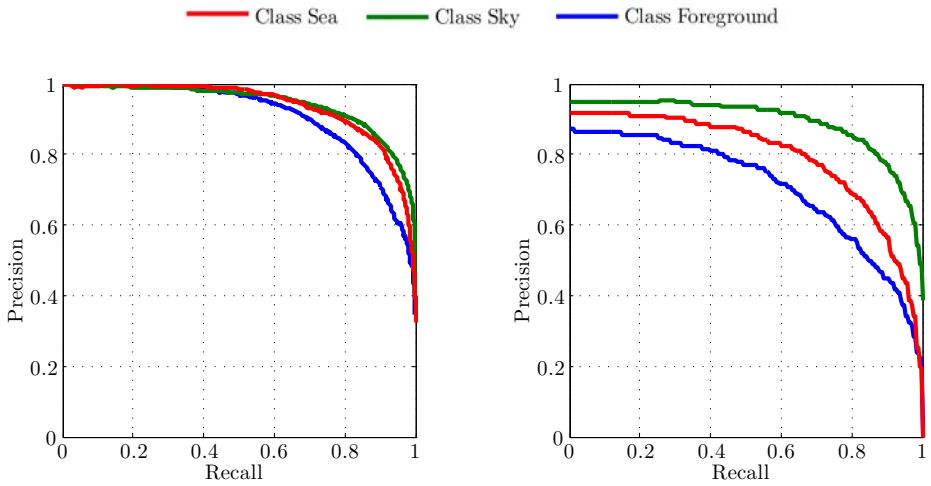
5.1.4 Evaluation

The proposed sea/sky-classifier is intended to be trained just once and then be applied to any dataset of interest. It is first evaluated on the learning set using cross-validation and then tested on the *shipspotting* dataset. The learning set, extracted from the PASCAL VOC dataset, is divided into ten sets for ten-fold cross-validation with each set containing 90% training and 10% testing data, while the *shipspotting* dataset is used for testing only. Following the evaluation criterion defined in Section 2.8, the values for precision, recall, and F-Scores are then computed for each class and dataset, see Table 5.1. A precision/recall plot compiled per class and dataset is depicted in Figure 5.11.

The classifier faces an average drop of performance of approximately 25% for class *sea* and 20% for class *foreground* when evaluated on the *shipspotting* dataset. The overall performance for class *sky*, however, remains constant. In fact, the precision of the classifier dropped from 0.780 to 0.738 for this class but the recall increased from 0.923 to 0.964, which yields a decrease in F_1 and increase in the recall emphasised F_2-score. The precision/recall plot in Figure 5.11 shows an almost identical curve for *sky* (green). This consistency in performance on unseen test data is not surprising as the descriptor computed for class *sky* (Figure 5.8(a)) shows a very distinct peak for a single primary colour and high variance in orientation, allowing for accurate classification. This is due to the unique primary colour of sky as discussed earlier in this chapter.

The primary colour of class *sea* is slightly more diverse compared to *sky* (Figure 5.5(a) and (b)), yielding a lower precision for classification on the *shipspotting* dataset. While the gradient of this class has a unique shape on the learning set (Figure 5.7(b)) due to dominant horizontal lines in waves. However, horizontal lines are also observed in images of class *foreground* (Figure 5.7(c)). This is especially the case for images from the *shipspotting* dataset as established earlier in Section 4.3.1, where a detector sensitive to horizontal and vertical lines was employed to detect man-made objects. Overall, precision of class *sea* dropped only slightly, indicating that regions classified as *sea* in the *shipspotting* dataset

are indeed of the predicted class. The recall rate of this class, however, dropped from 0.806 to 0.551, indicating that not all parts of sea in that dataset were actually detected as sea. On closer inspection it becomes clear that this is due to the higher variation of images in this class in the *shipspotting* dataset. However, because class *sea* did not suffer any loss in precision, parts that are misclassified are therefore classified as *foreground* at worst.



(a) Learning Set (b) *shipspotting* dataset

Figure 5.11: Precision/Recall plot of the Sea/Sky Classifier.

When evaluating the sea/sky classifier for class *foreground* on the *shipspotting* dataset, the drop in recall and precision is not as severe as for class *sea*. The recall of *foreground* dopped from 0.786 to 0.620, indicating that less areas of *foreground* are actually detected compared to the learning set. On the same issue, the drop in precision from 0.842 to 0.691 indicates that the accuracy for the detected areas in the *shipspotting* dataset is lower than on the learning set. As mentioned in the previous paragraph, the performance for class *sky* remained almost constant, concluding that regions of *sea* get classified as *foreground* and vice versa. One difference between *sea* and *foreground* is the shape of the weighted edge histogram (Figure 5.7 (b) and(c)). While class *sea* has clear peaks at 90° and 270°, the edges for *foreground* are more diverse. A closer inspection of the images in *shipspotting* reveals that some images actually show flat sea without the presence of any edges – no images of this type are in the learning set, yielding a misdetection. However, it is not practical to train the classifier on images with no waves as it would mean learning a histogram of an edge image with very weak edges, which would not contribute to the descriptor.

Overall, the results of the sea/sky classifier on the *shipspotting* dataset are satisfactory given that it is pre-trained on an separate learning set. The classifier is intended to

be used as supplementary to the visual attention framework, which allows variations in performance as it does for all other features as well.

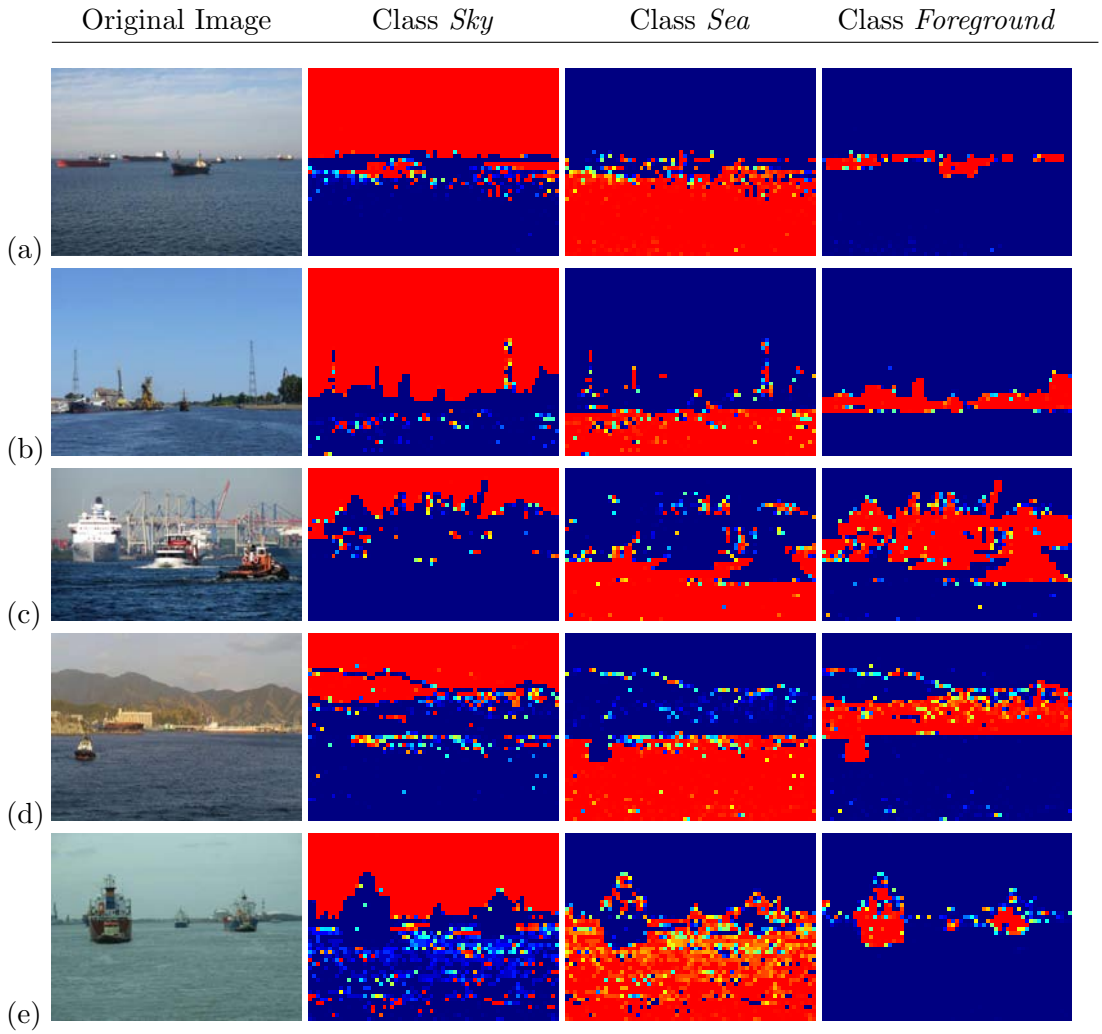


Figure 5.12: Probability maps for classes *sky*, *sea*, and *foreground* as produced by the sea/sky classifier, indicating good classification results on sample images from *shipspotting* dataset.

Classification results for sample images from the *shipspotting* dataset are depicted in Figure 5.12. The image in (a) shows a good overall classifier performance. Some parts of *sea* get classified as *sky* however. This happens only close to the horizon where there are no edges observable due to the distance towards the waves. The detector therefore does not have any edge information for these blocks and classifies based on colour, where the primary colour of *sea* and *sky* is similar (Compare Figure 5.5(a) and (b)). Class *Foreground* in this image is correctly detected. The classifier shows a good performance for class *sky* for the image in (b). The masts in the image are correctly discarded from class sky. However, they get misclassified as *sea* instead of being assigned to class *foreground*. The rest

of the image is correctly classified. (c) shows an image which is almost entirely correctly classified. Class *sky* is, except some few scattered blocks, detected correctly. *Foreground* is correctly classified including the delicate cranes. In image (d), class *sky* is assigned a number of false positives on the mountain on the left side of the image. Again, parts of *sea* in the horizon region get classified as *sky* due to missing edges. Apart from this, detection towards class *sea* is accurate and performance of class *foreground* is satisfactory – except for the mountain as mentioned earlier. (e) yields a good result of the detector with *sky* correctly classified as *sky* with only a minor number of false positives within *sea*. *Foreground* is correctly detected where only some coastal objects in the far distant are missed. *Sea* is correctly detected except some parts of the masts and antennas of the ships getting classified as sea due to the strong presence of horizontal edges in this structure.

5.2 Feature Selection and Classification

The low-level features introduced in Chapter 4 were designed empirically. Therefore this chapter will perform a formal assessment of their contributing strength because they can contain redundant or conflicting information given the class. The Information Gain Criterion (*InfoGain*) can be used to estimate the contributing factor of a feature in classification tasks (Kullback and Leibler, 1951; Russell and Norvig, 2010). Estimating the contributing factor allows the ranking of the features and the disregarding of irrelevant features.

The InfoGain criterion makes use of *entropy* as introduced by Shannon and Weaver (1962). Let $Y = \{Y_1, Y_2, \ldots, Y_N\}$ be a random set of length N with $P(Y_n)$ representing the probability of each set member Y_n, then the entropy, H, of the set is defined as

$$H(Y) = -\sum_{n=1}^{N} P(Y_n) \log P(Y_n). \tag{5.5}$$

InfoGain is then defined as the logarithmic ratio of the entropy of the set member Y_k to the entire set,

$$IG(Y_k) = \log \frac{H(Y)}{H(Y_k)}, \tag{5.6}$$

which can be interpreted as the gained information with respect to the specific set member.

Table 5.2 ranks all features using the *InfoGain* criterion. No strong prevalence for a specific low-feature or locality cue is observable, but on closer inspection it can be seen that three of the top four features are edge based features, suggesting the importance of edges for

visual attention. This corroborates the findings of Rosin (2009) and Alexe et al. (2010), which are based entirely or partly on edge density using either a local or a regional density measure (as discussed in Chapter 2) – their measurements correspond approximately to $\mathbf{Y}_E^L$ and $\mathbf{Y}_E^S$ respectively.

Another notable finding of the InfoGain ranking is that the centre-surround colour feature, $\mathbf{Y}_C^S$, which is the foundation of the saliency approach proposed by Achanta and Süsstrunk (2010) (see Chapter 2 for a detailed discussion), is clearly outperformed by other features. At first, this finding is contrary to the results reported by Achanta and Süsstrunk (2010) and estimated in Section 4.5 of this thesis, where their approach outperforms all other compared algorithms on the *MSRA* database. However, when comparing sample images of both dataset (e.g. Figure 4.22), it becomes clear that the *shipspotting* dataset has less variety in colour than the *MSRA* dataset, for which their algorithm was designed.

#	Gain	Feature	
1.	0.163	$\mathbf{Y}_E^G$	Global Edge Feature
2.	0.149	$\mathbf{Y}_R^G$	Global Right Angle Feature
3.	0.149	$\mathbf{Y}_F^L$	Local Frequency Feature
4.	0.146	$\mathbf{Y}_E^L$	Local Edge Feature
5.	0.141	$\mathbf{Y}_T^L$	Local Textural Feature
6.	0.127	$\mathbf{Y}_E^S$	Centre-surround Edge Feature
7.	0.127	$\mathbf{Y}_F^S$	Centre-surround Frequency Feature
8.	0.122	$\mathbf{Y}^{fg}$	Segmentation: Foreground
9.	0.120	$\mathbf{Y}_C^S$	Centre-surround Colour Feature
10.	0.104	$\mathbf{Y}_T^S$	Centre-surround Textural Feature
11.	0.101	$\mathbf{Y}_R^L$	Local Right Angle Feature
12.	0.086	$\mathbf{Y}_F^G$	Global Frequency Feature
13.	0.084	$\mathbf{Y}_R^S$	Centre-surround Right Angle Feature
14.	0.080	$\mathbf{Y}^{sky}$	Segmentation: Sky
15.	0.077	$\mathbf{Y}_T^L$	Local Textural Feature
16.	0.031	$\mathbf{Y}_C^G$	Global Colour Feature
17.	0.020	$\mathbf{Y}_C^L$	Local Colour Feature
18.	0.003	$\mathbf{Y}^{sea}$	Segmentation: Sea

Table 5.2: Features ranked by InfoGain criterion for the *shipspotting* dataset.

Note that the result for classification into sea or sky with respect to class sea is expected, as the specific sea and sky features are used in a three-class classification problem and therefore redundant because $\mathbf{Y}^{sea} = 1 - \mathbf{Y}^{fg} - \mathbf{Y}^{sky}$.

To estimate the optimum number of features, a learning curve is plotted using the F_2

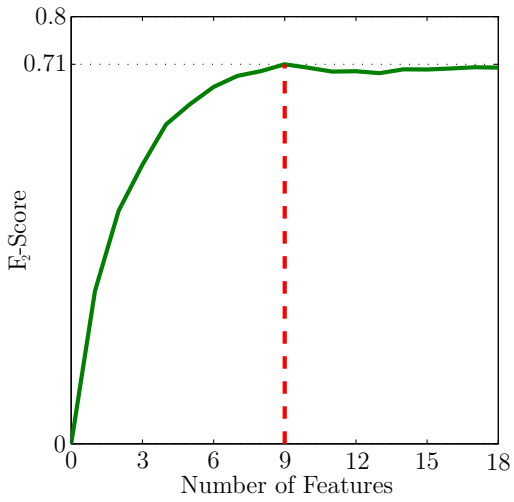


Figure 5.13: Learning curve showing the performance of the framework given features used for classification from the ranking (see Table 5.2). The curve shows that the performance peaks when using the first nine features.

measure for all the features to carefully select the optimal number of features needed for classification. This is important to achieve best performance while minimising the risk of overfitting the classifier. The F_2 measurement is chosen as it combines precision and recall but puts more emphasis on recall than precision as that is what the classifier should be optimised for. From the learning curve depicted in Figure 5.13, a peak can be observed when using the first nine features. Using more than the first nine features does not increase the classification accuracy based on the F_2 measure. The used features are listed in the top part of Table 5.2.

Based on the feature selection process, an updated Bayesian network is created for Naïve Bayes classification using the best nine features as depicted in Figure 5.14.

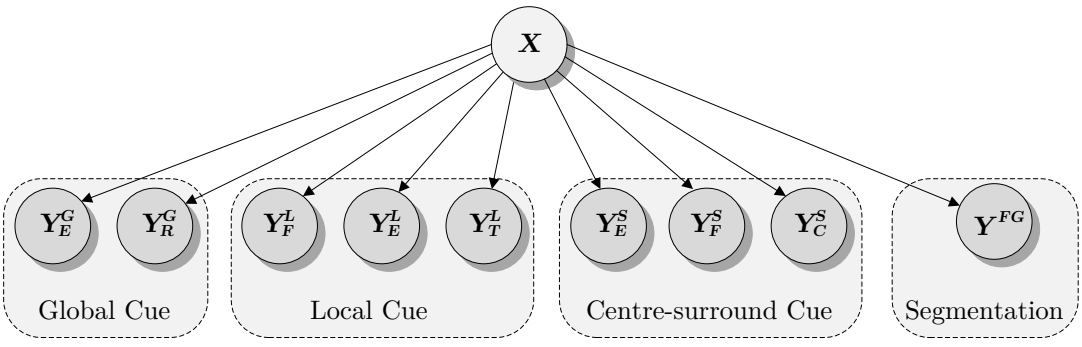


Figure 5.14: Naïve Bayes Network used for classification.

Note that $\mathbf{Y}^{fg}$ is actually the probability $P(S = fg \,|\, d_1,\, d_2, \ldots, d_N)$ from Equation (5.4a). In other words, the probability is reinterpreted as a continuously valued feature. Whilst it would have been possible to integrate the entire sea/sky Baysian classifier as a sub network underneath X, feature selection analysis indicated that the probability information for classes *sea* and *sky* is of less use than *foreground*. Hence, in the interest of a simpler network, $\mathbf{Y}^{fg}$ is treated as a feature by itself.

The learning and evaluation process of the Naïve Bayes classifier is as described in Section 4.4 and 5.1.3.

5.3 Experiments

The proposed approach is compared to the same algorithms as in Chapter 4 (repeated here for the sake of completeness), as well as the approach proposed in Chapter 4:

- Achanta and Süsstrunk (2010) because it is amongst the most recent saliency detectors and has been shown to be highly effective. The authors demonstrated that their proposed method outperforms the works of Itti et al. (1998), Harel et al. (2007), and Hou and Zhang (2007).
- Rosin (2009) due to its simple and parameterless approach which outperforms Itti et al. (1998) and Ma and Zhang (2003), and can keep up with Liu et al. (2007). Although Rosin recommends performing erosion to reduce the overfitting produced by the algorithm, it is evaluated based on the raw results to avoid introducing an additional parameter that must be optimised.
- Alexe et al. (2010) because their *objectness* measure can be used to approach the problem of visual attention in a unconventional way. The authors showed that their approach outperforms Itti et al. (1998) and Hou and Zhang (2007).
- the earlier version of the framework as proposed in Chapter 4.

The resulting maps are normalised to range from $0 \ldots 1$ and evaluated according to the classification criterion introduced in Section 2.8 and the results are shown in precision/recall plots.

The precision/recall plot in Figure 5.15 shows that the proposed approach outperforms all other methods, including the earlier version proposed in Chapter 4 in both precision

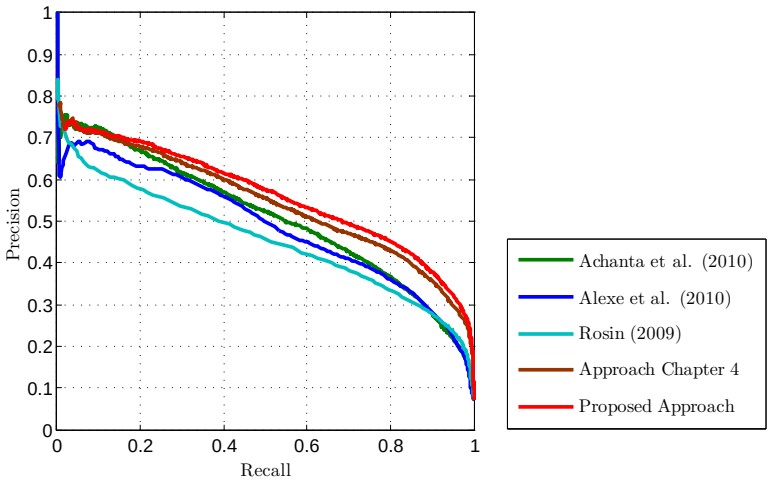


Figure 5.15: Precision/Recall plot comparing the performance of the four evaluated algorithms. The presented algorithms outperforms other algorithms in both precision and recall performance.

and recall. The methods of Achanta and Süsstrunk (2010), Rosin (2009), and Alexe et al. (2010) remain unchanged. The difference between the proposed version for visual attention and the earlier version is the incorporation of a sea/sky classifier for background segmentation as well as the selective use of the available features in the Baysian network. The evaluation is performed on the same *shipspotting* dataset as in Chapter 4, responses to sample images are depicted in Figure 5.16. Evaluation on *MSRA* was purposely omitted due to the domain specific nature of the proposed background segmentation technique, which makes tests on a generic dataset not feasible.

The *objectness* measure proposed by Alexe et al. (2010) yields a consistent detection of most objects in the test set. While their approach typically puts emphasis on the dominant object in the image due to the uniform window approach, it correctly separates multiple objects in some images. In Figure 5.16(a), Alexe et al. (2010) detect the dominant ship on the right with a good recall and acceptable precision. Merely the delicate shape of the superstructure is not correctly highlighted. However, the smaller ship on the left side is weighted much lower than the ship on the right. In fact, the weighting of this area and the connecting region of false positives have almost the same weight. The false positives are caused by the window approach: if a sampled window covers both objects, it yields a high *objectness* measure. But due to the uniform distribution of weights, the entire window and subsequently the empty space between the ships gets highlighted as well. The images in (b) and (c) have been discussed earlier in Chapter 4 when they were compared in Figure 4.26(b) and (c). In (d), the oversegmentation by Alexe et al. (2010) is not as severe. However, not all parts of the two ships are equally weighted. The

superstructure of the ship on the left is not highlighted, so is the bow of the ship on the right. The image in (e) reveals a different problem of the approach. Small objects such as the boat right of the cruise ship are not easily detected. Interestingly the boat on the left is detected, even though no apparent difference between the two boats compared to their respective surrounding region is observable. The entire horizon is highlighted as well, for the same reasons as established for (a). The multitude of ships in (f) challenges the window approach. Because the probability of a window containing multiple objects is much higher, more windows get assigned a high *objectness* measure, which again gets uniformly distributed over the entire window region, yielding a drastic oversegmentation of the image.

A variable performance is produced by the approaches proposed by Achanta and Süsstrunk (2010) on the *shipspotting* dataset. In (a), both ships are reliably detected with acceptable performance and no false positives. The saliency map, however, is undersegmented and the borders of the objects are not highlighted. The performance of Achanta and Süsstrunk (2010) in the images depicted in (b) and (c) has been discussed in Section 4.5.2.1. In (d), the ship on the left is not detected and only the aft of the ship on the right is highlighted. Failure to detect the ship on the left is due to insignificant difference between the ship and the surrounding window mean, which is made up of half sky and half sea, pushing the mean colour vector towards the colour of the target object. The red ship on the right has a significantly different colour than the rest of the image. However, only parts of it get highlighted by Achanta and Süsstrunk (2010). This is due to the low cut-off of the surrounding window at this position of the image. When comparing the bow of the ship, the window includes almost the entire ship, shifting the mean colour vector of the window towards the colour of the ship. The area subsequently does not get highlighted. The images in (e) and (f) are challenging for Achanta and Süsstrunk (2010) as the targets do not differ by much in perceived colour. Only the small boat right of the ship in (e) is detected as the most salient object of the image. Achanta and Süsstrunk (2010) generates mostly noise for the rest of these two images.

The edge density measure proposed by Rosin (2009) is a reliable detector on the *shipspotting* dataset. Almost all objects are detected even though the detector oversegments significantly. Erosion techniques that were suggested as a possible solution to this issue by the author were not performed as target objects in the *shipspotting* dataset are typically very small. A systematical erosion of the result map would risk the detection of small objects and delicate parts thereof. In (a), both ships are detected by Rosin (2009), however, the ship on the right side is weighted higher than the one on the left. The detector oversegments but no false positives are detected elsewhere in the image. The images in (b) and (c) have been discussed earlier in Section 4.5.2.1. Rosin (2009) detects the two ships

in (d) with equal weight and does not emphasise one over the other. Only the borders and the hull of the ships that do not have a strong presence of edges are less highlighted than the rest of the targets. In (e), Rosin (2009) highlights four components in the image. While the lighthouse in the left of the image is a false positive given the definition of maritime visual attention, Rosin (2009) is the only detector – including the proposed one – that is able to detect all targets in this image. The uni-coloured background yields a strong contrast towards the two small boats favouring the edge based approach. While a number of the ships are detected, a high region of false positives are generated in (f). In this particular image, eroding the resulting map might have reduced the false positive rate but also puts at risk the detection of the ships that are very small in size.

The proposed detector for maritime visual attention builds upon the approach described in Chapter 4. The response map of the detector highlights both ships in (a) uniformly and with equal weight. The recall of the detector is very good, only it oversegments slightly towards the bottom for the ship on the right and does not detect the correct contour of the superstructure. However, no false positives are detected in the image resulting in a good overall performance. The image in (b) was previously evaluated by the approach proposed in Chapter 4 – see Figure 4.26(b) and Section 4.5.2.1 for a discussion. There, the sailing boats were detected; however, the image was oversegmented with a high number of false positive regions, mostly at the bottom of the image, where sea is present. The proposed detector correctly discarded these regions using the incorporated sea/sky detector, resulting in a map with good precision and no false positives. Some of the smaller sailing boats are joined into one object however. The image in (c) has been evaluated previously with the approach proposed in Chapter 4 as well. Figure 4.26(c) shows that the detector yielded acceptable recall, detecting all objects – with the exception of a small part in the centre of the big ship on the right side of the image. However, the attention image was overlaid with a significant number of false positive regions. Furthermore, the detector failed to separate the individual ships. The image in (b) on the other hand, shows the three ships on the left separated as individual objects. A number of false positive regions in the bottom of the image that were present in Figure 4.26(c) have also been eliminated due to the sea/sky detector in the proposed approach. The image in (d) shows good recall performance detecting both ships while slightly overfitting the targets. The targets are, however, uniformly highlighted. In (e) the proposed approach detects the cruise ship and the boat to its right. However, the small boat on the left side of the cruise ship is not detected. Instead, the lighthouse on the left side of the image is highlighted, which as mentioned earlier is not a target given the definition of maritime visual attention and has to be treated as a false positive. Almost all ships in (f) are detected by the proposed detector. Some targets that are close together are joined in the resulting attention map. While the targets are slightly oversegmented in the joined map, no false positive objects

are detected.

5.4 Summary

This chapter extended the maritime visual attention framework developed in the previous chapter by integrating a sea/sky classifier into the visual attention detection and eliminating features that play a minimal role in detection accuracy.

The chapter began with a discussion about image background segmentation and its usability to improve classifier performance by reducing the number of false positives. It was established that the dominant background in maritime scenarios is sea and sky. Based on this, a three class classifier that computes the probability of each block of the image belonging to classes *sea*, *sky*, or *foreground* was then proposed. The classifier makes use of the dominant primary colour of sea and sky – information available from the hue channel of the HSV colour model. It further utilises a histogram of orientations built from a weighted edge image, used to detect the dominating horizontal orientation of waves for class *sea*. The classifier was trained on publicly available images containing only parts of sea, sky, as well as random images for class *foreground*. The performance of the sea/sky classifier was evaluated using cross-validation. It was then shown that it can be applied to the *shipspotting* dataset with satisfactory performance without the need for retraining, which is important as it enables the classifier to be used on unseen data. The results of the classifier were then used as additional features in the maritime visual attention framework.

Next, the need for a feature selection process was introduced to reduce the complexity of the framework and to avoid conflicting information incorporated in the framework. The *InfoGain* criterion was employed to rank the input features of the framework. The performance of the framework given the ranked features was then plotted as a learning curve to utilise the optimal number of features for the framework.

Interestingly, colour was found to be a fairly weak feature. This is a seeming contradiction with the highly successful colour-based approach proposed by Achanta and Süsstrunk (2010) but can be explained by the fact that the maritime dataset is markedly less colourful than standard saliency datasets. In fact, Rosin (2009) also comes to a similar conclusion that colour features are more limited in their applicability for general purpose saliency.

The chapter concluded with an experimental evaluation of the extended framework for maritime visual attention on the *shipspotting* dataset and it was shown that it outperforms

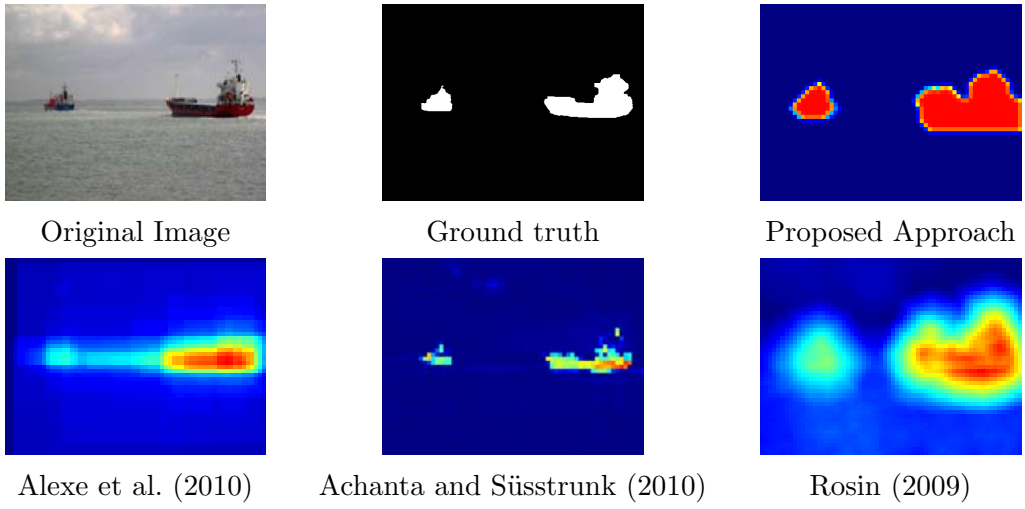
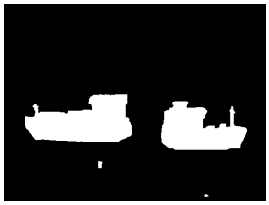
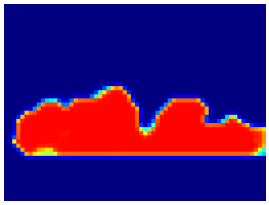


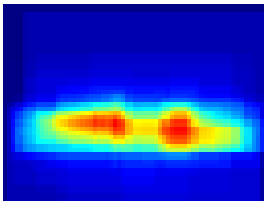
Figure 5.16: Results for *shipspotting* (continued on next page).

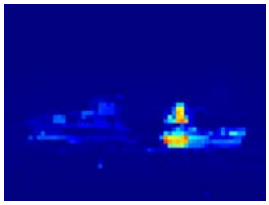
(d)

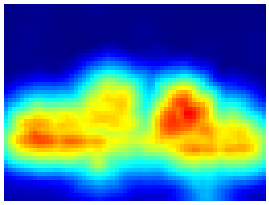

Original Image


Ground truth


Proposed Approach


Alexe et al. (2010)

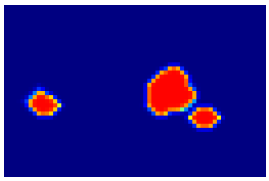

Achanta and Süsstrunk (2010)

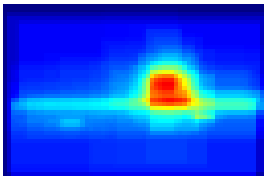

Rosin (2009)

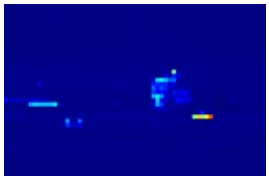
(e)

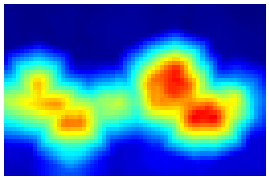

Original Image


Ground truth


Proposed Approach

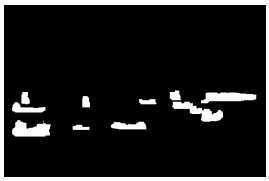

Alexe et al. (2010)

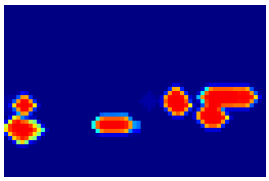

Achanta and Süsstrunk (2010)

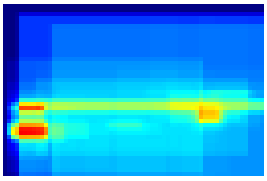

Rosin (2009)

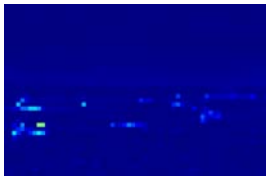
(f)


Original Image


Ground truth


Proposed Approach


Alexe et al. (2010)


Achanta and Süsstrunk (2010)

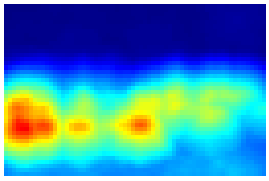

Rosin (2009)

Figure 5.16: Results for *shipspotting* (continued from previous page).

related approaches as well as the approach developed earlier in Chapter 4 in both precision and recall.

Chapter 6

Real World Target Detection and Tracking

In Chapter 3, a target-centric stabilisation technique for omnidirectional cameras was proposed. The technique utilises the extraction of virtual cameras from the full spherical view and adjusts the parameters of these virtual cameras to achieve stabilisation with respect to a target object. An inertial measurement unit (IMU) was utilised to provide an initial guess for the ego-motion of the camera system. A probabilistic feature tracking approach was then applied to track a target object and ultimately adjust the parameters of the virtual camera to achieve stabilisation. It was shown that the approach is robust to loose calibration and inconsistent synchronisation of the hardware components.

However, Chapter 3 only evaluated the performance of the stabilisation approach on a simple stationary target under lab conditions. Therefore, no changes in appearance due to target movement (orientation changes) or lighting changes had to be considered. Moreover, the stabilisation process had to be initialised manually – a shortcoming that the visual attention frameworks of Chapters 4 and 5 seek to address. With the maritime domain in mind, a visual attention framework that detects areas of interest in maritime imagery was proposed in these chapters. It was shown that the approach outperforms generic saliency detectors in domain specific scenes.

This chapter combines the two aforementioned concepts to present a system that is capable of detecting and tracking multiple target objects independently and simultaneously. For detection and initialisation, the visual attention framework presented in Chapter 5 is employed. Then, the stabilisation technique presented in Chapter 3 is utilised and a stabilised virtual camera is created from the omnidirectional view for each detected object. The system is applied to the problem of stabilised tracking of nearby objects in omnidirectional views on a real moving maritime platform. In addition to the platform's forward motion due to propulsion, the platform is also subject to significant and unpredictable motions and disturbances due to the speed through the waves – challenging conditions that have been discussed in Chapters 1 and 3.

This chapter begins with an analysis of the benchmark datasets used in Chapters 4 and 5, in order to explore why the *shipspotting* dataset was significantly more difficult for state-of-art detectors to correctly find objects and identify regions of interest. It will be shown that these datasets in fact have almost certainly been unintentionally influenced by human shot selection and the algorithms are tending to reflect these selection methods rather than provide unbiased true object detection.

The finding motivates the need to evaluate the proposed visual attention framework on genuine imagery that has no possibility of unintentional human interference in order to provide a true test of the ability of the computational algorithm to find objects of interest autonomously in a real world deployed situation. Omnidirectional cameras are ideal for this purpose as they capture the entire scene without any selective choices of the field of view that a human photographer would have. Subsequently, the following challenges are considered: Firstly, the omnidirectional camera captures the full scene, even sections of the environment that are destructive to vision (and ultimately computer vision algorithms), such as direct exposure of the sun or its reflective glare on the water. Secondly, a full spherical view means that objects become easily very small relative to the overall image size, which is in contrast to saliency datasets where objects are always significant in the image.

In order to fit with the domain of application, the camera system was mounted on a small boat and a video sequence was recorded. Detection of visual attention was performed on the extracted omnidirectional video. It is important to note that the framework is trained only on the *shipspotting* dataset introduced in Chapters 4 and 5. Furthermore, the sea/sky detector introduced as a part of the extended visual attention framework has not been re-trained. Specifically, the whole proposed visual attention framework is used verbatim and applied to a far more challenging scene.

Detection then provides potential targets to be tracked, and stabilised tracking is initialised from this. Tracking is then performed for the duration of the video to show the ability of the system to track multiple targets simultaneously whilst stabilising all targets independently within their fields of view. This allows for tracking of different target movements and compensates for any parallax effects – issues that are significant in the scenes since targets are significantly closer than background objects.

Finally, the difficulties in detecting relevant regions of interest are highlighted and the adequacy of the proposed approach as well as state-of-the-art algorithms for this task are discussed.

The remainder of this chapter is organised as follows: In Section 6.1, a statistical analysis of the datasets utilised in Chapters 4 and 5 is performed, motivated by the different performance of the detectors on the two datasets. Then, the specific challenges introduced by the use of omnidirectional video recorded in maritime scenes are discussed in Section 6.2. The chapter continues with an evaluation of the proposed visual attention framework on omnidirectional imagery and a subsequent use of the results to initialise the tracking part of the proposed stabilisation framework. The chapter concludes with a summary given in Section 6.3.

6.1 Analysis of Benchmark Datasets

The proposed visual attention framework was compared to several other approaches in Chapter 4 using a publicly available benchmark dataset, *MSRA*. Furthermore, tests on a domain-specific dataset (*shipspotting*) that has been compiled for the purposes of this thesis, have been performed in Chapter 4 and 5. A difference between the two was that all algorithms performed poorer on *shipspotting* than *MSRA*, except Rosin's that performed marginally better. However, *MSRA* is a dataset with a high variety of object classes and backgrounds whereas *shipspotting* contains a very low variety of backgrounds and significant similarities between foreground objects since they are mostly maritime vessels. Thus *shipspotting* should have been a lesser challenge to the algorithms tested but in fact the reverse is true.

To resolve this contradiction, a closer look was taken at the overall characteristics of the two datasets. Specifically, the ground truth was analysed to examine the placements and properties of the objects in the scene. Note that the ground-truth data for both datasets are at the pixel level rather that the conventional approach using bounding boxes, i.e. defined by the outer boundary of the object. As mentioned in Chapter 4, this produces the actual shape of the objects and provides a more realistic measure. Analysis on the datasets was performed as follows:

Object Placement. To gain statistics on the overall placement of objects in the images, the average across all ground truths was taken to produce an image that indicates the average occurrence of an object at each pixel. This will indicate the diversity of placement of objects and uncover any favoured positions. In effect, it is a probability map of the likelihood that a given pixel will be part of an object. Thus, highly diverse placement should provide a uniform distribution across all pixels whereas strongly favoured positions should result in peaks at those positions. Since images are not

all of the same resolution or aspect ratio, ground-truth images were resized to the average image resolution of each data set: 323×369 for *MSRA* and 359×510 for *shipspotting*.

Object Count. Another important statistic describing the challenge in a saliency dataset relates to the number of salient objects in an image. Early saliency approaches tended to focus on finding the single most salient objects, e.g. Itti et al. (1998). This limitation has largely been overcome as the proposed algorithms shows. However, it follows that a more challenging dataset will contain more objects per image. Thus the average count of objects per image is also examined.

Relative Size of Objects. Of crucial importance to the application that this thesis addresses is the need to find objects that are small relative to the overall size of the captured image. This arises due to the use of an omnidirectional image that has very high resolution and therefore target objects, whilst well-described and with significant numbers of pixels themselves, are in fact only a very small portion of the overall image. This is both because of the distance to objects as well as the high resolution and the full spherical field of view.

6.1.1 Placement Analysis of MSRA

The ground truth maps of *MSRA* are averaged and the result is shown as a heatmap in Figure 6.1. The heatmap reveals a curious phenomenon: specifically, it is clear that objects in *MSRA* tend to be clustered around the centre of the images. The symmetry and regularity of the average ground truth image is striking – it indicates that human shot selection has had a major influence on the dataset, choosing shots that roughly centre the object in every image. Due to the variety of objects and number of images, this ends up being a circular pattern.

In light of the regularity of the ground truths, it raises the possibility of producing the simplest possible saliency detector and evaluating it on the dataset: a detector that simply "detects" a *single fixed area* of every image. Due to the circular nature of Figure 6.1, a circle with the origin at the centre of the image was chosen and a precision/recall plot is produced by varying the radius of the circle from a single pixel through to the full size of the image. This "naïve detector" should be viewed in two lights: first as a baseline performance for saliency algorithms, and second as a measure of the challenge that a dataset provides.

The resulting precision/recall plot is depicted in Figure 6.2 alongside the precision/recall

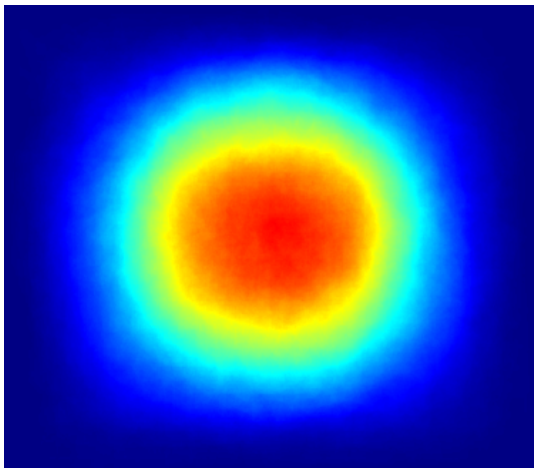


Figure 6.1: Average object placement in the *MSRA* dataset.

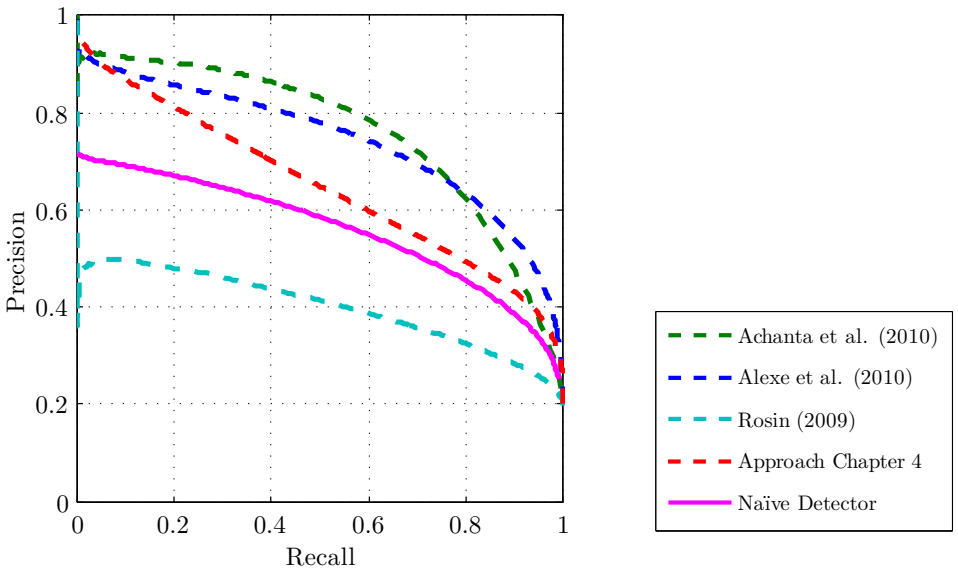


Figure 6.2: Precision/recall plot of the naïve detector on the *MSRA* dataset.

results from the other algorithms from Chapter 4. Indubitably, the naïve detector performs quite well given its simplicity. Note that Rosin (2009) actually performs worse than the naïve detector. However, lower performance for this detector is in terms of weaker precision and it is already known that the precision of the approach can be improved via Rosin's suggestion to erode the final map result. Thus it is probable that the results can be markedly improved with such erosion on *MSRA* given the effectiveness of the naïve detector.

6.1.2 Placement Analysis of Shipspotting

The ground truth data of the *shipspotting* dataset was processed in the same fashion as for *MSRA*. Figure 6.3 shows the average placement of objects in a heatmap representation. Note that the peaks in the heatmap are much more diverse and far less regular than *MSRA* and that a horizontal spread can be observed, which occurs due to the horizon and maritime objects being on the sea surface. However, the horizontal spread away from the centre is fairly uniform – indicating that objects tend to be distributed randomly along the horizon, probably due to amateur photographers seeking to juxtapose multiple maritime objects in a single image, hence objects are often on both sides of the image.

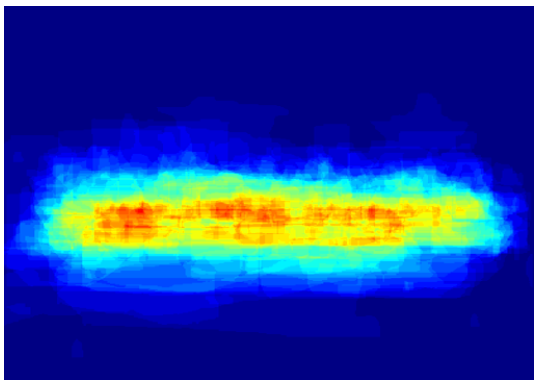


Figure 6.3: Average object placement in the *shipspotting* dataset.

Based on these observations, the parameters for the naïve detector were adapted to suit this dataset. Instead of a circular shape, a rectangle with a 1:3 aspect ratio was selected as a detector shape (other ratios were tested but were slightly less effective). The rectangle was placed in the centre of the image and its size was varied from 1 pixel to the full size of the image to produce a precision/recall plot. Figure 6.4 shows the performance of the naïve detector alongside the results from Chapters 4 and 5.

The precision/recall plot in Figure 6.4 reveals that the performance of the naïve detec-

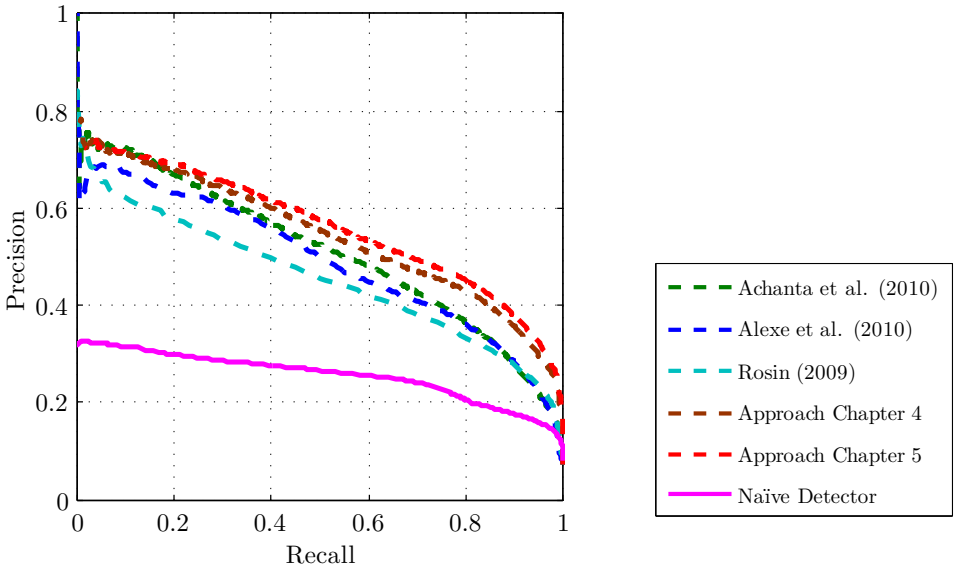
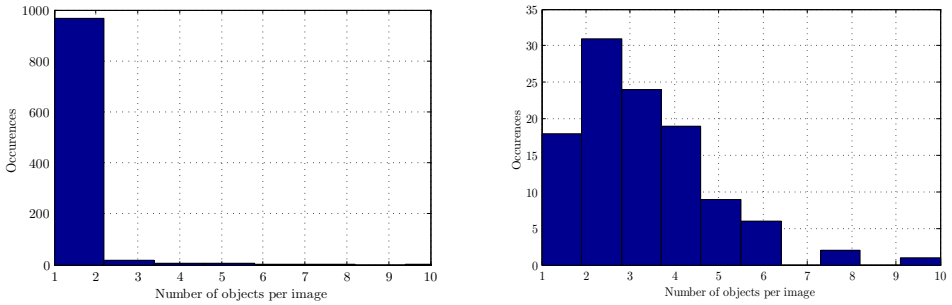


Figure 6.4: Precision/recall plot of the naïve detector on the *shipspotting* dataset.

tor on the *shipspotting* dataset is much lower than on *MSRA*. Specifically, the precision performance almost halved, indicating that the naïve detector produced much more false positives compared to *MSRA*. Note that all algorithms yield better performance than this baseline detector on the *shipspotting* dataset. This implies that the dataset is providing a more genuine indication of the performance of saliency detection since position is far less consistent.

6.1.3 Analysis of Object Count

From the ground truth data of the *MSRA* and *shipspotting* datasets, the number of objects in each image were counted using the connected component technique. The resulting histograms are shown in Figure 6.5. Note that the histogram of *MSRA* is very narrow and dense compared to *shipspotting*. This indicates that the *shipspotting* dataset has a higher variety and is much more diverse than *MSRA*. In fact, more than 95% of the images in *MSRA* only contain a single object, compared to less than 20% for *shipspotting*. Furthermore, more than 80% of the images in the *shipspotting* dataset contains at least two objects. The average number of objects in *MSRA* is only 1.18 compared to 3.04 in the *shipspotting* dataset (see Table 6.1). Moreover, the standard deviation of *shipspotting* is almost twice as high indicating a higher variation of object counts.

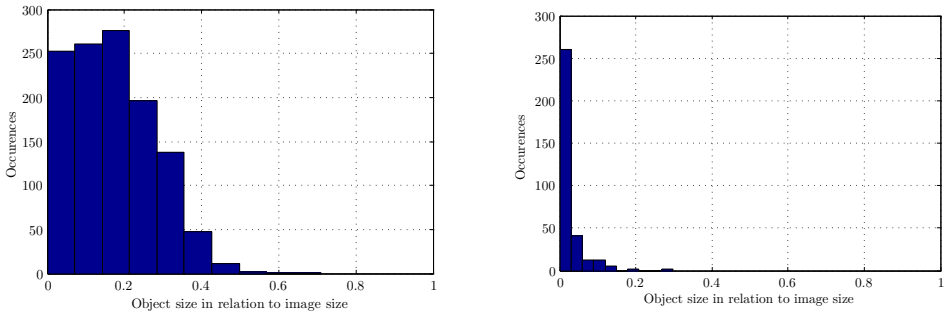


(a) *MSRA* (b) *shipspotting*

Figure 6.5: Number of objects per image in the *MSRA* and *shipspotting* datasets.

6.1.4 Analysis of Object Size

For each image of the *MSRA* and *shipspotting* datasets, the sizes of the contained objects were computed using the ground truth data. The sizes were then converted into a ratio with respect to the image size, where a value of 1 represents an object that covers the entire image. Figure 6.6 shows a histogram of the computed object sizes for each dataset. For the *MSRA* dataset, the histogram shows a high variation of the object sizes up to 40%



(a) MSRA (b) shipspotting

Figure 6.6: Relative Size of objects in the *MSRA* and *shipspotting* datasets.

of the image size, while most of the objects of the *shipspotting* dataset are smaller than 5% and hardly any objects are larger than 10% of the image. This is an important finding as the ultimate aim of the proposed vision system is the deployment on an omnidirectional camera, where target objects are expected to be very small due to the large field of view of the camera. In fact, in an omnidirectional image of 2048×1024 pixels, an object size of 16.88% (as with *MSRA*) would relate to a pixel area of roughly 350 000 – an edge length of more than 590×590 in the case of an object with a square shape. In an omnidirectional image this would cover a field of view of almost 30°. Given that the distance to a target object in an outdoor setting is typically large, the physical size of the object would have to be enormous in order to match the properties of the *MSRA* dataset. In comparison,

the average size of an object of the *shipspotting* dataset would relate to a square object with an edge length of 270×270 pixels or 13° of the field of view. Although this is still quite large given the real footage examined later in this chapter, it is nevertheless far more realistic.

	MSRA	*shipspotting*
Average image size [pixel]	$119\,317 \pm 14\,579$	$183\,016 \pm 19\,395$
Average number of objects in image	1.18 ± 0.90	3.04 ± 1.69
Average relative size of object [%]	16.88 ± 0.11	2.25 ± 0.03

Table 6.1: Average statistics of the *MSRA* and *shipspotting* datasets.

6.1.5 Summary of Analysis

With respect to the domain of application, the dataset analysis revealed that both the *MSRA* and *shipspotting* datasets are far from being representative for real world scenarios.

In terms of object placement, *MSRA* cannot be seen as a dataset with a high diversity as the objects are prominently placed in the centre of the images. The *shipspotting* dataset provides a higher variety. While most of the spreading is in the horizontal direction, this is actually feasible because in a panoramic image, maritime objects such as ships would be distributed mostly along the horizon.

The analysis of the number of objects in an image revealed that *MSRA* is mostly concerned with detecting a single object. In actual outdoor settings, this constraint or assumption is violated as these images (especially when using an omnidirectional camera) will contain many more objects. With almost thrice the number of objects, *shipspotting* shows a higher variety but even this is still relatively low given that busy maritime scenes such as ports could easily contain a dozen or more vessels moving around.

For the object size, the *MSRA* dataset showed a high variation indicating that a large number of differently sized objects are present in the dataset, while the *shipspotting* dataset only contains small objects with less variation in their size. The object sizes of images from *shipspotting* are much smaller, which would correspond better to real-world scenarios, especially when using an omnidirectional camera.

6.2 Visual Attention and Stabilisation in Omnidirectional Video

The stabilisation framework presented in Chapter 3 was developed for an omnidirectional camera system with the application of a maritime surveillance platform in mind. The ultimate task of the platform is to detect and track target objects, therefore a visual attention system that directs attention towards maritime objects was proposed in Chapter 4 and further extended in Chapter 5 to be utilised for detection of potential objects. This section now presents the combination of the two systems and evaluates the performance of the proposed system as a multi target detector and tracker in a maritime outdoor setting.

As discussed in Section 6.1, datasets utilised for evaluation of the visual attention framework have shortcomings for their stated purpose with respect to the problem domain of this thesis. In fact, *MSRA* only shows significantly sized objects at prominent positions within the image and although the *shipspotting* dataset provides more realistic imagery for this domain, it still contains objects that are difficult to overlook in the scene. In contrast, in an omnidirectional image, there is no centre of the image and objects will be much smaller compared to the overall size of the image. Hence, it is important to test detection in omnidirectional imagery.

Once detected, a target object needs to be observed and tracked by the camera system. On a moving platform, with significant ego-motion the conventional approach is to stabilise the image first and initialise and run an object tracker on the stabilised image. As discussed in Chapter 3, stabilisation is essential to reduce the search space of the feature tracker.

Zhou et al. (2010) pointed out that tracking in spherical omnidirectional video is a difficult task and proposed using a cubic panorama representation, where the full omnidirectional view is projected on the inside of a cube, resulting in six independent images with a field of view of 90° each. They constructed epipolar lines across the sides of the cubes to handle the hand over of a target from one image to an adjacent one. This thesis argues that the proposed approach of using virtual cameras by dynamically extracting regions of interest from the continuous full spherical representation of the omnidirectional image not only overcomes the aforementioned hand over problem but also, as a general approach, allows for arbitrary movement of both target and camera and can subsequently be applied without prior stabilisation.

The proposed approach makes use of an image feature tracker for stabilisation by tracking the object and computing the optimal inverse orientation of a virtual camera to focus the

view onto the target. As demonstrated in Chapter 3 this allows for reliably stabilising the view towards a target object despite the presence of significant ego-motion and parallax. This also means that the very same system can be used for stabilisation when both camera ego-motion and target motion are present.

The major issue facing tracking is a change of appearance in the target when it changes its orientation with respect to the camera. However, this is a standard tracking problem and has been investigated by a number or researchers – see Yilmaz et al. (2006) for an overview. In this thesis the approach proposed by Shi and Tomasi (1994) was utilised and correctly matched feature descriptors were updated and new features computed within the target region if the matching quality dropped below a certain threshold.

One of the reasons to choose an omnidirectional camera over a pan-tilt-zoom (PTZ) camera in this thesis was that it allows for simultaneous views in all directions. It is therefore easily possible to simultaneously extract multiple regions of interest from the omnidirectional camera, effectively creating multiple *independent* virtual cameras. The clear advantage of the independence of these cameras is that the feature tracker can run independently within each camera as well, therefore problems with overlapping targets are minimised (since each view tracks its own set of features) and no combinatorial multi-target-tracking issues such as track coalescence have to be solved.

6.2.1 Experiments

To evaluate the capability of the system to track multiple targets, an omnidirectional video was captured from a moving small power boat. The camera was mounted near the rear of the boat on a pole approximately two metres above the deck. The boat was then driven at speed around an ocean port near the coast. The video contains two target objects (both boats) with these targets at different distances and moving with different speeds, which introduces more challenges due to parallax effects. The visual attention detector from Chapter 5 was employed on the first frame of the omnidirectional video to produce a set of candidate regions of interest for initialising the tracker. A set of tracks was initialised from these regions and each region tracked over the duration of the video (1600 frames).

It is important to note that the visual attention framework (including the sea/sky detector) is using the training data gathered from the *shipspotting* dataset. It is not re-trained on the scene but applied verbatim.

6.2.1.1 Visual Attention

Processing of omnidirectional video in outdoor environments is challenging because of the distortions introduced by sunlight. As discussed earlier, especially in maritime scenes, this plays an important role because of the reflective characteristics of the water and subsequent glare. An initial manual inspection of the image revealed strong distortions in the form of Moire patterns in the image, therefore a median filter was applied to all images before processing.

From the omnidirectional view, the first frame was extracted (Figure 6.7(a), (b) shows the ground truth). In (a), the front of the boat can be seen in the bottom half of the image on the left side. The wake can be seen towards the right of the image. Challenges revealed for the omnidirectional input image are:

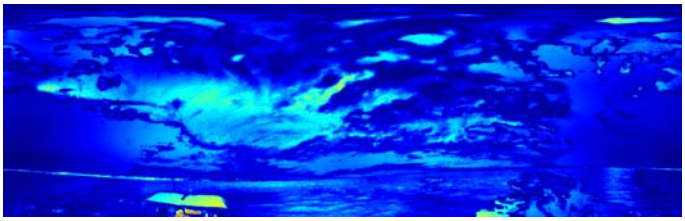
- The target objects in the image are extremely small compared to the overall image size (as can be seen by the size of the black blobs in Figure 6.7(b).
- A significant region of the image (the upper half) is occupied by complex cloud formations. Almost all of these clouds appear in a very bright light due to sun glare.
- Reflections of the sunlight in the water are strongly visible on the left and right side of the image (glare).
- The surveillance platform is partly visible in the image (bottom left of image).
- A region of the image is filled with the wake caused by the surveillance platform itself (right side of image).

Accordingly, even a human has difficulty correctly identifying the targets of interest in the scene. The image was evaluated by running all saliency algorithms from Chapter 5 on it, including the proposed detector. Figure 6.7(c)–(f) show the responses of Achanta and Süsstrunk (2010), Alexe et al. (2010), Rosin (2009), and the proposed approach respectively. Clearly, it can be seen that the responses are an overreaction to what is required – all detectors find significant areas of interest despite the ground truth's sparsity. However, after the analysis of *MSRA* and *shipspotting* in Section 6.1, the results are not completely unexpected.

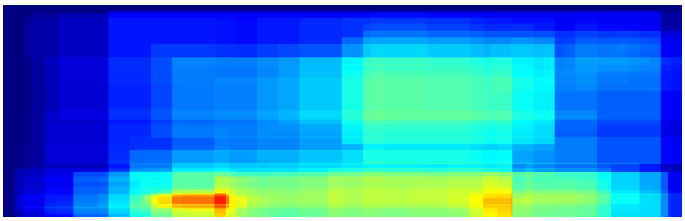
The response of the detector proposed by Achanta and Süsstrunk (2010) is shown in Figure 6.7(a). Due to the maximum symmetric window approach of their detector, the clouds and glare on both sides of the image is suppressed as the window are only comparing

(a) Input image

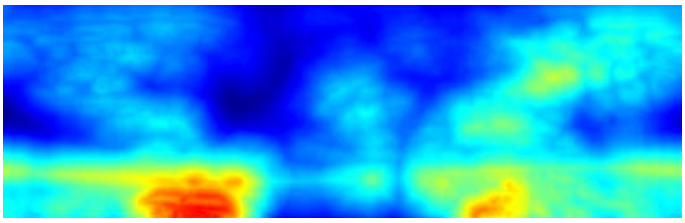
(b) Ground truth



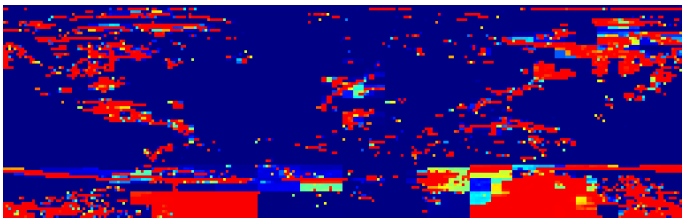
(c) Achanta and Süsstrunk (2010)



(d) Alexe et al. (2010)



(e) Rosin (2009)



(f) Proposed Approach

Figure 6.7: Visual Attention on the omnidirectional image

within this region. The highest response of the detector can be seen on the surveillance platform itself, where the detector highlights parts of the dashboard and of the boat. Whilst this is undoubtedly a salient region in the general sense, the approach almost completely misses the actual targets themselves.

The method proposed by Alexe et al. (2010) shows a high objectness measure and subsequently a high probability for the surveillance platform to be a target object. A second peak is observable on the wake that is caused by the platform. Furthermore it highlights a cloud formation as potentially relevant. However, the detector highlights the area between wake and surveillance platform, which includes the target objects as well, even though the weighting is lower than with the platform indicating that the detector does not find anything of high interest in that intervening region.

The approach of Rosin (2009), based on edge density, also highlights the surveillance platform and the wake in the right part of the image. However, the two target objects themselves are actually being picked up quite well. More importantly, the target ships are detected as relatively separate objects. However, the confidence is not very high compared to the rest of the image so that segmenting them from the background could be a difficult task.

The proposed approach tends to produce the lowest raw number of false positive blocks, rejecting a lot of the sky as background. However, the complex cloud formations remain due to unexpected colours. Furthermore, the horizon line is detected as a potential target due to the high contrast towards the glare on the left and right side of the image. As with the other approaches, the maritime platform and wake are strongly detected. However, the targets themselves are also strongly detected and separable from the rest. Unfortunately, the detector also tends to produce many fractured smaller detections, making it difficult to determine what is an object and what is noise – particularly given that the smaller target vessel is only a couple of 8×8 blocks in size. These false positives and fracturing would make automatic initialisation a very difficult task.

6.2.1.2 Initialisation of Tracks

Table 6.2 shows the precision and recall of the various algorithms with a threshold of 0.5. Note that the precision is very low in all cases. Figure 6.8 shows the output of the algorithms with their optimal respective threshold (tailored for this image). Note that with a threshold of 0.5, the edge density based approach proposed by Rosin (2009) produces a map that entirely covers both target objects yielding a recall value of 1. The

map has the potential to clearly separate the smaller vessel but combines the larger vessel with the wake. In contrast, Achanta and Süsstrunk (2010) and Alexe et al. (2010) produce segmentations that in no way could be used to initialise a tracker on the target objects.

	Precision	Recall
Achanta and Süsstrunk (2010)	$8.43 \cdot 10^{-5}$	$6.72 \cdot 10^{-3}$
Alexe et al. (2010)	$7.79 \cdot 10^{-3}$	0.31
Rosin (2009)	$4.33 \cdot 10^{-3}$	1
Proposed Approach	$8.80 \cdot 10^{-3}$	0.93

Table 6.2: Precision/Recall values for the various algorithms for the omnidirectional input image.

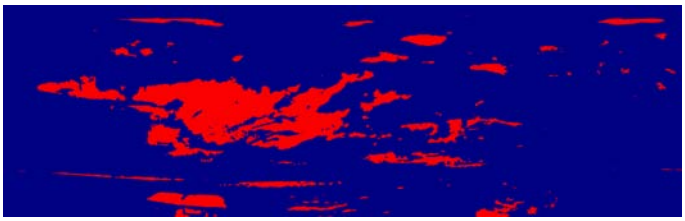
The proposed approach can manage to fully separate both targets from their surroundings, however there are also many small false positives (as well as the large blobs for the surveillance platform on the left and its wake on the right). Hence no approach could feasibly be used to initialise tracking automatically. However, the proposed approach at least does not undersegment and if fractured detection of objects such as clouds and the wake could be combined whilst not merging with the actual targets, initialisation would be a feasible prospect – false positives would be tracked, but so too would the true targets.

Unfortunately, such a merging operation is complex and beyond the scope of this thesis. Hence to gain an understanding of ability of the tracker to work in such a complex scene under conditions of many false positives, a set of 16 initial tracks were manually extracted based on heuristically clustering the responses into components. Thus in addition to the blobs describing the target vessels, the surveillance platform and wake are false positives as are several sections of the clouds.

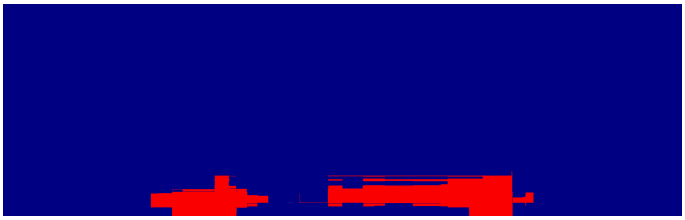
6.2.2 Multi Target Tracking

The stabilisation framework proposed in Chapter 3 had only been tested in an indoor environment. Now, the omnidirectional video recorded by the maritime platform is used.

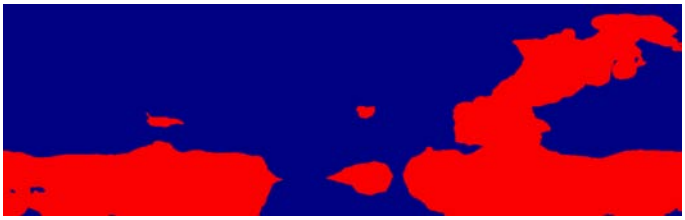
It is important to note that one of the key achievements of the proposed stabilisation framework is that it does not require precise calibration and synchronisation. Therefore, no re-calibration of the camera and IMU has been performed prior to running the experiments. In fact, the timespan between calibration and recording of this footage was more than one year and the assembly has been taken apart and reassembled a number of times in between.



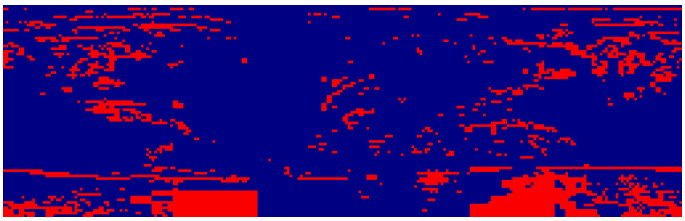
(a) Achanta and Süsstrunk (2010)



(b) Alexe et al. (2010)



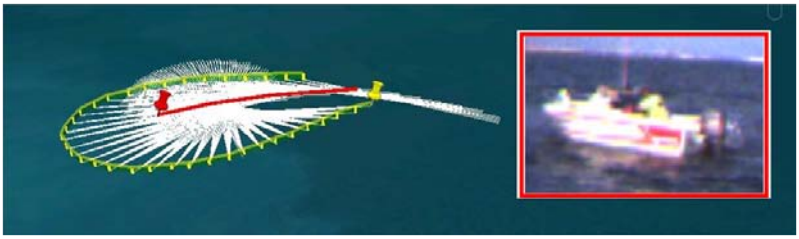
(c) Rosin (2009)



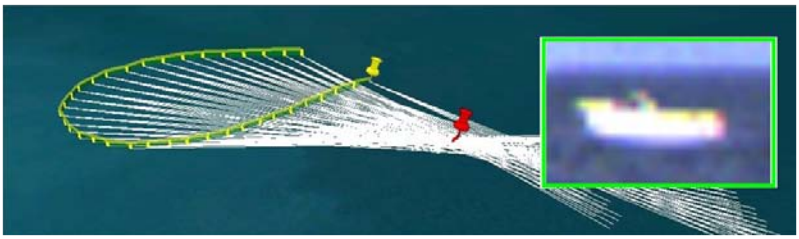
(d) Proposed Approach

Figure 6.8: Optimal visual attention maps.

Given the manual detections, a number of virtual cameras are extracted from the omnidirectional view with the 16 targets (both actual and false positive targets) centred in the respective views using the initialisation procedure described in Section 3.2.2. Tracking then proceeds independently for each target in its respective virtual camera, using the algorithm described in Chapter 3. Results of tracking are depicted in Figure 6.9, showing example frames every 4 seconds. The figures show the tracks within the omnidirectional view at the bottom and the stabilised views of the two true targets at the top of each figure. True target positions are highlighted in red and green in the omnidirectional view whereas false positives are shown in blue. Note that the true targets are tracked very well, despite the fact that they eventually recede very far away from the camera. Furthermore, stabilisation of the targets is good, with the horizon uniformly located and rarely tilted significantly. Tracking succeeds stably despite occasionally overlapping with false positive virtual cameras, demonstrating that the feature-based tracker is not diverted by other, independent, tracks. Not shown explicitly is how the false positive tracks behave – these tend either to stay relatively fixed or, in the case of the wake, move with the flow of the water.



(a) Target 1 (red) – moving



(b) Target 2 (green) – stationary

Table 6.3: Trajectory of the platform (yellow) and position of the two target objects.

For visualisation purposes, the maritime platform has been equipped with a GPS, recording position data in the earth coordinate system, $\{E\}$. Figure 6.3 shows the trajectory of the platform (yellow) with the starting position indicated by the yellow pin. At each time step, the orientation of the virtual cameras with respect to the global coordinate system, ${}^{V_n}_{G}\mathbf{T}_t$, where $n = \{1, 2\}$, was used to compute the projection of the target objects onto the unit sphere of the global coordinate system that is spanned at the current location of the maritime platform, see Section 2.1.2.1 and 2.1.2.2 for details. The projection was then

used to plot rays originating at the current position towards the target objects. Due to the projection onto the unit sphere no depth information is available. However, it can be seen that the rays intersect at the position of the boats. In case of target one (Figure 6.3(a)), the intersections actually form a line indicating that the target was moving. This qualitatively demonstrates that the various transformations, mapping, and tracking works well if good initialisation (correctly detecting objects using visual attention) is realised. For the moving target (red), the predicted position varies whereas for the stationary target, all the projected rays approximately converge to the same location as expected.

In terms of performance, the tracker runs in near-real-time with no optimisation of the C++ code, despite tracking 16 individual targets. This is largely due to the efficiency of the Lucas-Kanade tracker (Shi and Tomasi, 1994) and the linear scaling of the particle filter. Tests show that scaling with number of targets is roughly linear. However, the initial detection itself is quite slow and could not be performed for every frame – new detection runs could only be performed every few seconds (exact expected performance of detection is difficult to define since much of the detection code was written in Matlab and runs in batch rather than online).

In all, the stabilised tracking is quite robust. Although only a limited evaluation (on one video) has been performed and generalisations are thus difficult to make, the tracking results indicate that robust automated tracking should be an achievable goal in a real-world omnidirectional scenario. The main issue remains the problem of initialising the tracker with reasonable starting estimates – if the false positives in the detection phase can be greatly reduced, then subsequent tracking should be a feasible prospect.

6.3 Summary

This chapter presented an application that deployed the image stabilisation technique proposed in Chapter 3 and the visual attention framework proposed in Chapters 4 and 5 in a real-world setting. Specifically, the omnidirectional camera system was utilised on a maritime platform and utilised to capture full spherical omnidirectional imagery. As expected, the platform was subject to significant motion disturbances, demanding the use of a stabilisation technique. The proposed stabilisation technique required manual initialisation. This shortcoming has been addressed by applying the visual attention framework to the omnidirectional imagery and subsequently stabilising the image on attentive regions.

The chapter began with an analysis of the datasets that were used for evaluation in the

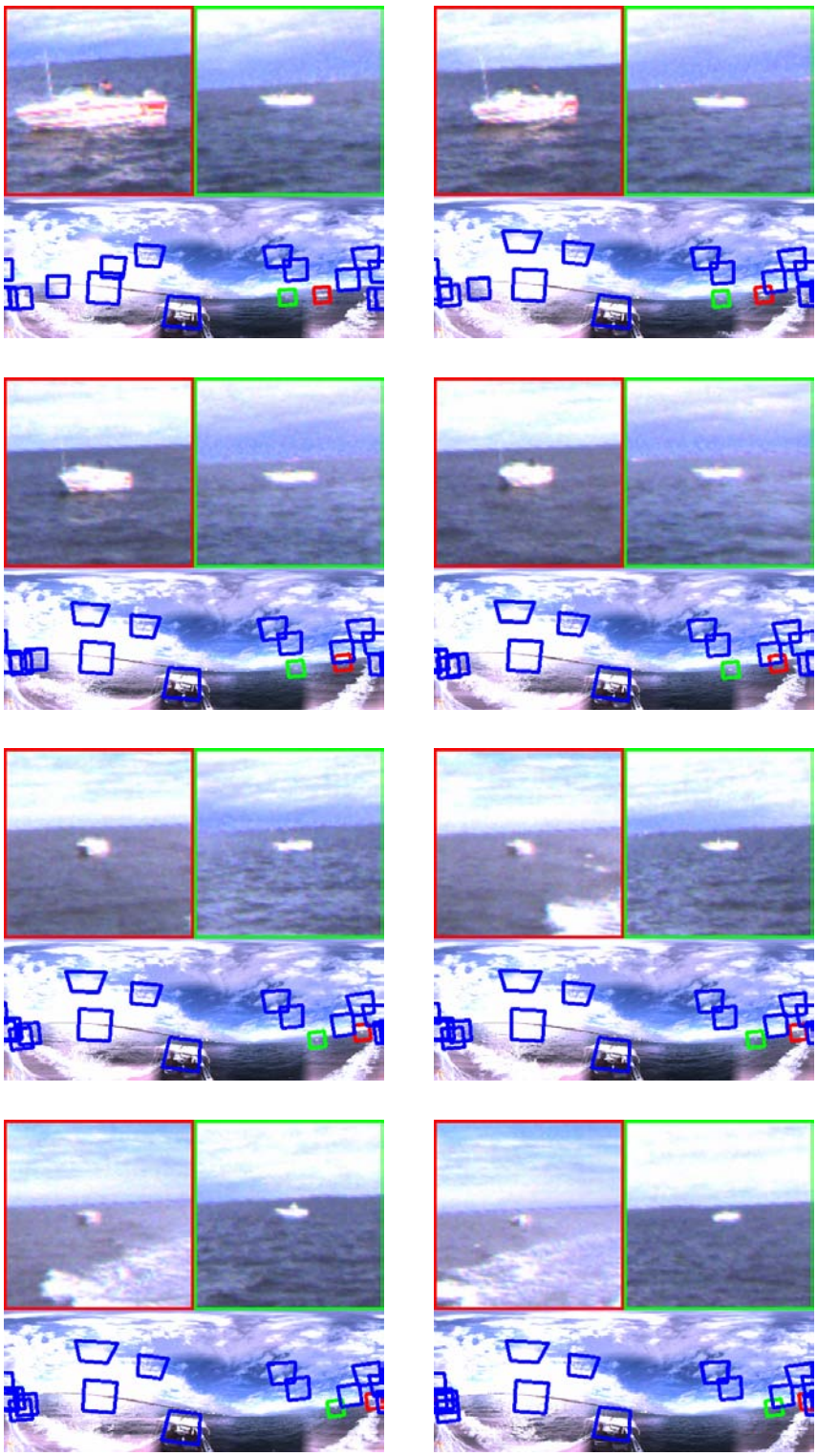


Figure 6.9: Frames 1–800 (left to right, top to bottom) showing the raw omnidirectional view together with two extracted virtual cameras fixed on targets (red and green). The omnidirectional view also shows a number of false positives being tracked by other virtual cameras (continued on next page)

.

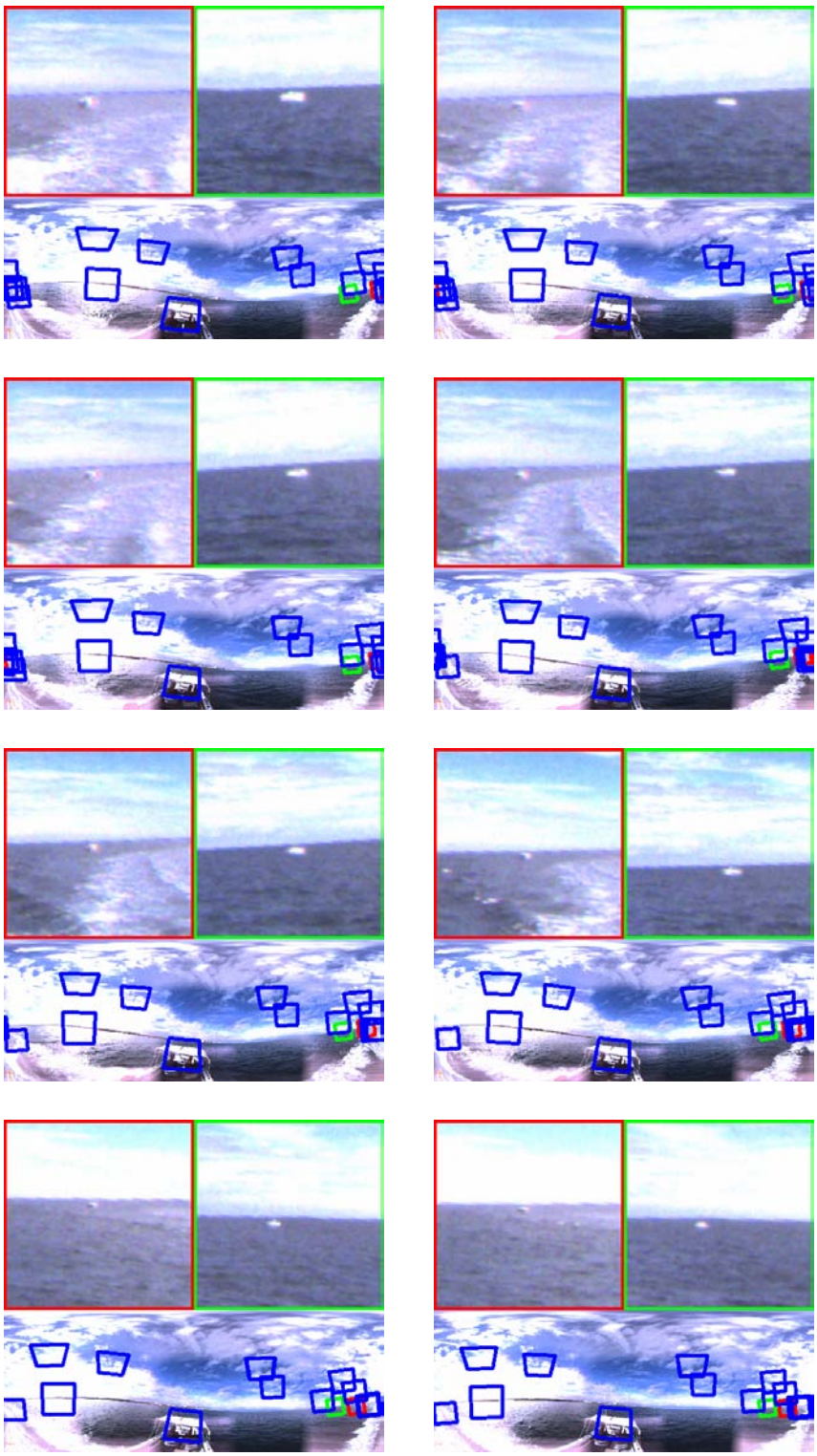


Figure 6.9: Frames 900–1600 (left to right, top to bottom) showing the raw omnidirectional view together with two extracted virtual cameras fixed on targets (red and green). The omnidirectional view also shows a number of false positives being tracked by other virtual cameras (continued from previous page).

previous chapters. Specifically, these datasets were analysed for object placement, count and relative size of objects. It was found that the *MSRA* saliency dataset is insufficient to reflect real-world conditions with respect to the maritime domain. Subsequently, *shipspotting*, the dataset that was compiled for the purposes of this thesis, was found to provide better testing environment as it better represented the situation that would occur in a real world maritime environment.

In the following section, experiments on actual maritime omnidirectional imagery showed that the proposed visual attention framework cannot be deployed on the camera system in a real maritime environment despite the promising results from the dataset as shown in Chapters 4 and 5, as the visual attention framework produces to many false positives in the high resolution omnidirectional image. However, an encouraging aspect of the results is that in particular two objects of most importance were detected even though they were very small and moving.

From the optimal thresholded visual attention map, the dominant components were used to initialise the tracking algorithm provided by the stabilisation framework. It was shown that the subsequent stabilisation using the virtual camera approach is very effective in challenging conditions both on targets and background blobs.

Chapter 7

Conclusion

The use of a maritime surveillance platform allows coverage in hazardous and hostile environments without the need to put people to risk. This thesis dealt with the computer vision aspects of the platform. The camera system is designed to aid the operator in the first instance, and later act autonomously. In particular, this thesis explored the challenging outdoor conditions of the maritime domain and proposed an image stabilisation technique that allows for stabilised tracking of target objects. Furthermore, a visual attention framework was proposed that is capable of directing attention towards regions of interest with respect to the maritime domain. Subsequently, a novel tracking method was presented that is capable of tracking multiple targets simultaneously in omnidirectional imagery by using one virtual camera for each target.

In Chapter 3, the combination of an omnidirectional camera and an IMU using a probabilistic sensor fusion approach was proposed. A probabilistic model was utilised to allow for loose calibration and synchronisation of the hardware components. This allows for dynamic and quick assembly of off-the-shelf equipment without the need for re-calibration. The advantage of an omnidirectional camera is apparent as it allows for an instantaneous full spherical view, which is essential for full situational awareness. However, for closer inspection of a target object only a small field of view is required, therefore the use of a virtual camera was proposed that extracts a limited field of view from the omnidirectional image. The virtual camera is then used to provide a target-centric stabilisation by adjusting the virtual camera according to the platform's ego-motion. For this, the measurements of the IMU and an image feature tracker were combined. Experiments showed that the framework provides robust stabilisation towards a target object while the camera is subjected to significant rotational and translational disturbances.

The virtual cameras for image stabilisation have to be initialised manually, a shortcoming that was addressed in Chapter 4. In this chapter, a visual attention framework was proposed that is capable of directing attention to areas of interest. For this purposes, multiple multi-scale low-level features such as edges, texture, and colour information were extracted and evaluated using local, regional, and global distance measurements (locality

cues). These features and locality cues were extensively explored individually to determine their characteristics before being considered in combination. The features were eventually fed into a Bayesian classifier to compute probability maps that indicate the presence of a maritime object. The visual attention framework was subsequently evaluated using a standard dataset (*MSRA*) and later a dataset with maritime imagery that was compiled for the purposes of this thesis (*shipspotting*). The approach was compared to related detectors and was found to give reasonable results on the generic dataset and outperform existing approaches on the *shipspotting* dataset.

In Chapter 5, the proposed visual attention framework was further improved by making use of domain specific knowledge of the background. Here, the dominant background in a maritime environment (sea and sky) was examined in terms of colour and edge orientation and it was found that both sea and sky mostly consist of the same primary colour that can be expressed in Hue coordinates. Furthermore, the shape of waves was found to be sufficiently different due to dominant horizontal directions and was able to provide a reliable cue for detection. A sea/sky detector was proposed and fused into the existing visual attention framework. A subsequent feature selection analysis provided information about the importance of each feature and allowed the reduction of the feature space without compromising classification accuracy. The improved detector was evaluated using the *shipspotting* dataset and found to outperform generic approaches.

Chapter 6 began with an analysis of the evaluation datasets. In Chapter 4, it was found that the evaluated generic saliency detectors performed worse on domain specific dataset (*shipspotting*) than on the generic *MSRA* dataset. Both datasets were examined for placement of the objects within the image, count of objects, and the relative size of an object in an image. It was found that *MSRA* is in fact a dataset with limited diversity in terms of these properties. The *shipspotting* dataset, on the other hand, provided a more challenging task due to its higher variety in object placement, higher amount of object counts per image, and much smaller objects. The chapter continued with the deployment of the camera system on a maritime platform in real-world conditions. It was shown that despite the performance of the visual attention framework on the two benchmark datasets, the omnidirectional image is far more challenging and no detectors were able to produce reasonable results, with many false positive being detected by all methods. However, the experimental investigation revealed that the proposed framework was the only approach able to successfully detect and isolate the target maritime objects in the omnidirectional view and with fewer false positives. The subsequent multi target tracking was found to be very effective even in challenging conditions due to the sensor fused stabilisation framework. However, due to the high number of false positives generated by the visual attention framework, it cannot be seen as a sufficient means for target detection in omnidirectional

imagery of the maritime domain and requires further investigation and refinement.

7.1 Future Work

The stabilisation framework developed in Chapter 3 uses a virtual camera to keep a target object in view regardless of the ego-motion of the camera. The virtual camera has three parameters: the orientation with respect to the camera coordinate system, the field of view, and the resolution. While Chapters 4 and 5 proposed a visual attention framework that has the potential to provide an auto-initialisation of the orientation of the virtual camera, the field of view and resolution were manually selected for the experiments conducted in this thesis. However, depending on the situation, an intelligent selection of these parameters could be performed based on confidence maps that not only can be used to estimate the location of a region of interest but also their spatial extent. This information can be used to compute the optimum field of view of the virtual camera. As can be seen in Figure 6.9 in the previous chapter, the target objects were far away by the end of the recording. An adaptive change of the field of view could make the field of view narrower if a target moves away and broaden the view if the target moves towards the camera, allowing the target to appear at the same size in the image at all times.

The main deficiency of the proposed visual attention framework is the high number of false positives generated in the omnidirectional view. Compared to Chapter 4, it was possible to increase the accuracy of the framework by incorporating a sea/sky detector in Chapter 5. However, when applied to real-world omnidirectional imagery, a high number of false positives were generated in typical background regions. Therefore the integration of domain specific knowledge of the background is required. With the results from Chapter 6 in mind, typical areas containing false positives were the wake caused by the platform itself, the sun and the glare it causes, and complex cloud constructs. Building detectors that specifically find the presence of such phenomena would greatly reduce the false positive rate.

The tracking approach selected in Chapter 6 was sufficient for its intended purpose. However, in case of major occlusions which can easily happen in areas with high traffic (for busy environments such as ports), the integration of a dedicated multi-target tracking approach to handle coalescence is favourable.

Finally, motivated by the analysis of the *MSRA* and *shipspotting* datasets, the compilation of more goal directed datasets is desired, and in particular, the compilation of an

omnidirectional maritime imagery dataset is recommended.

Bibliography

R. Achanta and S. Süsstrunk. Saliency detection using maximum symmetric surround. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2653–2656. IEEE, 2010.

R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

T. Albrecht, T. Tan, G. A. W. West, T. Ly, and S. Moncrieff. Vision-based attention in maritime environments. *International Conference on Information, Communications and Signal Processing*, 2011.

B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

D. Ballard and C. Brown. *Computer Vision*. Prentice Hall, 1982.

S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. Sift features tracking for video stabilization. 2007.

C.F. Bohren and A.B. Fraser. Colors of the sky. *Phys. Teach*, 23(5):267–272, 1985.

J.Y. Bouguet. Camera Calibration Toolbox for Matlab, 2004.

C.L. Braun and S.N. Smirnov. Why is water blue? *Journal of chemical education*, 70(8): 612, 1993.

J. Braun. Visual search among items of different salience: Removal of visual attention mimics a lesion in extrastriate area v4. *The Journal of Neuroscience*, 14(2):554, 1994.

P.J. Burt. Fast filter transform for image processing. *Computer graphics and image processing*, 16(1):20–51, 1981.

J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, 26(6):519, 2007. ISSN 0278-3649.

J.J. Craig. *Introduction to Robotics, Mechanics and Control*, volume 74. Pearson Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2005.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. 2005. ISSN 1063-6919.

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.

R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

D. Devarajan, Z. Cheng, and R.J. Radke. Calibrating distributed camera networks. *Proceedings of the IEEE*, 96(10):1625–1639, 2008.

A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000. ISSN 0960-3174.

HR Everett. *Sensors for mobile robots: theory and application*. AK Peters, Ltd., 1995.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.

M.D. Fairchild. *Color appearance models*, volume 3. Wiley, 2005.

O. Faugeras, Q.T. Luong, and T. Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. The MIT Press, 2004.

P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. ISSN 0920-5691.

P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2009. ISSN 0162-8828.

M. Fiala, D. Green, and G. Roth. A panoramic video and acoustic beamforming sensor for videoconferencing. In *Haptic, Audio and Visual Environments and Their Applications, 2004. HAVE 2004. Proceedings. The 3rd IEEE International Workshop on*, pages 47–52. IEEE, 2004.

L.M.G. Fonseca and BS Manjunath. Registration techniques for multisensor remotely sensed imagery. *Photogrammetric Engineering and Remote Sensing*, 62(9):1049–1056, 1996.

S. Frintrop, E. Rome, and H.I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception (TAP)*, 7(1):6, 2010.

M. Galer and L. Horvat. *Digital Imaging: Essential Skills*. Focal Press, 3rd edition, 2005.

C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical implications. *Computer Vision ECCV 2000*, pages 445–461, 2000.

E.B. Goldstein. *Sensation and perception*. Wadsworth Pub Co, 7 edition, 2007.

R.C. Gonzalez and E. Richard. *Woods, digital image processing*. Prentice Hall Press, 2002.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.

M. Handford. *Where's Wally?* Walker Books, 1987.

R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973. ISSN 0018-9472.

J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in neural information processing systems*, 19:545, 2007.

J. Heikkila and O. Silven. A Four-step Camera Calibration Procedure with Implicit Image Correction. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR'97)*, page 1106. IEEE Computer Society, 1997. ISBN 0818678224.

J.D. Hol, T.B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The International Journal of Robotics Research*, 29(2-3):231, 2010.

B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988.

X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. Ieee, 2007.

Yiqun Hu, Deepu Rajan, and Liang-Tien Chia. Detection of visual attention regions in images using robust subspace analysis. *Journal of Visual Communication and Image Representation*, 19(3):199 – 216, 2008. ISSN 1047-3203. doi: DOI:10.1016/j.jvcir.2007.11.001. URL http://www.sciencedirect.com/science/article/B6WMK-4RDR1C8-1/2/a18094db458d609aec24fa9f28c9ee49.

C. Hue, J.P. Le Cadre, and P. Perez. Sequential monte carlo methods for multiple target tracking and data fusion. *Signal Processing, IEEE Transactions on*, 50(2):309–325, 2002.

International Commission on Illumination. Colometry. Technical report, Commission Internationale de l'éclairage, 2004.

L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 1998. ISSN 0162-8828.

Omar Javed and Mubarak Shah. Tracking in multiple cameras with disjoint views. In Mubarak Shah, editor, *Automated Multi-Camera Surveillance: Algorithms and Practice*, volume 10 of *The Kluwer International Series in Video Computing*, pages 1–26. Springer US, 2008.

D.J. Jobson, Z. Rahman, and G.A. Woodell. Properties and performance of a center/surround retinex. *Image Processing, IEEE Transactions on*, 6(3):451–462, 1997.

R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(Series D):35–45, 1960.

C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–27, 1985.

R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(June):271–274, 1998.

S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

C.H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. Ieee, 2008.

A. Lawrence. *Modern inertial technology: navigation, guidance, and control*. Springer Verlag, 1998.

D.D. Lewis and W.A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales,". *Journal ofapplied statistics*, 21(2):225–270, 1994.

T. Liu, J. Sun, N.N. Zheng, X. Tang, and H.Y. Shum. Learning to detect a salient object. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

J. Lobo and J. Dias. Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1597–1608, 2003.

J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561, 2007. ISSN 0278-3649.

D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. ISSN 0920-5691.

Y.F. Ma and H.J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 374–381. ACM, 2003.

D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187, 1980. ISSN 0962-8452.

M.K. Masten. Inertially stabilized platforms for optical imaging systems. *Control Systems Magazine, IEEE*, 28(1):47 –64, 2008.

T. Mauthner, F. Fraundorfer, and H. Bischof. Region matching for omnidirectional images using virtual camera planes. In *Proc. of Computer Vision Winter Workshop*, 2006.

P.S. Maybeck. *Stochastic models, estimation and control*, volume 1. Academic Pr, 1979.

R.S. Michalski, J.G. Carbonell, and T.M. Mitchell. *Machine learning: An artificial intelligence approach*, volume 1. Morgan Kaufmann, 1985.

D. Michie, D.J. Spiegelhalter, and C.C. Taylor. Machine learning, neural and statistical classification. 1994.

F.M. Mirzaei and S.I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156, 2008.

S.K. Nayar. Catadioptric omnidirectional camera. pages 482 –488, jun. 1997. doi: 10.1109/CVPR.1997.609369.

J.M. Ogden, E.H. Adelson, J.R. Bergen, and P.J. Burt. Pyramid-based computer graphics. *RCA Engineer*, 30(5):4–15, 1985.

Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura. Visual surveillance and monitoring system using an omnidirectional video camera. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 588–592. IEEE, 1998.

CA Párraga, G. Brelstaff, T. Troscianko, and IR Moorehead. Color and luminance information in natural scenes. *JOSA A*, 15(3):563–569, 1998.

H. Pashler. Attention and visual perception: Analyzing divided attention. *An Invitation to Cognitive Science: Visual cognition*, 2:71, 1995.

H.E. Pashler. *Attention*. Psychology Pr, 1998.

H.E. Pashler. *The psychology of attention*. The MIT Press, 1999.

R.A. Peters and R.N. Strickland. Image complexity metrics for automatic target recognizers. In *Proceedings of the Automatic Target Recognizer System and Technology Conference*, pages 1–17. Citeseer, 1990.

R.M. Pope and E.S. Fry. Absorption spectrum (380–700 nm) of pure water. ii. integrating cavity measurements. *Applied Optics*, 36(33):8710–8723, 1997.

Paul Rosin. A simple method for detecting salient regions. *Pattern Recognition*, 42(11): 2363–2371, 2009. ISSN 00313203. doi: 10.1016/j.patcog.2009.04.021.

S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice hall, 2010.

D. Salomon. *Transformations and projections in computer graphics*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.

T. Sato, S. Ikeda, and N. Yokoya. Extrinsic camera parameter recovery from multiple image sequences captured by an omni-directional multi-camera system. *Computer Vision-ECCV 2004*, pages 326–340, 2004.

D. Schneider, E. Schwalbe, and H.-G. Maas. Validation of geometric models for fisheye lenses. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(3):259 – 266, 2009. ISSN 0924-2716. doi: DOI:10.1016/j.isprsjprs.2009.01.001. Theme Issue: Image Analysis and Image Engineering in Close Range Photogrammetry.

K. Seo, J. Ko, I. Ahn, and C. Kim. An intelligent display scheme of soccer video on mobile devices. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17 (10):1395–1401, 2007.

C.E. Shannon and W. Weaver. *The mathematical theory of communication*, volume 19. University of Illinois Press Urbana, 1962.

J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

S.N. Sinha and M. Pollefeys. Pan-tilt-zoom camera calibration and high-resolution mosaic generation. *Computer Vision and Image Understanding*, 103(3):170–183, 2006.

D. Slater. Panoramic Photography with Fisheye Lenses,© 1995. *Published in the IAPP Journal*, 1996.

G.S. Smith. Human color vision and the unsaturated blue color of the daytime sky. *American journal of physics*, 73:590, 2005.

J.P. Snyder. Map projections: a working manual. *US Geological Survey professional paper*, 1395, 1987.

C. Soto, Bi Song, and A.K. Roy-Chowdhury. Distributed multi-target tracking in a self-configuring camera network. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1486–1493, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/CVPRW.2009.5206773.

X. Sun, J. Foote, D. Kimber, and BS Manjunath. Region of interest extraction and virtual camera control based on panoramic video capturing. *Multimedia, IEEE Transactions on*, 7(5):981–990, 2005.

Y. Sun and R. Fisher. Object-based visual attention for computer vision. *Artificial Intelligence*, 146(1):77–123, 2003. ISSN 0004-3702.

T. Svoboda and T. Pajdla. Panoramic cameras for 3D computation. In *Proceedings of the Czech Pattern Recognition Workshop*, pages 63–70, 2000.

US Department of Defense. World Geodetic System. Technical report, National Imagery and Mapping Agency, 2000. Third Edition.

Y. Utsumi, Y. Iwai, and H. Ishiguro. Face tracking by using omnidirectional sensor network. pages 2172 –2179, 2009.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

Y. Yagi. Omnidirectional Sensing and Its Applications. *IEICE Transactions on Information and Systems*, 1999.

Junlan Yang, D. Schonfeld, and M. Mohamed. Robust video stabilization based on particle filter tracking of projected camera motion. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(7):945 –954, 2009.

A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(11):1531–1536, 2004.

Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38, December 2006. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/1177352.1177355. URL http://doi.acm.org/10.1145/1177352.1177355.

H. Zhang, J.E. Fritts, and S.A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

Z. Zhou, B. Niu, C. Ke, and W. Wu. Static object tracking in road panoramic videos. In *2010 IEEE International Symposium on Multimedia*, pages 57–64. IEEE, 2010.