ISBN 978-82-326-4252-6 (printed version)
ISBN 978-82-326-4253-3 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:328

Christopher Dahlin Rodin

# Applications of High-Precision Optical Imaging Systems for Small Unmanned Aerial Systems in Maritime Environments

Doctoral Thesis

Christopher Dahlin Rodin

Doctoral theses at NTNU, 2019:328

NTNU
Norwegian University of
Science and Technology

**NTNU**
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU

Christopher Dahlin Rodin

# Applications of High-Precision Optical Imaging Systems for Small Unmanned Aerial Systems in Maritime Environments

Thesis for the degree of Philosophiae Doctor

Trondheim, December 2019

Norwegian University of Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

**NTNU**
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics



ISBN 978-82-326-4252-6 (printed version)
ISBN 978-82-326-4253-3 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2019:328


Printed by Skipnes Kommunikasjon as

*To Ulla, Yalin, Pia and Shuifang.*

# Summary

Enabling technologies such as more powerful and miniaturized embedded computers, sensors, and power systems have rendered small Unmanned Aerial Systems (sUAS) cheap and readily available for industry, consumers and university departments and enterprises of all sizes. Videos and photographs captured by cameras attached to sUAS have become a common sight in video and photo sharing websites, corporate offerings, as well as in scientific journals. While the purpose of the first group of users often is to take beautiful photos, the latter two user groups often use the camera attached to their sUAS as a sensor to either navigate the sUAS, measure certain parameters, or detect objects in the physical world. For these types of missions, either a visible-light camera or a camera in the infrared or other bands can be used, e.g. thermal infrared cameras which often prove to be good at detecting people and animals, as well as oil spills.

This thesis focuses on evaluating and improving upon the methods used when the camera system is being used as a tool to geometrically measure the environment and detect objects. Initially in the thesis, image quality degradation caused by external movements and vibrations are discussed. The source of the movements are discussed, along with suggested solutions on how to mitigate the image degradation. This includes both camera and optics design parameters (pixel size, optics or camera sensors with built-in optical image stabilization), as well as active and passive vibration dampening (electric motors, passive dampeners), software image and video stabilization, and aerodynamic design of the camera stabilization platform.

For evaluating and improving on the geometrical measurements, an algorithm for producing a highly accurate camera attitude estimate – the orientation of the camera in the geographic coordinate system – is developed. The reason for this focus is the large error an incorrect camera attitude estimate can introduce in geometrical measurements. The algorithm matches a horizon detected in a camera image with a horizon synthetically generated using a Digital Surface Model. Working in close to real-time, the algorithm's accuracy exceeds that of similar methods while performing global search in the yaw angle – commonly the angle that is most difficult to

estimate. Camera images were later acquired in a real-world mission taking place in the Bothnian bay in Northen Europe. A camera was attached to a moored balloon system with the purpose of tracking and estimating the size of ice-floes floating into a marine vessel. The attitude estimation algorithm was used on the camera images with the purpose of estimating the dimensions of the ice-floes, and proved to produce highly accurate shape estimates even at very slant camera angles.

The final focus point of the thesis is on object detection and classification, which is a common task for camera systems mounted on sUAS, for example in search-and-rescue missions at the sea. In order to evaluate the camera performance, a mission was performed where various objects were placed at the sea surface. The objects were selected as common objects present in search-and-rescue missions, e.g. a human, pallets, buoys, and a small boat. Images of the objects were taken with both a visible-light camera and a thermal infrared camera. The detectability of the objects was first evaluated using common image processing algorithms like edge detection. This showed the large effect that the scene can have on such tasks, with noise from ocean waves causing difficulties in detecting objects smaller than 3 pixels in the image plane. A study on classification of the objects are finally performed. This uses the the thermal camera image dataset from the aforementioned mission together with a Gaussian Mixture Model for image segmentation, and finally a Convolutional Neural Network for object classification. The conclusion is that object classification is possible with an acceptable accuracy, but that the low pixel count of objects in the image plane is a limiting factor.

# Contents

# Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Philosophiae Doctor (PhD) at the Norwegian University of Science and Technology (NTNU). The research has been carried out between the summer of 2015 and the summer of 2018 at the Centre for Autonomous Marine Operations (NTNU-AMOS, grant no. 223254), the Department of Engineering Cybernetics at NTNU, and at Maritime Robotics AS. The work has been supervised by Professor Tor Arne Johansen. As a PhD candidate, I was an Early Stage Researcher (ESR) in the Innovative Training Network on Autonomous Unmanned Aerial Systems for Marine and Coastal Monitoring (MarineUAS), which was funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 642153.


## Acknowledgments

The research presented in this thesis could not have been done without the support and involvement of many.

I would like to thank my supervisor Tor Arne Johansen, who has shown great patience and provided invaluable support in both my research and other areas of my PhD candidacy, and has always encouraged me to follow my research path. I would also like to thank the department of Engineering Cybernetics, the faculty of Information Technology and Electrical Engineering, and the Norwegian University of Science and Technology for providing an excellent and supportive research environment, and a friendly and creative atmosphere.

As an industrial PhD candidate, I was not only part of a great academic environment, but also a great engineering environment at Maritime Robotics AS in Trondheim, Norway. I would like to thank my supervisors there, Vegard Evjen Hovstein and Arild Hepsø, as well as my other colleagues who provided great discussions on the practical use of the research I was performing. In addition, I would like to thank Equinor and the UAV-Lab at NTNU for providing me with the op-

portunity to work with data from missions and experiments organized by them, as well as all the interesting discussions I had with them regarding data, sensors, and project execution.

I would also like to show my gratitude towards the European Union – being part of an Innovative Training Network was an invaluable experience that did not only help me improve as a researcher, but in all professional aspects. For this experience I would also like to thank all the lecturers, organizers, and fellow early stage researchers involved in the Marine UAS programme.

During my studies, much of my research has also been done together with Fabio Augusto de Alcantara Andrade and Anthony Reinier Hovenburg. I would not have enjoyed my life, both during and after my PhD, as much as I have without you.

I would like to thank friends whom I did not meet during my PhD, but have stuck with me throughout. Most notably are Joachim Jonsson and Marcus Andersson, who I have shared many great moments with both prior to and during my PhD.

To my family – my brothers Philip and Jonathan, my mother Pia – as well as my extended family in Shangyu – Yalin and Shuirong. You have supported and encouraged me to pursue my own path in more ways than you know.

Finally I would like to thank Shuifang, my life partner and best friend, who selflessly supports me in whatever I decide to pursue. You always provide me with an oasis of happiness, good food, and understanding.

Thank you all.

- Christopher Dahlin Rodin

# Chapter 1

# Introduction

This chapter introduces the motivation for the research included in this thesis. The general scope of the thesis and its content are briefly stated, as well as its structure. Finally, the chapter contains a list of the articles written and published during the duration of the PhD programme, some of which forms the basis for this thesis.

## 1.1 Background and Motivation

In the recent decades, Unmanned Aerial Vehicles (UAV) technology has transitioned from being expensive, high-tech, and reserved for military applications and larger enterprises, to a readily available and affordable technology. UAV technology is today being used by hobbyists, companies, and research organization in their daily operations and projects. Some of the main reasons for their popularity is their ability to transport smaller packages, as well as to collect valuable data in vast amounts, while having the potential of being both much safer and require far less resources than the alternatives such as road transport and manned aviation.

When the data is the main reason for the UAV mission, the data acquisition is commonly done by the use of remote sensing – using sensors to capture data at a distance from the observed target. The alternative to this would be to e.g. drag a sensor through a body of water to acquire data about the water quality. One example of a remote sensing mission is the Station Keeping Trails (SKT) performed by Equinor (formerly Statoil) [86] [1]. In this mission, the effects of ice-floes on marine vessels were analyzed in the Bothnian bay. For this, the size and position of the ice-floes were important information in the quantitative analyses. During the trials, this information was gathered by a camera attached to a moored balloon system. Other missions and research projects include precision agriculture [89] [60] [148] where multi- and hyperspectral cameras can be used to analyze various health

factors of crops, and e.g. apply the required pesticide of the right amount of water; as well as two- and three-dimensional mapping and modelling missions [112] [79] [81].

The missions mentioned above mostly use a optical imaging sensors (cameras) as the main sensors. While many examples also exist where other sensors are used, e.g. radars, lidars, and air quality sensors, camera systems remain one of the most commonly used sensor due to their relatively low cost, low weight, and ease of understanding the data they produce. This makes the camera an interesting and important sensor to analyze more in-depth in order to get a better understanding of how it performs for the various types of data that are commonly sought the be extracted from camera images. The camera performance can be divided into a multitude of sub-fields, such as optics and imaging sensor performance, read-out circuit performance, as well as light source and scene understanding and modeling. Each of these sub-fields can further be broken down, and have been analyzed in great depth, both in specific books (e.g. [6]) and in more general handbooks, e.g. [77].

By taking one step from the theoretical foundation of the different sub-fields of imaging towards the end-users, i.e. photographers and remote sensing data producers and users, much research has also been produced and published. Early and impactful research include Johnson's criteria [119], which analyzed how well objects can be detected in thermal camera images. Later on, in e.g. [37] [80] [79] [25] [123], many camera systems have been built and had their performance evaluated. These articles serve as good hands-on showcases of how a camera system can perform, but often leaves out details on how the camera and other components are selected. On the other side is research like [58] [94] and the aforementioned research on imaging components, which presents theoretical surveys and guidelines on how to design high end stabilized camera system. A designer skilled in the art might be able to draw the full benefit of those kind of articles. However, literature to aid non-imaging experts in the design and component selection for stabilized camera systems for sUAS is missing in the field.

The aim of this thesis is thus to survey current literature, and gather data from camera systems, in order to aid non-imaging experts in the design, component selection, and evaluation of stabilized camera systems and to document the performance of common tasks performed by such systems with a focus on remote sensing.

### 1.1.1 The Problem of Georeferencing Accuracy

Georeferencing is the process of relating information to a geographic coordinate system [59]. For sensors, this often means mapping data from the sensor coordinate system to the geographic one. In the case of e.g. time-of-flight radar sensors, the mapping is between two three-dimensional coordinate systems. For camera systems it is commonly a two-dimensional to three-dimensional coordinate mapping, which require additional knowledge of the scene. In addition to this third dimension, the pose (position and orientation) of the sensor coordinate system in the geographic coordinate system needs to be known. This information is commonly acquired from a Global Navigation Satellite System (GNSS), magnetometer, and Inertial Measurement Unit (IMU). However, there are two problems that are commonly encountered when using external sensors such as GNSS and IMU to determine the sensor pose. The first is that the pose of the imaging sensor coordinate system in relation to the GNSS and IMU coordinate systems need to be known; And the second is that GNSS and IMU in many cases lack the required accuracy due to e.g. disturbances in the magnetic field from other magnetic fields caused by e.g. large metal structures or electric motors, causing the heading accuracy to reach tens of degrees. These problems can cause e.g. georeferencing errors of hundreds of meters in a common UAV mission.

The camera system itself has a very high spatial accuracy under certain circumstances (e.g. when using lens with low or well defined distortion). This makes it an attractive option to relate the camera coordinate system directly with the geographic coordinate system. This requires part of the scene to be both spatially well defined (its geographic coordinates known), and be detected within the camera image. With the geographic data being readily available, as well as recent advancements in object detection and image segmentation using low-cost computers mountable on-board the UAV, using the camera-based pose estimation methods are becoming more and more feasible. Some problems still remain, e.g. when no pre-defined scene objects are detected in the camera image due to blocking objects or fog. This could however potentially be solved with e.g. dead-reckoning estimation of the camera pose using the lower accuracy GNSS and IMU sensors.

### 1.1.2 The Problem of Scene Complexity

The utilization of measuring objects in a camera image can be greatly enhanced if the objects can be automatically detected in the scene first. Object detection also has many other uses where it is the main purpose, e.g. search-and-rescue missions. Knowing beforehand whether the known or unknown objects can be detected by the selected camera at the UAV mission trajectory is vital, but can be highly complex to calculate or simulate. In some controlled environments, a priori information about

the scene and objects can be acquired to select a suitable camera. In this type of missions, the scene and objects can also be manipulated in order to make object detection easier. Quick Response (QR) codes can be attached to boxes, and their corners can be marked with a known pattern so that the intelligent sensor platform knows what pattern to scan for. In a natural scene it can sometimes be done using landmarks or other markers to navigate or to limit the area where the scan should be performed. In many cases, in particular in remote sensing or search-and-rescue missions taking place in large, natural areas, it is not feasible to rely on a priori information or environmental manipulation to facilitate easier object detection.

There are some methods for estimating how well objects can be detected in a real world mission before the mission takes place. One such method is camera system and scene simulation. In simulations, camera noise models (e.g. as described in [40]) can be used together with artificially limiting the image resolution to introduce the imaging sensor noise and limitations in the simulation. For simulating the performance of the optics, a model such as the Modulation Transfer Function (MTF) [14] can be used. However, the data required for using the MTF for simulations can be difficult to obtain – the lens manufacturer rarely makes this available, so experiments would have to be performed to determined this for the lenses to test. Simulating the scene could also become complicated due to the different absorption and reflectance characteristics of the different materials. Modern game engines are doing a good job of producing photo-realistic image sequences of a wide variety of scenes, but they may contain effects which are meant to be pleasing to the eyes rather than imitating the real world. They can also be time-consuming to set up to imitate the planned mission.

An alternative to simulate the optical imaging system and the scene is to do an in-depth field study of real images. This can become an iterative process, and can be used with the simulation methods mentioned above by e.g. starting off with a high quality camera system and then simulating noise present in lower quality camera systems until the right price-weight-quality balance have been found. In addition to using real world data to select the right camera for a future mission, it is also important to gather intelligence on how well the camera performs in detecting objects using both traditional image processing and computer vision algorithms, as well as modern machine learning methods.

## 1.2 Structure of the Thesis and Main Contribution

The remainder of this thesis is divided into seven chapters, transitioning from design considerations for components surrounding the camera, to analyses of the camera system itself in the form of georeferencing and object detection applications. The

final chapter discusses the main conclusions of the research, and gives suggestion for future work.

- **Chapter 2:** Previous research in camera stabilization systems has been focused on either theory of high-precision pointing and stabilization systems (e.g [58] and [94]), or evaluating custom-made optical imaging systems (e.g. [37], [80], [79], [25], [123]). Research in these areas can be very helpful in designing certain aspects of high-performance stabilization platforms, but does not satisfactory provide an overview of the more practical design considerations required when addressing image degradation caused by vibrations of the camera system. This chapter contains a survey of practical design considerations of camera stabilization systems, with a focus on cameras mounted on sUAS. The stabilization aims to mitigate blur caused by vibrations present in the sUAS either by mechanically dampen or counteract them, by using a camera with optical image stabilization, or by using software methods e.g. image and video stabilization. The chapter presents the most common causes for vibrations, and by extension also blur, as well as the most common and recent methods for reducing the effects of the vibrations.
- **Chapter 3:** A number of methods exist for estimating the attitude using camera images: [51] [34] [47]. However, the important criteria of running in real-time (or near real-time) on a computer small enough to be part of the payload of an sUAS while also estimating the yaw angle are not fulfilled in pre-existing research. This chapter presents a method for estimating the attitude of the camera by comparing the horizon detected in the camera image with a synthetic horizon. The camera horizon is found using a simple flood-fill algorithm, while the synthetic horizon is generated from a Digital Surface Model (DSM). The algorithm uses grid-search to find the camera attitude, and performs a global yaw angle estimate while running in near real-time. The algorithm is finally evaluated using an experimental dataset.
- **Chapter 4:** Estimating the size of ice-floes has been touched upon in previous research, e.g. [90]. The focus has however mostly been on evaluating algorithms and analyzing the ice-floe size distributions, rather than an evaluation of how well the size can be estimated in different camera configurations. In this chapter, data from a mission in the Bothnian bay was used to evaluate the size estimate of ice-floes from camera images. The ice-floes were floating towards a marine vessel, and images of them were captured with a visible-light camera attached to a moored balloon system. The algorithm from the previous chapter was in part used to estimate the camera attitude in order to reduce the errors caused by an incorrect camera attitude estimate. Spe-

cial focus was on the relation between the accuracy and the camera-to-object (ice-floe) angle.

- **Chapter 5:** Detectability of objects in camera images have been studied extensively, with perhaps the most well known research being done by John Johnson several decades ago [119]. However, with the scene having a major effect in detectability of objects in camera images both for humans and in computer vision algorithms, it is important to further study the detectability of objects in different scenes to being able to better predict how well something can be detected. This chapter evaluates the detectability objects at the sea surface in camera images. A mission was performed where objects were placed at the sea surface, where the objects were chosen as common objects present in search-and-rescue missions at the sea. Both a visible-light and thermal infrared camera was used in order to showcase two common cameras used in this kind of mission. Simple edge detection algorithms were used to evaluate the detectability of the objects compared to the background of the images as opposed to using multi-stage object detection and segmentation algorithms, in an attempt of reducing the algorithm-dependability of the results.
- **Chapter 6:** Object classification of high resolution objects in visible light cameras is a widely explored topic, with machine learning algorithms in many cases out-performing humans. Classification of objects in low resolution thermal camera images is a less explored area, but has great importance in e.g. search-and-rescue missions. Some research has been done on this, e.g. [79]. However, the methods used were traditional image processing and computer vision algorithns rather than the more modern and higher performing deep neural networks. This chapter aims to evaluate the object classification performance, using the thermal infrared dataset presented in the previous chapter. To segment the objects (the foreground) from the background, a Gaussian Mixture Model was used. For the found objects, a Convolutional Neural Network (CNN) was trained on part of the dataset, and evaluated on another part. The CNN used multiple layers, e.g. two convolutional and two max-pooling layers in addition to the fully connected layers.
- **Chapter 7:** This chapter contains the final remarks regarding the research contained in the thesis as a whole. It also contains suggestions for future work considered to be important research topics of the field in the future.

## 1.3 Publications

This thesis is based on research articles published in peer-reviewed conferences and journals. Below are listed the publications included in the main body of this thesis;

publications included as appendices; and publications not included in this thesis due to their irrelevance to camera systems.

**Publications included in the thesis:**

- [114] C. Dahlin Rodin and T. A. Johansen. Accuracy of sea ice floe size observation from an aerial camera at slant angles. In Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Linköping, 2017.
- [117] C. Dahlin Rodin, T. A. Johansen, and A. Stahl. Skyline based camera attitude estimation using a digital surface model. In IEEE 15th International Workshop on Advanced Motion Control (AMC), Tokyo, 2018.
- [116] C. Dahlin Rodin, L. N. de Lima, F. A. de Alcantara Andrade, D. B. Haddad, T. A. Johansen, and R. Storvold. Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. In 2018 International Joint Conference on Neural Networks(IJCNN), Rio de Janeiro, 2018.
- [115] C. Dahlin Rodin and T. A. Johansen. Detectability of objects at the sea surface in visible light and thermal camera images. In MTS/IEEE-OCEANS, Kobe, 2018.
- [28] C. Dahlin Rodin, F. A. A. Andrade, A. R. Hovenburg and T. A. Johansen. A Survey of Practical Design Considerations of Optical Imaging Stabilization Systems for Small Unmanned Aerial Systems. In Sensors, 2019.

**Publications included as appendices:**

- [30] F. A. A. Andrade, A. R. Hovenburg, L. N. Lima, C. Dahlin Rodin, T. A. Johansen, R. Storvold, C. A. B. Correia, and D. B. Haddad. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. Sensors, 2019.

**Other related publication not included in the thesis:**

- [62] A. R. Hovenburg, F. A. de Alcantara Andrade, C. Dahlin Rodin, T. A. Johansen, and R. Storvold. Contingency path planning for hybrid-electric UAS. In Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Linköping, 2017.

- [29] F. A. de Alcantara Andrade, C. Dahlin Rodin, A. R. Hovenburg, T. A. Johansen, and R. Storvold. Path planning of multi-UAS communication relay by decentralized MPC. In MTS/IEEE-OCEANS, Kobe, 2018.
- [64] A. R. Hovenburg, F. A. de Alcantara Andrade, C. Dahlin Rodin, T. A. Johansen, and R. Storvold. Inclusion of horizontal wind maps in path planning optimization of UAS. In International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, 2018.
- [61] A. R. Hovenburg, F. A. A. Andrade, R. Hann, C. Dahlin Rodin, T. A. Johansen, and R. Storvold. Long range path planning using an aircraft performance model for battery powered sUAS equipped with icing protection system. Submitted for publication.

**Collaborations and contributions:**

The work done to perform the research and writing for the articles cited as being part of this thesis were all conceptualized, drafted, modified, and reviewed by more than one author. Below are detailed descriptions of the contributions by me (Christopher Dahlin Rodin) for each article. In general, my focus has been on the camera system design and theory, e.g. camera models, performance, and applications.

For [28] [117] [114] [115] [116] I contributed with part of the conceptual design, literature review, methodology, system development, investigation, writing the paper draft and administrating the project.

For [30] my work was limited to contributing with part of the conceptual design and investigation.

# Chapter 2

# A Survey of Practical Design Considerations of Optical Imaging Stabilization Systems for Small Unmanned Aerial Systems

Optical imaging systems are one of the most common sensors used for collecting data with small Unmanned Aerial Systems (sUAS). Plenty of research exist which present custom made optical imaging systems for specific missions. However, the research commonly leaves out the explanation of design parameters and considerations taken during the design of the optical imaging system, specially the image stabilization strategy used, which is a significant issue in sUAS imaging missions. This chapter surveys useful methodologies for designing a stabilized optical imaging system by presenting an overview of the important aspects that must be addressed in the designing phase and which tools and techniques are available and should be chosen according to the design requirements.

## 2.1 Introduction

The use of small Unmanned Aerial Systems (sUAS) has increased rapidly during the last years. While some hobby users operate sUAS without any particular purpose, research institutes and corporations commonly operate sUAS for the purpose of collecting information about the environment. Environmental data that can be of interest includes readings of the Earth's magnetic field, elevation data acquired by a LiDAR, and various wavelengths of transmitted and reflected light acquired by a camera system.

Optical imaging systems are one of the most common sensors used for collecting data with sUAS, and their images can be used to take snapshots of beautiful scenery, provide situational awareness to an operator during a mission, and in various photogrammetry applications. Several commercial off-the-shelf optical imaging systems for sUAS exist, in particular, for missions with high requirements on the visual appeal of the images rather than the accuracy of the world time, position, and attitude of the camera when the image was acquired. This is often the case for TV and video production, photographers, and hobby users.

Image meta data such as world time, position, and attitude are, however, of high importance in many sUAS missions performed by corporations and research institutes. With the fast development of new sensors and methods to improve the accuracy and decrease the cost of acquiring this meta data, together with a lack of off-the-shelf customizable imaging systems to test the latest technology with, many have created their own payloads. Plenty of research exists which present custom-made optical imaging systems for specific missions [37] [80] [79] [25] [123]. The research shows proofs of concept, which can be very valuable since the image acquisition process is highly complex - experimental data from multiple systems can show how design parameters affect the data quality in general, or it can be used to replicate the system for a similar mission. However, the research commonly leaves out the explanation of design parameters and considerations taken during the design of the optical imaging system, especially the image stabilization strategy used, which is a significant issue in sUAS imaging missions. On the other side of the spectrum, there is research explaining intricate details of the design of inertially stabilized platforms [58] [94].

This chapter aims to contribute with filling this gap in the literature by providing useful guidelines and methodologies for designing a stabilized optical imaging system. However, it does not consider more advanced topics for making a finished product such as structural analysis, or broad, non-specific topics such as software design. Therefore, the goal of this article is to give an overview of the important aspects that must be addressed when designing a stabilized optical imaging system and which tools and techniques are available and should be chosen according to the design requirements.

This chapter is structured as following. In the beginning of the next section, the main vibration sources are discussed, and the main techniques used to identify and evaluate them, as well as their effect on image quality. As there are different ways to administer the vibration issue, the main techniques are presented in the following sections. First, the mechanical installation of dampers is presented, followed by optical image stabilization and software solutions. Finally, the next sections are

focused on gimbal design, including the design considerations, followed by the impact of stabilized image systems on the sUAS aerodynamics. Therefore, in the end, the reader can make the appropriate choice of which techniques to be explored, according to the project requirements and limitations.

## 2.2 Vibration Sources and Effects

Vibration is one of the main concerns when designing sUAS camera systems: it potentially adds blur, decreasing the image quality and the potential to distinguish detail in the image, and consequently has the potential to compromise the entire mission. Therefore, it is important that the causes of vibration are understood so that the proper mitigation actions are taken.

### 2.2.1 Vibration sources in sUAS platforms

This section goes through the main sources of vibration in the two main popular designs of sUAS platforms: fixed-wing and rotary-wing small Unmanned Aerial Systems. Some sources are shared between them, such as rotors/propellers, and other sources are fairly unique to each platform (e.g. combustion engine of fixed-wing sUAS).

#### Fixed-wing platforms

In [85], with the objective to choose the most suitable sUAS for LiDAR mapping, the authors studied the vibration, capacity, reliability and stability of many sUAS platforms. With the acquired knowledge about the different platforms, they developed two sUAS especially optimized for the LiDAR mapping. With respect to the sources of vibration on fixed-wing sUAS, the authors based their theoretical analysis on a study performed by [91] about positioning errors on LiDAR systems caused by manned aircrafts platform vibration. In this study, the author enumerates four main sources of vibration on manned fixed-wing aircrafts: engine; external wind flow; internal wind flow within open cavities; and (4) airframe structural motions. Regarding the combustion engine, its noise impinges on the aircraft structures, causing vibrations mainly on the frequency of the engine's rotation speed and also on double frequency, from the reciprocating motion of the piston. The second and third main sources of vibration are both due to turbulent aerodynamic flow. One is caused by the flow over external aircraft structures and the other is caused by the flow and acoustic resonance phenomena within cavities open to the external airflow. However, according to [140], these vibrations can be considered less significant for sUAS due to lower speeds compared to the commercial and military aircraft. Finally, the fourth main source of vibration pointed by [91] is with regards to airframe structural motion caused by maneuvers, aerodynamic buffet, landing,

taxi, etc. Vibration can also be also caused by specific installed items, however, according to the authors, the effect is only locally on the surroundings of the item.

#### Rotary-wing platforms

Battery powered rotary-wing sUAS have the rotors as the main source of vibration [155] [142]. In [142], measurements were performed for three setups: motor without propeller; plastic propeller; and wooden propeller. For the first setup, where no propeller was mounted, low force levels for both radial and axial vibrations were recorded, indicating that propellers are the main sources of vibration. In that case, the frequency of vibration is related to the rotation speed of the rotors/propellers. According to [155], the second main source of vibration on rotary-wing sUAS corresponds to the vibration of the sUAS' structure, mainly the platform and extension arms. However, the frequencies of structural natural frequency vibrations depend on the sUAS structure. Reference [142] compared numerical simulations of the vibrations on a hexacopter structure using finite elements (FE) model with experimental results obtained by the impulse hammer excitation method. The results were satisfactory, achieving a vibration frequency accuracy between 0.047% and 2.852%. In addition, a third source of vibration was identified by [155], caused by the vibration of the payload, such as batteries and other weight sources located on the bottom of the sUAS.

### 2.2.2 Vibration effect on sUAS image quality

There is not a significant number of studies about the effect of sUAS vibration on image quality. Reference [83] investigated the effects of sUAS vibration on Binary Optical Elements (BOE). BOE is a diffraction imaging element and the diffraction efficiency can be impinged by the vibration of the platform, affecting the image quality. In other words, the relative position between the object point and the optical system changes by the movement of the platform, deteriorating the quality of the image. Therefore, the study simulates the effect of one dimension sinusoidal vibrations with different amplitudes and frequencies on an image with pixel size of 9 µm and integration time of 20 ms (Figure 2.1). First, different amplitudes at a constant frequency of 50 Hz are applied (Figure 2.2). For 5 µm of amplitude, no significant changes on the quality can be noticed. For 10 µm of amplitude, the image quality is affected, however, a more significant degradation is noticed on the edges of the image than on its center. For 20 µm of amplitude, the whole image gets blurred, and for 40 µm of amplitude, the image quality is heavily affected.

Figure 2.3 shows the result when the same vibration amplitude (15 µm) is applied to the image, but with different frequencies. There are not significant differences between Figure 2.3a and Figure 2.3b. The conclusion is that when a vibration pe-

Figure 2.1: Original image [83].

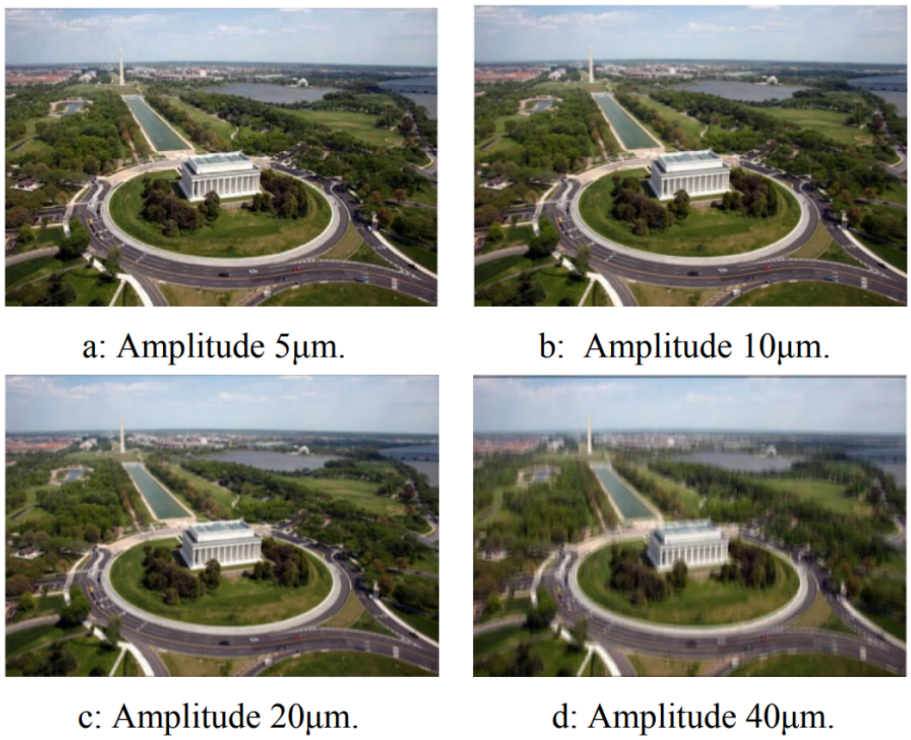


Figure 2.2: Vibration simulations with different amplitudes [83].

riod is lower than the integration time, increasing the vibration frequency has little effect on the image quality. Figure 2.3c and Figure 2.3d have the same frequency and amplitude, but the vibration was applied on different moments. This may be due to the varying speed of the camera during sinusoidal vibration. This means that, in case of low-frequency vibrations, the resulting image quality can be different for different time periods since the phase between the vibrations and the image

integration period is likely to shift randomly during a flight.

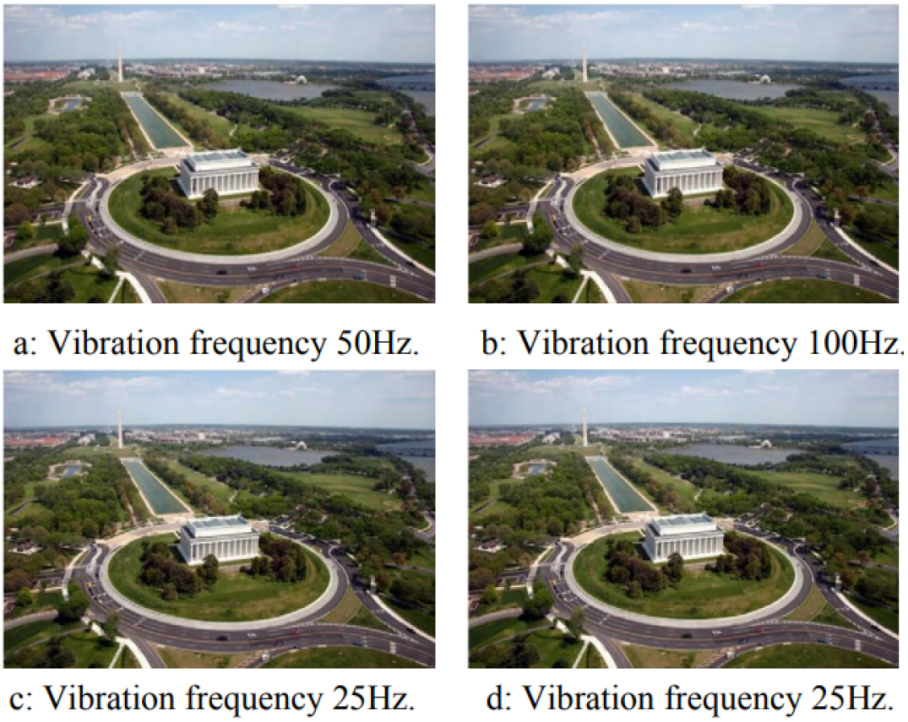


Figure 2.3: Simulation with different frequencies [83].

Based on the results, [83] proposes two modifications on the optical system to reduce the effect of vibration on the image quality. As small amplitudes of vibration cause minor effects, one way to reduce the degradation of image quality is to reduce the focal length and increase the pixel size. However, this will result in a decrease in the angular resolution of the optical system. Another way is to use a CCD (charge-coupled device) with a shorter integration time, so that the vibration period is lower than the integration time.

Other studies that investigated the effect of sUAS vibration on image quality were performed by [52] and [53]. In the latter, the study was not focusing on vibration but also investigating other aspects such as lens calibration, orthorectification and mapping. Both studies worked with images taken on the same research mission and by a Point Grey Research "Flea" camera with a Fujinon YV 2.2 × 1.4 A-2 fish-eye lens mounted both on a sUAS and on a manned aircraft (Cessna 172). In the sUAS setup, the camera was mounted on a small supporting platform that was isolated from vibrations using a special anti-shock material. In the manned aircraft, the camera was mounted on a simple mounting bracket. The main problem

caused by the aircraft vibration was the movement of the lens relative position to the camera (Figure 2.4). The authors concluded that the non-interference fit between the camera and the lens housing is responsible for the vibration effect. Also, the problem of vibration is more noticeable on the sUAS, because of its small size, making it more susceptible to maneuvers and turbulence. The comparison between the roll angle noise of the sUAS and the manned aircraft can be seen in Figure 2.5. The sUAS has variations approximately twice of the variations on the Cessna 172. In [52], where the focus of the investigation was on vibration effects and compensation, the vibration was divided into rotation and translational vibrations. The largest translational movement detected was ±5 pixels measured from randomly selected images. That is a big issue since vibrations causing only a one-pixel shift in a fisheye image captured by a sUAS operating at an altitude of 1000 ft above the ground would result in a displacement of approximately 2.5 m, using a 0.8 megapixel resolution camera.

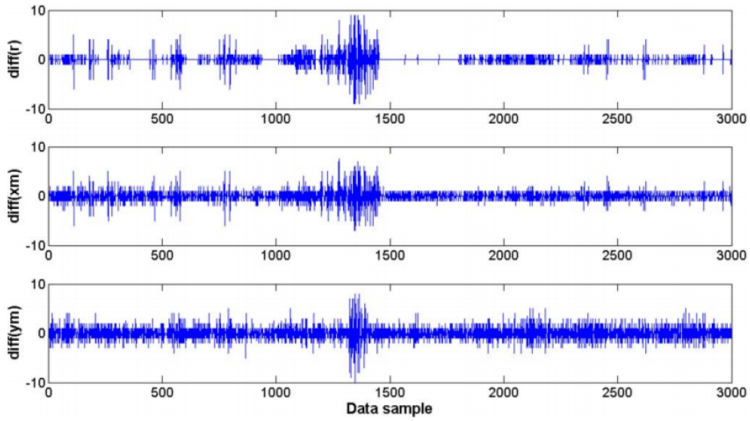


Figure 2.4: Estimated displacement of the center of the fish-eye image in pixels (xm,ym) and the estimated radius of the entire fish-eye lens image circle boundary (r) [53].

The available literature about sUAS vibration and image quality is very limited. However, there are a few studies about the effect of mechanical platform vibration on satellite imaging, such as in [54]. In this case, vibrations limit the maximum resolution and performance of remote sensing and are caused by turbines, motors, reaction wheels, actuators etc. The study performed by [54] was based on [147], which analyzed the relation between blur, vibration, exposure time and resolution, with focus on vehicular or airborne imaging systems and in robotic systems. The calculations can be used to determine the most appropriate sensor for a given task, and the number of images of the same scene that are necessary to achieve a required resolution.

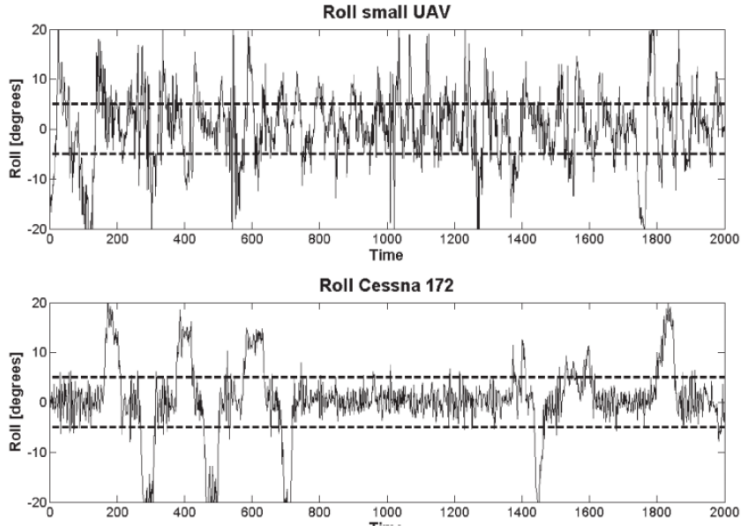


Figure 2.5: Comparison of roll angles on sUAS and manned aircraft [53].

## 2.3 Mechanical Vibration Mitigation

Dampers are very popular devices for vibration mitigation. A wide selection of vibration dampers are available off-the-shelf. The knowledge of how to select the best damper is however not widespread, and it is common to just try a few and see which one works best. By using a more systematic design method together with collected vibration data (or estimated from the vibration sources), it is possible to remove targeted vibration frequencies more efficiently. Since trial-and-error might be resource-intensive in both man-hours and components, a systematic approach is likely to reduce both cost and time of developing a stabilizing system.

Also, to reduce the vibration effects, actions can be taken in the sUAS platform design phase, when selecting the materials and when designing the structure.

### 2.3.1 Dampers on the optical imaging system

Dampers can be used for vibration isolation, to lower the natural frequency of the system to below the excitation frequency, and for vibration damping, where the aim is to absorb the mechanical energy and convert it to other energy forms, such as heat. Three types of dampers (Silicone Foam; Kyosho Zeal; and Sorbothane 30 Durometer Sheets) (Figure 2.6) were tested by [155] to mitigate the vibration on a rotary-wing sUAS. In total, six aspects were taken into consideration when choosing the dampers: 1. Electrical insulator to avoid short-circuit; 2. Soft and flexible; 3. Natural frequency outside sUAS structural resonance zone; 4. Low compression set

and low creep; 5. Good resistance to outdoor conditions; and 6. Easy installation and adjustment.

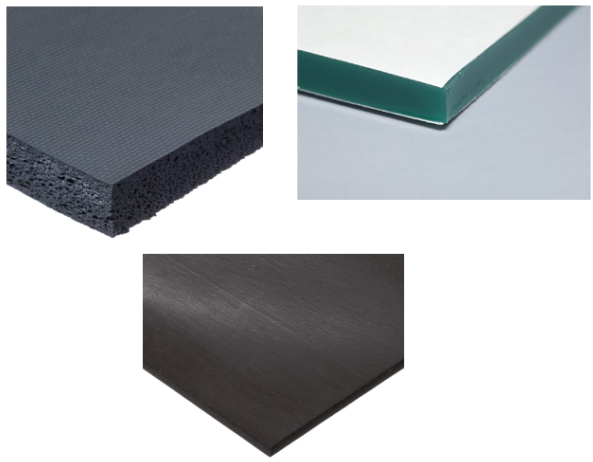

Figure 2.6: Silicone Foam (top left); Kyosho Zeal (top right) and Durometer Sheet (bottom) (Adapted from Amazon.com)

To study the effect of the dampers on the sUAS vibration mitigation, a structural vibration analysis was done before the installation of the dampers. In the first step, the sUAS structure was modeled in SolidWorks simulation, considering also the materials properties. The frequency analysis was carried out and the high vibration frequencies observed were 39.90 Hz on the x and y axes, and 80.48 Hz on the z-axis. Similar behavior also existed for 160.17 Hz and 321.82 Hz respectively, which the author suspects are the 3rd and 4th mode natural frequencies, related to the payload. Small vibrations were obtained between 100 to 200 Hz and a stronger one at 273.16 Hz. The last ones are probably related to the structure, such as arm extensions and components. After the simulation, flight tests were performed and the vibrations were measured by an additional IMU with a sampling rate above 800 Hz. Results (Figure 2.7) show that the vibration data obtained on the flight tests was very similar to the simulated one. Peaks (A) related to the payload were verified on the vibration frequencies of 40, 80, 160 and 320 Hz, a peak (C) related to the structure was verified on around 270 Hz, and another peak (B) was verified on around 50 Hz, which is related to the rotation of the rotors. The last one was not verified in the simulations because the vibration related to the rotation of the rotors was not included on the simulated model.

The tests of the effectivity of the dampers were performed on a lab vibration table for a frequency range from 10 to 300 Hz. Different sizes of each of the three dampers were used in order to change the transmissibility curve. Among the selections,

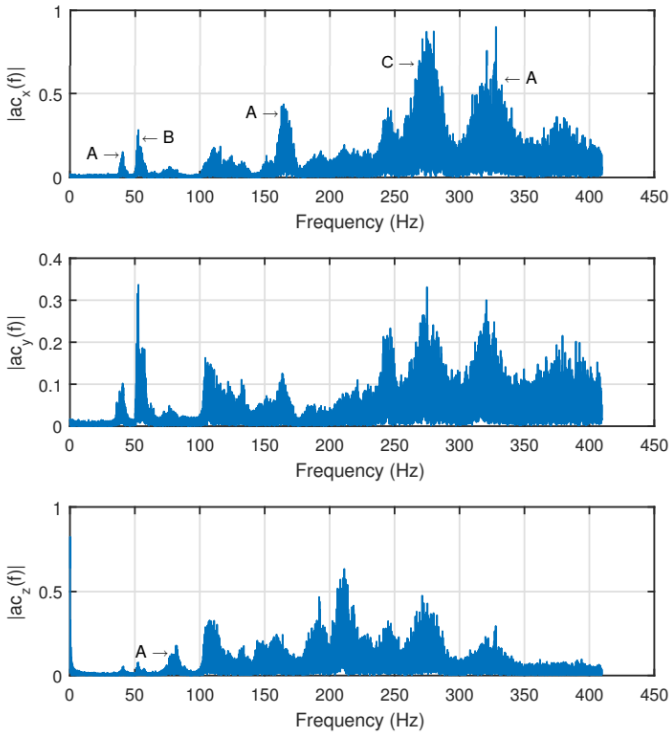


Figure 2.7: sUAS vibration measured by an IMU before the damper installation [155].

Kyosho Zeal Sheet had the best performance. Therefore, it was installed on the sUAS to mitigate the vibrations on the additional IMU. The size of the damper was chosen so that the natural frequency of the damper (around 50 Hz) was further apart as possible from the highest disturbing frequencies (around 270 Hz) to be mitigated. Results of the flight experiment (Figure 2.8) show a significant reduction on the vibrations. However, as expected from the transmissibility curve of the damper, vibration caused by the rotors at around 50 Hz was slightly amplified.

### 2.3.2 Other mechanical solutions

In addition to the use of dampers on the optical imaging system, other actions can be taken to reduce vibrations on sUAS. In rotary-wing platforms, an accurate balancing of the propeller blades may reduce the propeller-induced vibrations significantly [96]. Other suggestions are given by [140], focusing on the reduction of noise produced by sUAS, but, as in many cases the noise is related to the vibration of parts of the platform, the same actions can be applied to vibration mitigation. The author categorizes the methods into five groups: conventional noise

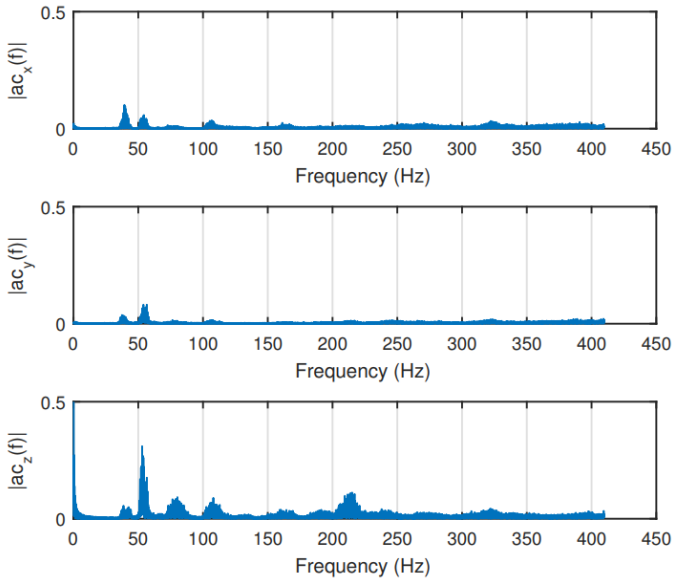


Figure 2.8: sUAS vibration measured by an IMU after the damper installation [155].

control methods by modifying the structure; futuristic methods; reduction of engine noise; operation time adjustment-based approach; and noise reduction targets of the federal agencies. The first method consists on passive and active techniques. An example of a passive technique is to use vibration-absorbing materials on the structure and therefore reduce the vibration on a specific frequency band. Actives techniques are, e.g., closed loop adaptive feed forward control techniques with electromechanical systems, such as piezoelectric actuators, to reduce the vibration of surfaces [33]. The second category is regarding the design and selection of materials that could reduce the vibration, such as "owl wings". Engine noise mitigation is the third category and the author points out some methods from the literature, such as structural modifications; active noise control systems; slightly changes the phase between the propeller sets; and modifying blades and controlling the rotation speed. All these techniques are to be considered during the design and manufacturing phases and should be taken into account if the purpose of the sUAS being developed is sensitive to vibrations. The last two categories discussed by the author are not related to vibration mitigation.

### Flow-induced oscillations

During the design of a optical imaging system it may prove beneficial to consider the effects of flow induced oscillations on the structure. By minimizing the occurrence of the oscillations, the overall system performance may increase, while the need for

mitigation through dampening could be reduced. Here the most relevant aspects with regards to the stabilized imaging system design are discussed. Vortex-induced vibrations (VIV) is a phenomenon where the generated vortices cause vibrations of the object. This is caused by the asynchronous periodical release of vortices along the object. The magnitude and impact of VIV is highly dependent on the flow conditions and the shape of the body.

Because the optical imaging sensor is mounted on the airframe, it may experience vibrations that are not necessarily caused by the optical imaging sensor itself, but by the airframe or components that are installed on the airframe. Besides more obvious sources of vibrations, the aircraft may suffer from secondary aerodynamic effects. One prominent example of this is *Dutch roll*. This is an oscillatory movement caused by a change in the aircraft's yaw, which is coupled into a roll movement. Because a yaw movement forces one wing forward in relation to the other, a differential in that wing's lift and drag occurs. This causes the aircraft to wiggle. The dynamics behind Dutch roll are considered a difficult dynamic mode to analyze [150]. However, if the performance of the optical imaging sensor suffers from the effects of Dutch roll, the typical remedies include an increase of the aircraft's vertical tail, or the installation of a yaw dampener [110]. The occurrence of Dutch roll can be recognized by an oscillatory motion, where the roll motion lags behind the yaw motion by approximately $\pi/2$.

## 2.4 Optical Image Stabilization

In imaging missions, an alternative or supplement to mechanical vibration mitigation is Optical Image Stabilization (OIS) [151]. This technique consists of using motion sensors readings to detect vibration and to move the lens or sensor in order to correct the jitter. Nowadays, many camera systems, especially the ones installed on the most modern phones come with this capability. Basically, actuators move the camera system parts according to the detected vibrations, cancelling the effect. OIS is considered superior to digital image stabilization as it acts before the image acquisition and therefore there is no image distortion or degradation. Despite its advantage, not all imaging systems have this feature and the installation of OIS on the existing imaging systems is very challenging as the intervention happens in the hardware of the imaging system. OIS can also alter camera parameters, reducing the accuracy of remote sensing data.

[84] evaluated the performance of an OIS system using fuzzy sliding-mode controller under the effect of sinusoidal signals of 6 Hz, 8 Hz, 10 Hz and 12 Hz. The camera acquired images of a standard ISO-12233 chart with OIS ON and OFF. Figure 2.9 shows the comparison between the chart picture taken from a camera with and

without OIS turned on for vibrations of ±0.15 degrees on the vertical axis.

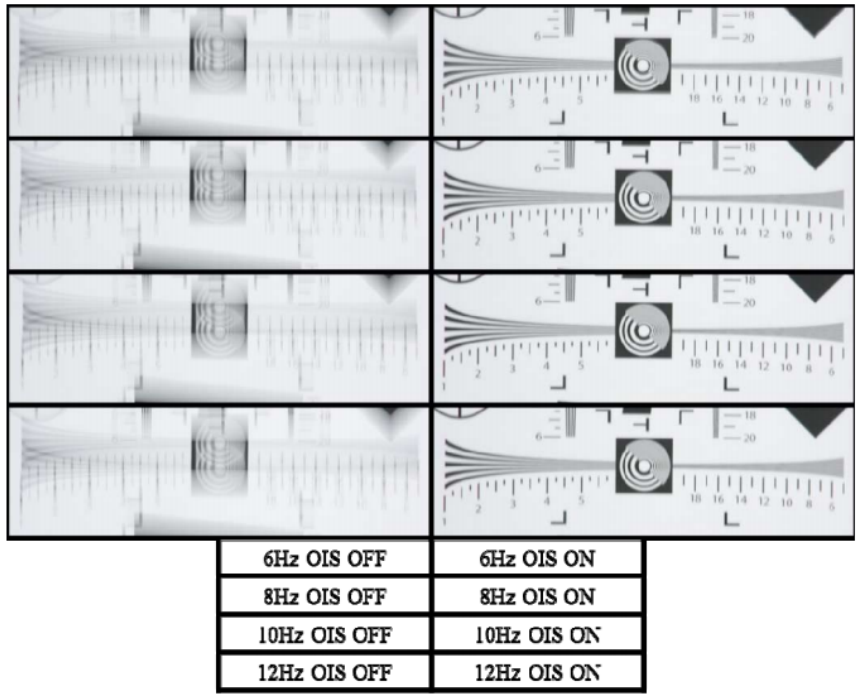


Figure 2.9: Comparison of OIS OFF and ON on a standard ISO-12233 chart [84].

## 2.5 Software Image Stabilization

Image stabilization algorithms are also a way to reduce the effect of vibration on images. This category of image processing is usually referred to as Digital Image Stabilization (DIS). It is important to make a distinction between DIS and digital video stabilization. Digital video stabilization consists of removing the effects of unwanted camera motion from video data; and Digital Image Stabilization (DIS) consists of correcting the effects of unwanted motions that are taking place during the integration time of a single image or video frame [137], by estimating the motion between frames in sequential imaging and then removing unwanted camera motions.

### 2.5.1 Digital Image Stabilization

In DIS, motion estimation techniques can be classified into two categories: feature-based [111] or direct pixels-based [70] (also called "image-based"). The main difference is that feature-based approaches extract characteristics of the frames such as

corners, edges, etc, while direct pixels-based approaches use every single pixel on the calculations. Therefore, techniques using the feature-based approach are usually faster and more effective but implies non optimal use of the available information. Also, in images where the degradation caused by vibrations is too accentuated, the number of detectable features is small and the features may not be sufficiently reliable, therefore, a direct pixels-based approach would be more suitable because it uses the intensity of every single pixel of the image. The motion estimation is usually done by estimating a parameter vector, which is a two-dimensional mapping function that overlaps input images over a reference image [137]. The reference image has to be chosen among a sequence of images and a good candidate to be a reference image may be the one the least affected by blur. To identify such image, a sharpness measure can be used.

Figure 2.10 shows an example of a comparison between an image captured with exposure time of 1.8 seconds where DIS was not applied (Figure 2.10a) and a resulted DIS image (Figure 2.10b) using 4 frames captured with 0.3 seconds of exposure time each. It is possible to notice that the image for which the DIS algorithm was applied is less blurry.

(a) (b)

Figure 2.10: (a) non stabilized image taken with exposure time of 1.8 sec; and (b) stabilized image by fusing four frames with exposure time of 0.3 sec each [137].

### 2.5.2 Digital Video Stabilization

In digital video stabilization, where the goal is to make the video flow less trembled due to the movement of the camera, the motion between frames is also estimated by calculating the rotation and translation between frames. Then, the opposite motion can be applied to counteract image shake and realign the frames in order to make the transition between frames smoother [16]. This is a very popular topic on sUAS imaging because video taken from sUAS frequently suffers from unwanted motion of the sensors.

In [136], Scale Invariant Feature transform (SIFT) was used for key point detection and matching between successive frames taken by a sUAS. Then, an affine transformation model was used to estimate the global motion parameters between two successive frames. After that, the undesired motions were compensated and spatio-temporal filtering was used to remove the noises in the video. Finally, all frames were transformed to obtain stabilized video frames.

A fast video stabilization for sUAS was proposed by [124]. A polynomial fitting and predicting method was proposed to estimate the global motion parameters and to select undesired frames. After that, the undesired frames are compensated and all frames are transformed to obtain stabilized images. Figure 2.11 shows the compensation (Figure 2.11d) of an undesired frame (Figure 2.11c).

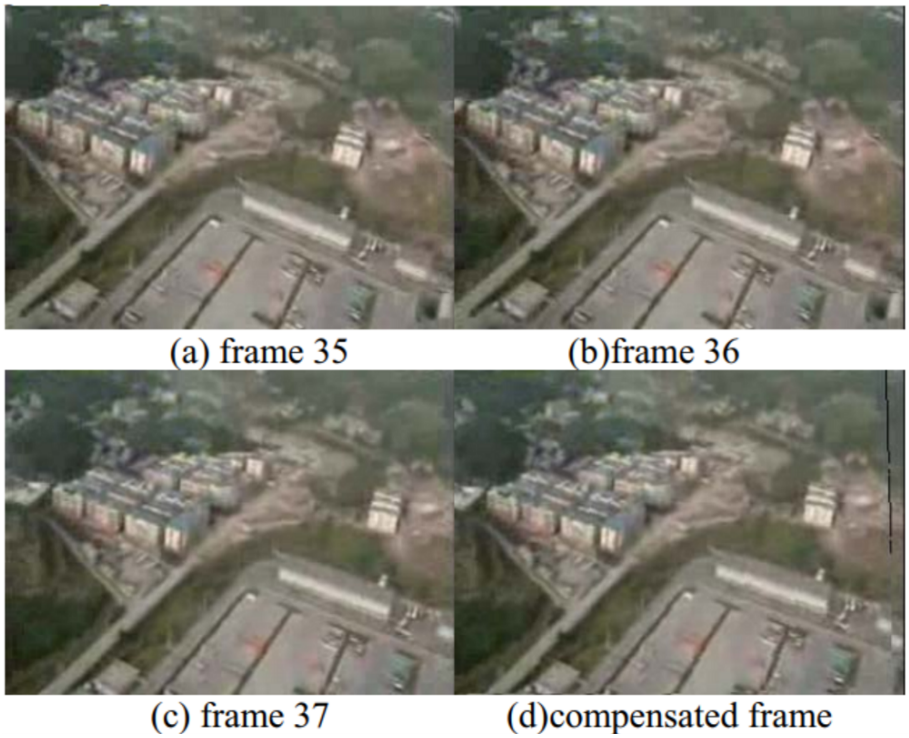


Figure 2.11: Compensation of an undesired frame for video stabilization [124].

As most of sUAS are equipped with inertial measurement units (IMU), an alternative to the feature- or pixel-based motion estimation for digital video stabilization is to use the IMU readings to calculate the camera motion between frames and use this information to stabilize the camera feed [100]. IMU readings can also be integrated with conventional motion estimation methods to increase the speed and accuracy [121]. In this case, the results can be improved significantly as the ac-

curacy of the motion estimation increases. A timing and navigation solution was developed by [4], which made it possible to synchronize the camera images with the sUAS position with high accuracy by using dedicate hardware time synchronization of GNSS, IMU and camera sensors readings.

## 2.6 Gimbal Stabilization

sUAS gimbal systems are electromechanical devices that can be used to stabilize a platform on a given attitude. Therefore, they are suitable to mitigate unwanted camera rotations caused by the UAS motion and also for pointing the platform on a desired direction, controlling the sensor's line of sight (LoS). Such systems have already been used in many areas before, such as spacecrafts, manned aviation and cinematography. However, due to the recent availability of small UAS and its market growth, there is a new demand for small and precise gimbal systems specifically optimized for particular requirements regarding size and precision. Also, this topic is benefiting from the advances in the miniaturization of key technologies such as high-performance gyros and drivetrain components, fast embedded microcontrollers and small cameras. Therefore, small gimbal system design is a topic just recently being researched, which the main challenges are regarding the limited size and weight of the device. In order to evaluate the different gimbal designs, [95] has undertaken a number of trade studies, investigating various gimbal configurations, sensors, encoders, drivetrain configurations, control system techniques, packaging etc.

### 2.6.1 Gimbal Systems Classification

[95] classifies the gimbal systems according to stabilization performance related to the LoS Jitter (µrad RMS). Low performance gimbal systems are the ones with more than 250 µrad RMS of LoS Jitter. Medium quality with from 25 to 250 µrad RMS and high quality with less than 25 µrad RMS. Reference [18] classifies the gimbal systems by crossing size and LoS stabilization performance in degrees. Small gimbal systems weight up to 4.5 kg and can achieve a LoS stabilization performance on the order of $\pm 0.5$ to $\pm 0.1$ degrees. Medium and larger gimbal systems weight from 4.5 to 9.0 kg and greater than 22.5 kg can and can achieve less than $\pm 0.1$ degrees of LoS stabilization performance.

### 2.6.2 Gimbal Systems Design Considerations

Regarding sUAS gimbal systems design, while [95] did a wide comparison between different configurations, in order to provide a broad overview of the design concepts, [18] focused his work on studying the topic to define the best approach for

the development of his specific gimbal system, designed to meet size, weight and performance requirements previously defined.

According to [95], the first aspect of gimbal system design is to decide the number of gimbal axes needed for a desired LoS control and field-of-regard (FoR), which is the area over which the gimbal can point. A minimum of two axes are required for controlling two degrees of freedom and point the LoS in a two-dimensions (vertical/tilt/pitch and horizontal/pan/yaw) desired direction. To control a third degree of freedom, such as the image orientation, a third axis is needed. In small sUAS, two-axis gimbal systems are most commonly used. In fixed-wing sUAS, it is common to use gimbal systems with an outer azimuth gimbal axis to control the pan so that it is able to rotate 360 degrees and therefore have a wider FoR.

Another important design aspect is to correctly align the center of gravity with the gimbal motors. By doing this, the required torque and power to make precise angular rotations can be greatly reduced [24].

Thermal considerations must also be addressed because sUAS gimbal systems are commonly too small to package cooling fans or heat exchangers. Therefore, correct electronics layout and proper materials selection are the best measures to mitigate thermal problems.

### 2.6.3 Stabilization

As the gimbal system will be mounted on a sUAS, which is subject to vibrations caused by rotors, engine or turbulent aerodynamic flows, the gimbal system vibration isolation must be addressed. Combustion engine powered sUAS produce large torque pulses, due to the non-continuous nature of their operation, often in the range of 50-80 Hz. This can cause significant image blurring and/or excitation of jitter in the gimbal's control system, if no specific vibration isolation is provided. Electric powered sUAS produce higher frequencies, which are easier to mitigate and have less of effect on the image quality.

Gimbal system stabilization can be active or passive. Passive stabilization is related to the fact that the platform, sensor, and target LoS move within inertial space. Therefore, low friction joints and high inner axis inertia can passively contribute to maintaining the desired LoS/attitude [18]. Active stabilization is done when the drivers will act based on sensor readings to keep the platform's desired attitude/LoS. Therefore, to mitigate these vibration effects, [18] designed an active inertial dampening to take care of frequencies of less than 5 Hz and the gimbal system mechanical design provides a good passive inertial dampening for frequencies on the order of 5-20 Hz. For higher frequencies, the gimbal mounting system

is responsible for dampening them out. According to [95], an option to mitigate severe effects of sUAS vibration could be to decouple the gimbal system from the sUAS by vibration isolation, isolating its parts from the sUAS structure, however, this solution may degrade the accuracy of any type of pointing relative to the vehicle, and can induce angular motion inputs. This degradation occurs due to the necessity of accurate measurements the gimbal position relative to the sUAS in order to send the correct controls to the drive systems.

Regarding the vibration caused by the gimbal system structure itself, fortunately, the frequency of the gimbal system structural resonance is typically higher for smaller gimbals than larger gimbals [95]. Therefore, structural vibration effects are often much less a design issue in small sUAS gimbal system design but nonetheless important to consider. The most common approach to deal with this is to include structural notch filters, which helps to improve the loop gain margin at the resonant frequency. Reference [58] does a deeper analysis of the gimbal structural interactions and suggests to stiffen the structure, as a first attempt to attenuate LoS motion due to bending and to modify the relevant structural transfer functions. The author also suggests to stiffen the torsional response of the mounting structure, to add mass to the stationary gimbal structure, and to employ the notch filters in the pointing servo system to achieve a better interaction of the control system with the structure.

### Drive System

The gimbal drive system can be direct, where the motor controls the axis directly, or indirect, via cables, gears or belts. Gimbals with brushless DC direct drive have the highest performance, being able to achieve very low friction and no reflected inertia [95]. However, it is usually heavier, bigger and more expensive than the other approaches to achieve the same torque, and needs more complex electronics. If an indirect drive with gears or belts is chosen, the solution is cheaper and smaller, but has increased backlash, hysteresis, cogging and compliance as result. The cable drive approach has a performance between the direct drive and gears/belts approaches but it has higher friction and lower stiffness, and the difficulty to achieve 360 degrees continuous motion for the yaw axis.

[18] designed the gimbal system using brushless DC servomotors with belts, pulleys and gears for its axes. The first design attempt achieved too high backslash in the pan axis. Therefore, the final decision was to use the motor without a gearbox driving a small rubber wheel directly on an interior bearing surface.

As part of a fixed-wing sUAS imaging system design, [127] designed and produced a new gimbal system aiming to achieve a better stabilization, wire handling, re-

pairing capability and robustness than the off-the-shelf gimbal systems. As self stabilizing direct drive gimbals using speedy brushless motors became available in the market for multi-rotors, the authors decided the use this kind of motor on their new gimbal system design specially because they allow gimbals to move fast enough to stabilize the camera from low frequency vibrations. The commercial brushless gimbal systems are often supplied with a dedicated controller which uses input from an IMU to be mounted on the camera. However, according to the authors, angular drift in heading is a potential worry when using a brushless motor for the yaw rotation, specially because of the bias instability of the IMU. A possible solution to remediate this problem could be to implement an estimator between the IMU and the controller board, integrated to the sUAS' heading estimator. Another advantage with using brushless motors and IMU is no need to index the gears when disassembling the gimbal. Therefore, as long as the IMU is reinstalled in the same location and orientation, the gimbal will calibrate itself on start-up.

The minimum torque of a motor used to stabilize a given sensor is the sensor's moment of inertia times the desired angular acceleration.

$$T = J\alpha, \tag{2.1}$$

where $T$ is the torque, $J$ is the moment of inertia and $\alpha$ is the angular acceleration.

Therefore, the first steps when choosing the right motor is to calculate the moment of inertia and to choose the desired angular acceleration.

[18] derives the equations detailed by [76] which includes all torque contributions and consequences. In the same axis, the contributions are from the torque of friction and cable restraint and the mass imbalance torque. In case of the inner axis (e.g elevation), where the gimbal is mounted on the sensor body, the mass imbalance torque is caused by the asymmetry of the sensor. In the outer axis (e.g azimuth), where the gimbal is mounted on the inner gimbal mounting that is connected to the sensor body, the mass imbalance torque is caused by the asymmetry of the sensor plus the inner gimbal mounting asymmetry.

In [24], where a gimbal system to house two imaging sensors was designed, the authors also opted to use brushless DC motors to directly drive the gimbal axes because of their superior small angular rotations compared to servos. Low weight motors (109 g per motor) capable to carry the payload (around 400 g) were chosen.

**Motion Sensors**

Gyros are the main rotational sensors used in gimbal systems. They measure gimbal angular velocities, and are used as system's feedback. Gimbal gyros are usually based on Micro-Electro-Mechanical Systems (MEMS) technology and should have the lowest noise and bias instability that the design constraints allow.

Resolvers and encoders are the most common gimbal angle transducers [95]. They are used to detect the orientation and to report the absolute position of the axes. Resolvers are more robust but incremental encoders are becoming very popular for gimbal angle measurement because they are smaller, lighter and cheaper and can achieve comparable resolution and accuracy. Encoders can be optical, capacitive or magnetic. Reference [18] used a 12 bits of resolution magnetic encoder for each axis of the gimbal system design.

## 2.7 Impact of the Stabilized Imaging System on sUAS Aerodynamics

Implementing a stabilized imaging system that is optimized for airborne vehicles requires consideration of its effects on the in-flight performance. Studying the effects early in the design process may enable a reduction in the negative impact on the in-flight performance. Presented in this section are the system design trade-offs in relation to the overall aircraft performance. Besides the energy consumption of the electrical components, the in-flight performance of the aircraft is primarily affected by the system's mass, shape and position. In addition to the flow-induced oscillations, as described in section 2.3.2, these parameters are elaborated to such an extend that it gives the essentials in design considerations in the context to aircraft performance.

### 2.7.1 Impact of weight

Aerial vehicles stay afloat in the air by generating a force that is equal and opposite to earth's gravitation force. In the case of conventional fixed-wing and rotary-wing aircraft creating such a force requires the consumption of energy. To generate such a thrust force the rotary-wing aircraft utilize one or more powered propellers which are positioned so that it directly counteracts the gravitation force. Fixed-wing aircraft utilize one or more propellers to generate a forward motion, which results in the main wings to generate a lift force that opposes the gravitational force. As the total mass of the aircraft increases, so does the required lift force. This results in an increase in energy consumption.

In an attempt to demonstrate the importance of weight reduction, the effects of

weight on the energy consumption of the aircraft are quantified. For both fixed-wing and rotary-wing sUAS, the maximum range by approximation is reduced proportional to the increase in weight. For sUAS with constant mission variables, the maximum endurance is reduced by a factor of approximately $W^{3/2}$, where $W$ is the weight, for both fixed-wing [110] [63] and rotary-wing [82]. Therefore, an increase of 5% on weight, for example, means a reduction of approximately 7% of endurance.

### 2.7.2 Impact of shape and size

Moving objects placed inside a viscous medium, such as air, are bound to create external forces. Overcoming the effects of such forces typically increases the in-flight power requirements. A stabilized imaging system that is placed inside moving air will contribute to the total drag force and will thus typically reduce the overall in-flight efficiency. It is therefore warranted to optimize the design of the system to reduce the impact through aerodynamic considerations.

#### Impact of shape and size on fixed-wing aircraft

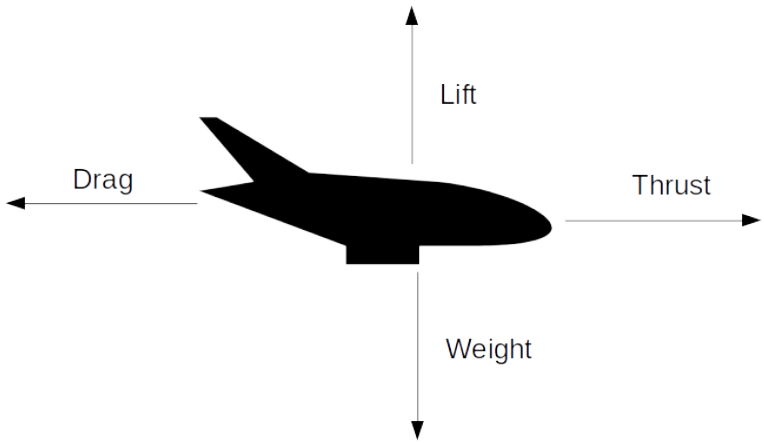


Figure 2.12: Aerodynamic forces acting on an airplane - Thrust, Lift and Drag.

Aerodynamic drag, also known as air resistance, is the force parallel to the airspeed [49] (Figure 2.12). With an increase in drag, the aircraft needs to compensate the energy losses by producing more thrust in order not to lose speed or altitude. Considering that a typical stabilized imaging system is not intended to generate a lift force, it may be assumed that the in-flight performance benefits from minimizing the total aerodynamic drag generated by system. To be able to reduce the aerodynamic drag it is important to understand how it is built up and how it is affected. The total drag force $D$ generated by an object can be determined through:

$$D = C_D \frac{1}{2} \rho v_a^2 A, \tag{2.2}$$

where

$$C_D = C_{DP} + C_{DI}, \tag{2.3}$$

where $\rho$ is the air density in kilogram per cubic meter, $v_a$ is the speed of the moving air before being affected by the object in meters per second, $A$ is the cross-sectional area of the object in square meters, and finally $C_D$ is a dimensionless drag coefficient that relates the object's shape, inclination and flow conditions to the resulting drag force.

The takeaways from Eq. (2.2) in the context of stabilized imaging systems design are that the drag generally can be reduced by minimizing the size of the object, and that the drag increases exponentially with airspeed. Thus, fast flying aircraft suffer much more from poor design choices than slow flying aircraft. Finally, the drag coefficient $C_D$ ought to be minimized. This can be done by optimizing the shape or placement of the object in such a way that it causes the least interference with the moving air. It should be noted that the complete theoretical basis of aerodynamic flow optimization falls beyond the scope of the study presented here, and is therefore limited to the most relevant aspects. Assuming that the casing of the camera system is not designed to generate lift, the lift-induced drag ($C_{DI}$) is negligible. The remaining parasitic drag ($C_{DP}$) can be categorized into:

1. *Form drag:* The form drag is influenced by the shape of the object (Figure 2.13). Although the droplet shape offers the most favourable aerodynamic characteristics, when pointed straight into the direction of the moving air, it also offers challenges related to possible viewing angles of the camera system. Therefore, the aerodynamic considerations may be considered a performance parameter within the overall geometrical optimization of the camera system.
2. *Skin friction drag:* As air moves over the surface of the body, close to the surface the flow will lose energy due to viscous effects. This type of drag is called skin friction drag. A turbulent boundary layer that is induced by a rougher surface may stay attached longer than a laminar boundary layer, thus reducing the form drag. This generally holds true for smaller object in relatively low air speeds [110]. Therefore, for smaller objects the negative effects of higher skin friction drag, which is caused by a rougher surface, may potentially be offset by a lower overall profile drag. Finally, it should be noted that such potential benefits are highly dependent on the specific design

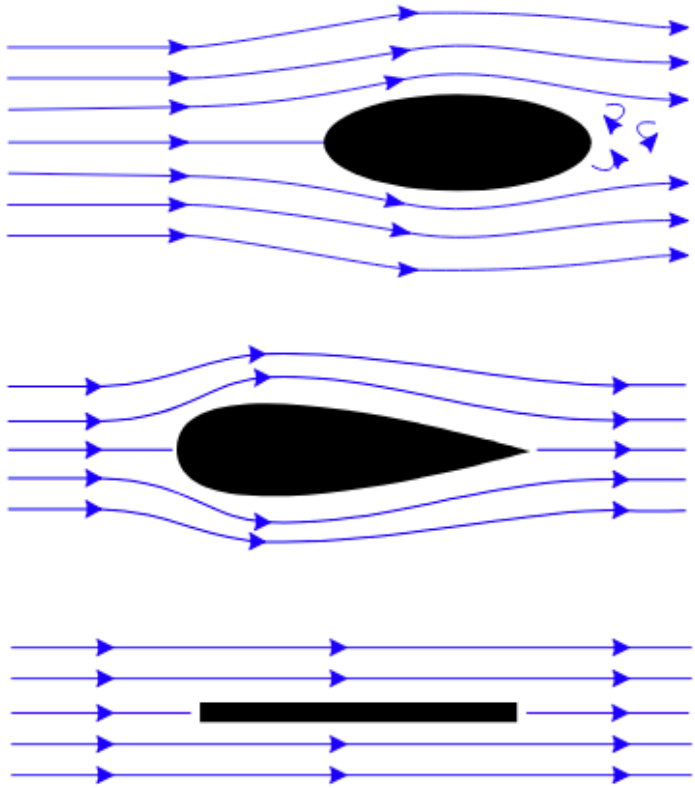

Figure 2.13: Flow visualization over different shapes (Modified from MikeRun / CC-BY-SA-3.0).

and flow conditions, and therefore require an in-depth aerodynamic analysis. Projects where such analyses is not within reach, may benefit from focusing on the form drag and interference drag instead.

3. *Interference drag:* In the context of aerodynamics, the interference drag can be explained as the airflow over one object disturbing the airflow over another object unfavourably. The actual effects of interference drag depend to large extend on the airspeed. Therefore in the context of aircraft performance it cannot be said that a closed and shielded system is necessarily superior to a system with exposed components as it may also be heavier. It is dependent on mission-specific parameters. The following section suggests methods to study the effects of on-board camera designs, including the interference drag characteristics.

To quantify the actual impact of a camera design on the aircraft performance, there

are three methods that ought to be considered. The first is through wind tunnel testing. Inside a wind tunnel the effects of moving air over an object are recreated, and the generated forces and moments are captured. The camera can be placed inside the wind tunnel without the airframe. However, in that case the aerodynamic interactions between the aircraft body and stabilized imaging system are not included. If executed and post-processed correctly, a wind tunnel test can give an accurate indication of the impact of the camera system on the flight performance. Also the airflow around the bodies can be visualized, which provides information for further optimizing the airflow. However, modelling through wind tunnel experiments are complex and require dedicated equipment. This causes wind tunnel experiments to be relatively expensive and time consuming. The iterative design process is commonly also slow when compared to its alternatives.

Nowadays, designers have embraced *Computational Fluid Dynamics* (CFD). This is a computer-based method that can approximate the behaviour of fluids, such as air, over an object. It may prove useful for design optimization as it allows for quicker iterative development. In popular terms it is sometimes called a *virtual* wind tunnel. It is to be noted that setting up such a simulation environment requires in-depth knowledge on the theory of fluid dynamics. Also, when results with a high accuracy are required, then a verification of the model is necessary. This is commonly done inside a wind tunnel. This is especially true for low velocities and/or smaller objects, such as camera systems, as the modelling of drag then becomes increasingly problematic.

The final method discussed here is through actual flight tests where the performance is compared with and without the camera system installed. Measuring the consumed energy in cruise flight may serve as an indicative measurement for the impact on the in-flight performance of the aircraft. Such in-flight comparisons are only valid when all mission parameters, including airspeed, altitude, atmospheric conditions and battery charge, are the same in each benchmark flights. Since this may be hard to accomplish and verify, it is important to note that the obtained results are non-conclusive, and can only serve as an approximation. The advantage of this method is that it is accessible and does not require in-depth knowledge of fluid dynamics. This may serve as a suitable method when an approximation is sufficient.

**Impact of shape and size on rotary-wing aircraft**

For rotary aircraft operating in *stationary* flight there is no forward motion of the vehicle. As the air in front of the rotor is accelerated by the rotors itself, the before mentioned aerodynamic effects require additional design considerations. By locating the camera system outside of the propeller slipstream ($v_s$), the impact of the

aerodynamic effects are limited (Figure 2.14). When the aerodynamic effects as a result become small, such a system may not require further aerodynamic optimization. Note, however, that when operating in atmospheric winds the aerodynamic effects remain. For rotary-wing aircraft, a center position ought to be considered. For conventional rotary aircraft, which typically utilize one main rotor, the camera mounting points are commonly found under the main body. As the aircraft's main body shields the camera system to a large extend from the propeller slipstream, the need for aerodynamic optimization decreases. However, due to other design considerations, such as clearance to the ground, these mounting locations may not be feasible. For conventional rotary-wing aircraft the camera systems are often found to be mechanically suspended in front or along the side of the aircraft body. In such cases the aircraft performance benefits from a minimized wetted area, which is the area exposed to the airflow, in order to reduce the interference drag and propeller blockage.

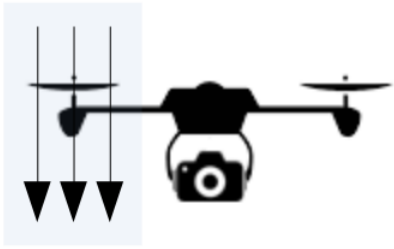

Figure 2.14: Visualization of the propeller induced slipstream vs in relation to the camera location.

### 2.7.3 Impact of Position

For fixed-wing and rotary-wing aircraft to be controllable in flight, it relies on the ability of the aircraft to compensate the generated forces which are experienced. Each individual component installed on an aircraft has a mass. When these components are exposed to the moving air it will create a drag force, while lifting bodies may also generate a lift force. First, for the aircraft to maintain altitude in the air, the aircraft needs to be able to generate enough lift force (or thrust) to compensate the aircraft's total weight. In other words, for level flight the sum of the vertical forces equals zero. In addition, all these individual forces, such as lift and drag will generate a moment around the aircraft's center of gravity (C.G.).

## 2.8 Results and Discussion

Vibration is one of the main concerns when designing sUAS optical imaging systems. In fixed-wing platforms equipped with a combustion engine, the engine is the main source of vibration, followed by the turbulent aerodynamic flow. Rotary-wing platforms have the rotors as the main source of vibration. To mitigate the effects of vibration, mechanical solutions such as dampers can be used. Also, it is possible to use optical stabilization by installing motion sensors to measure the jitter and use actuators to move the lens in order to correct it. Software solutions are also available, such as digital image stabilization or digital video stabilization algorithms. Gimbal stabilization platforms can be used for stabilization and also for pointing. The most important components in these platforms are the drive systems and the motion sensors. The installation of a stabilization platform may affect the sUAS aerodynamics and the impact of weight, shape, size and position must be taken into consideration in the design phase.

## 2.9 Conclusions

In this work we presented a wide survey of the literature about optical imaging stabilization systems and techniques applied to Unmanned Aerial Systems (UAS). This includes discussions about the sources of vibration, how to mitigate its effect using mechanical and software solutions, as well as the effects of stabilization platforms on the UAS aerodynamics.

# Chapter 3

# Skyline Based Camera Attitude Estimation Using a Digital Surface Model

In several motion control applications, e.g. in precise pointing devices and navigation systems, an accurate estimate of the attitude of a device in the world coordinate system is required. Readily available sensors used to estimate the attitude suffer from drift, magnetic disturbances, or a lack of information about the direction of the geodetic north. In this chapter, an algorithm which estimates the attitude of a camera using camera images and a Digital Surface Model (DSM) referenced by GPS is proposed. The algorithm uses shape features of a skyline extracted from a camera image and synthetic skylines rendered from a DSM in order to estimate the roll and pitch angles, while the yaw angle is estimated using grid search. The algorithm was evaluated using roughly 600 camera images captured with varying pitch and yaw angles. The standard deviations from the ground truth, provided by a high precision pointing device, are 0.037°, 0.015°, and 0.018° for roll, pitch, and yaw respectively. The results indicate a higher precision than current camera attitude estimation algorithms using a DSM, while also providing a robust yaw estimate.

## 3.1 Introduction

In a number of motion control applications, it is necessary to have an accurate estimate of the attitude and position of a device in the world coordinate system in order to navigate or rotate relative to a point with known world coordinates. A common solution to the problem of estimating the attitude is to use an Inertial Measurement Unit (IMU). The IMU commonly contains accelerometers, gyroscopes, and

magnetometers. The accelerometers are used to estimate the roll and pitch angles and the magnetometers are used to estimate the yaw angle. The roll and pitch angles can often be estimated with a sufficient accuracy, but additional sensors may be needed to capture the effects of highly dynamic motions. Moreover, the lack of accuracy in the magnetometer measurements makes the yaw angle difficult to estimate with the required accuracy [44]. In systems with significant magnetic field noise, e.g. when electric motors or large iron parts are near the magnetometers, the error can be very large – 10 or more degrees are not uncommon.

Cameras, being highly accurate and readily available sensors, are often used to measure the spatial dimensions of objects when its position and attitude in the world coordinate system are known. This can also be reversed – if the world position of some points present in the camera image are known, the camera position and attitude can be calculated. Using a camera for the purpose of estimating the attitude has its drawbacks. Most prominent is the requirement that the objects with known position must be visible in the camera image. This might not be the case during certain weather conditions such as fog, or when they are occluded by other objects. The methods are nonetheless suitable for e.g. calibration of other sensors due to its high accuracy, or continuous estimation when the visibility can be assumed to be good.

Using a camera for estimating the attitude can be performed with several methods, e.g. in order to estimate the attitude of spacecraft from the known position of stars [44], or using information about the local terrain represented by a Digital Surface Model (DSM) in order to estimate the attitude. A DSM is an elevation map which specifically includes all objects on the surface, such as vegetation and buildings. A skyline – the line that separates the sky from the terrain and objects – is often available in outdoor scenarios, and can be segmented relatively easy due to the difference in color, illumination, and texture between the ground and the sky. The skyline can also be rendered from a DSM, providing info about its position in the world coordinate system.

Gupta and Brennan [51] uses grid search to match a synthetic skyline generated from a DSM with a camera image skyline in order to refine the attitude estimated using IMU data. Using grid search for three parameters, even with a fast algorithm such as the proposed Random SAmple Grid Search (RSAGS), restricts the usage of the algorithm to refining an initial attitude estimate due to the number of calculations required to search the parameter space. The accuracy is also not on pair with recent similar algorithms with $\sigma_\varphi$ (roll standard deviation) of 0.25°, $\sigma_\theta$ (pitch standard deviation) of 0.13°, and $\sigma_\psi$ (yaw standard deviation) of 0.40°.

Dumble & Gibbens [34] presents an algorithm to estimate both the attitude and

position of a camera in the world coordinate system. The method matches a camera image skyline with precalculated synthetic skylines generated from a DSM. The algorithm performs well with a $\sigma_\varphi$ of 0.025°, $\sigma_\theta$ of 0.066°, and $\sigma_\psi$ of 0.024°. However, the paper only contains synthetic images to evaluate the performance, so it is difficult to say how it performs with real world data. Noise from terrain features, camera imperfections, and rendering of the skyline profiles might introduce additional errors. Additionally, the DSM needs to be preprocessed for the algorithm, which takes up additional storage space.

Grelsson et al. [47] estimates the roll and pitch angle of an aircraft using a fisheye camera and a DSM. The algorithm extracts the skyline from a camera image and matches it with a synthetic skyline generated from a DSM by minimizing the error between the synethetic skyline and edge pixels of a region in the camera image. The algorithm shows good accuracy through experimental data, with a $\sigma_\varphi$ of 0.035° and $\sigma_\theta$ of 0.044°. The method does however not estimate yaw, and assumes that the yaw angle is known. The authors showed that a small error in yaw does not affect the roll and pitch estimates considerably, and mentions that it is straightforward to modify the algorithm to estimate the yaw angle as well. However, there is no experimental data to verify how accurately the algorithm can estimate the yaw angle, or mentions of how it affects the execution time.

The mentioned methods are able to estimate the roll and pitch angles of the camera, and some can also refine or do a full estimate of the yaw angle. However, none of the methods have proven to estimate the yaw angle using experimental data when the initial yaw angle estimate is highly uncertain. In this chapter we present an algorithm to estimate the roll, pitch, and yaw angles of a camera using a DSM. The algorithm uses shape features of a skyline extracted from a camera image and a synthetic skyline rendered from a DSM, together with geometrical properties of the camera system in order to estimate the roll and pitch angles. The fast execution time of the roll and pitch estimation allows for the use of grid search for robust estimation of the yaw angle. The algorithm is evaluated in an experiment where about 600 images were taken using a camera with a 16 megapixel sensor. The camera was mounted on a PTU (Pan-Tilt Unit) which was rotated in pitch and yaw, but had a fixed roll angle and position. A DSM from the Norwegian Mapping Authority (Kartverket) with a grid resolution of 10 meters was used to render the synthetic skyline.

This chapter is structured as follows: Section 3.2 describes the attitude estimation algorithm. This includes the generation of the synthetic skyline from the DSM data, the extraction of the skyline from the camera image, and finally the attitude estimation. Then the experimental setup is described, including the hardware

and the particular DSM used in the experiment. Finally the results are presented followed by conclusions, discussion, and future work.

## 3.2 Attitude Estimation Algorithm

This section contains details of the different parts of the attitude estimation algorithm. The different parts are visualized in the diagram shown in Figure 3.1.

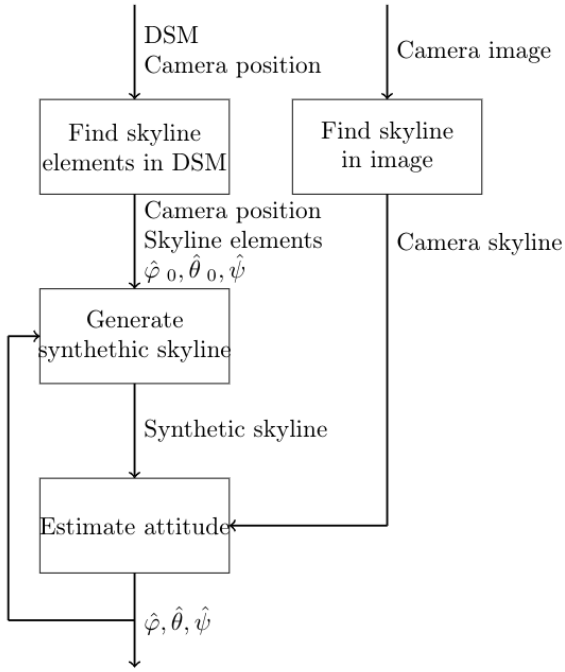


Figure 3.1: Overview of the attitude estimation algorithm. $\hat{\varphi}$, $\hat{\theta}$, and $\hat{\psi}$ is the roll, pitch, and yaw estimate respectively. $\hat{\varphi}_0$ and $\hat{\theta}_0$ are the initial guesses for roll and pitch respectively, which may come from an IMU or the user.

### 3.2.1 Finding skyline elements in the DSM

The first step of the algorithm is to find which elements in the DSM represent the skyline for the current camera position.

The current position of the camera is generally given as geographic coordinates (latitude, longitude, and altitude). In order to find the position of the camera in the DSM, the geographic coordinates can be converted to the coordinate system of the DSM (see e.g. Karney [73]). See figure 3.2 for an example of a DSM field.

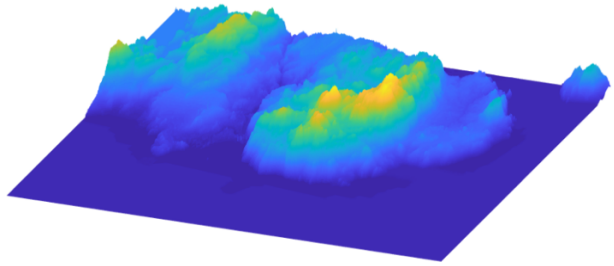

Figure 3.2: A sample DSM field. The view is at an angle to show the elevation difference, which is also indicated by the color change.

Several elevation profiles between the camera position in the DSM and the edges of the DSM are then generated. We experimentally determine the used number of elevation profiles, $n_{\text{lines}}$, as the best compromise between computation time and skyline accuracy.

In order to find which DSM elements should be included in the elevation profiles, Bresenham's line drawing algorithm [20] is used. The algorithm is used to find the elements between two points in a grid. In this application, Bresenham's algorithm is modified to stop at the end of the DSM grid rather than a predetermined point, and to step in the direction $\cos(\alpha_i)$, $\sin(\alpha_i)$ (north, east) where $\alpha_i$ is the angle between the north direction and the current considered direction with index $i$. The algorithm is visualized in Figure 3.3.

For Bresenham's algorithm to be applicable for a DSM grid, the Transverse Mercator projection of the DSM is approximated with a orthogonal projection. This means that Transverse Mercator features such as the scale factor are disregarded and the size of each element in the DSM grid is regarded as the same. The error caused by this approximation may be reduced by selecting a map projection with narrow zone widths. A common projection is UTM (Universal Transverse Mercator) which uses a zone width of 6° resulting in an error of 400 parts per million (ppm) at the central meridian of the zone [39], and no error 180 km west and east of the central meridian. More narrow projections include the NTM (Norwegian Transverse Mercator) which uses a zone width of 1° resulting in an error factor of 11 ppm at the central meridian of the zone [131]. An algorithm for precisely calculating the scale factor for each element in an Transverse Mercator grid is provided by [72].

Each elevation profile is created for the grid elements selected by each line in Bresenham's algorithm. The elevation in the elevation profile is referred to as the relative elevation, $e_{\text{relative}}$, and takes into account the curvature of the earth. The

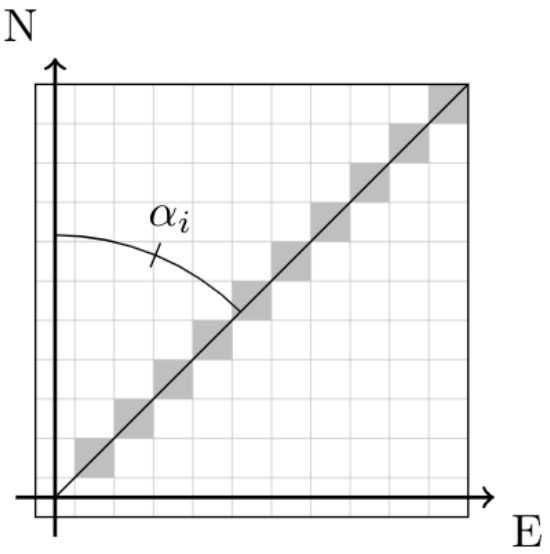


Figure 3.3: Bresenham's line drawing algorithm as used in the generation of a synthetic skyline. The line selects the shaded elements, which in turn are used to generate an elevation profile.

relative altitude for an element in the DSM grid can be calculated according to equation (3.1):

$$e_{\text{relative}} = e_{\text{element}} - \frac{r_{\text{earth}} + e_{\text{camera}}}{\cos\gamma} + r_{\text{earth}} \tag{3.1}$$

where $e_{\text{element}}$ and $e_{\text{camera}}$ are the elevations of the element in the DSM and the elevation of the camera respectively, $r_{\text{earth}}$ is the radius of the earth at the latitude of the camera, and $\gamma$ is the angle between the ground point of the camera and the ground point of the element. See Figure 3.4 for a visualization of the geometry used in equation (3.1).

Finally, the elements representing the highest relative elevations in the elevation profiles are selected as skyline elements for each direction $\alpha_i$.

Finding the skyline elements in the DSM takes several seconds with current hardware. The calculation is therefore performed in the background in order to update the skyline elements as often as possible. Which DSM elements are found to be skyline elements change slowly with the camera position unless the skyline elements are close to the camera. Exactly how this affects the results of the algorithm is

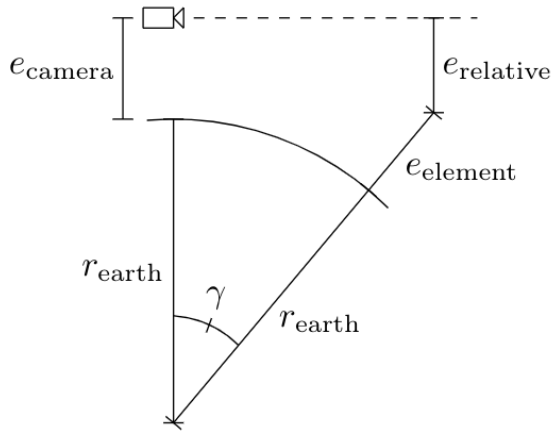


Figure 3.4: The geometry of equation (3.1), where the elevation of an element in the DSM relative to the camera is calculated.

therefore highly dependent on the terrain.

### 3.2.2 Finding the skyline in the camera image

A simple method is used for finding the skyline in the camera image since the focus of this chapter is the attitude estimation algorithm.

The image is first converted to gray-scale and filtered with a Gaussian filter in order to remove noise. For each pixel column in the image going from top to the bottom, the first change in intensity larger than a threshold is considered to be the skyline pixel in that column. The threshold is manually adjusted in order to produce the best segmentation. After each skyline pixel has been found in the image, outliers are removed and replaced using linear interpolation.

The assumptions made for this method to work is that the skyline must go from side to side in the image, the contrast between the sky and the ground must be larger than the noise intensity after filtering, and the intensity difference between the disrupting objects such as haze, clouds, and the sun.

### 3.2.3 Generating the synthetic skyline

The pinhole camera model is used to project the DSM skyline elements from the world coordinates, $p_{\text{world}}$, to the image plane coordinates, $p_{\text{image}}$. The projection from world coordinates to image plane is given by equation 3.2 [108].

$$\lambda \begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} K & 0 \end{array}\right] \left[\begin{array}{c|c} {}^{\text{c}}R_{\text{w}} & {}^{\text{c}}t_{\text{w}} \\ 0 & 1 \end{array}\right] \begin{bmatrix} x_{\text{world}} \\ y_{\text{world}} \\ z_{\text{world}} \\ 1 \end{bmatrix} \tag{3.2}$$

where $p_{\text{world}} = [x_{\text{world}}\ y_{\text{world}}\ z_{\text{world}}]^T$ are the coordinates for the local tangent plane with $x_{\text{world}}$ (east), $z_{\text{world}}$ (north) centered at the camera location, and $y_{\text{world}}$ (down) is zero at the sea level. $p_{\text{image}} = [x_{\text{image}}\ y_{\text{image}}]^T$, $K$ is the intrinsic camera matrix, ${}^{\text{c}}R_{\text{w}}$ the rotation matrix, and ${}^{\text{c}}t_{\text{w}}$ the translation vector from world coordinates to camera coordinates. The intrinsic camera matrix $K$ is given by

$$K = \begin{bmatrix} \phi_x & s & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

where $\phi_x$ and $\phi_y$ is the focal length in pixels in the $x$ and $y$ direction respectively, $s$ is the skew, and $\delta_x$ and $\delta_y$ is the central point in the image in pixels. The intrinsic camera matrix can be obtained either from the specification of the camera and lens, or from a camera calibration algorithm.

The rotation matrix ${}^{\text{c}}R_{\text{w}}$ and translation vector ${}^{\text{c}}t_{\text{w}}$ transform world coordinates into the camera coordinate system. The transformation of the camera in relation to the world coordinate system can then be found by the following equations [56]:

$${}^{\text{c}}R_{\text{w}} = ({}^{\text{w}}R_{\text{c}})^T \tag{3.4}$$

$${}^{\text{c}}t_{\text{w}} = -{}^{\text{c}}R_{\text{w}} \cdot\ {}^{\text{w}}t_{\text{c}} \tag{3.5}$$

${}^{\text{w}}t_{\text{c}}$ represents the camera position in the DSM and the altitude of the camera. Following the convention of the pinhole camera model, the $x$ axis points towards the right side of the camera, the $y$ axis towards the bottom, and the $z$ axis towards the front. I.e. the orientation of the camera coordinate systems agree when the camera is positioned flat on the ground at the sea level pointing north. ${}^{\text{w}}t_{\text{c}}$ is given by

$${}^{\text{w}}t_{\text{c}} = \begin{bmatrix} {}^{\text{w}}x_{\text{c}} \\ {}^{\text{w}}y_{\text{c}} \\ {}^{\text{w}}z_{\text{c}} \end{bmatrix} \tag{3.6}$$

where ${}^{\mathrm{w}}z_{\mathrm{c}}$ and ${}^{\mathrm{w}}x_{\mathrm{c}}$ are north and east respectively, as obtained by the algorithm provided by Karney [73], and ${}^{\mathrm{w}}y_{\mathrm{c}}$ is down, which is the same as the negative altitude of the camera.

${}^{\mathrm{w}}R_{\mathrm{c}}$ is obtained by multiplying the rotation matrices for roll, pitch, and yaw. In accordance with the formalism of the axes previously explained, ${}^{\mathrm{w}}R_{\mathrm{c}}$ is given by equation (3.7):

$$ {}^{\mathrm{w}}R_{\mathrm{c}} = R_y(\psi)R_x(\theta)R_z(\varphi) \tag{3.7} $$

where $\varphi$, $\theta$, and $\psi$ are the roll, pitch, and yaw angles of the camera respectively.

When the DSM skyline elements have been converted to image coordinates, the points that are outside the image are removed.

### 3.2.4 Estimating the attitude

When the camera image skyline has been obtained, it is possible to estimate the camera attitude by iteratively generating synthetic skylines and minimizing the distance-measure between the camera image skyline and synthetic skyline. This section describes how this is performed.

For a given roll estimate $\hat{\varphi}$, pitch estimate $\hat{\theta}$, and yaw estimate $\hat{\psi}$, a synthetic skyline is generated according to section 3.2.3. The distance-measure between the synthetic skyline and the camera image skyline, $e_{\mathrm{skyline}}$, is calculated as the mean of the Euclidean distance between the pixels in the synthetic skyline and the corresponding pixels in the camera skyline.

The yaw estimate is found using grid search. The yaw grid resolution and range can be varied during run-time, e.g. so the first estimate is global (0° to 360°) and subsequent estimates are local in order to decrease running time of the algorithm. For each yaw angle in the grid, the roll and pitch are estimated as described below, and $e_{\mathrm{skyline}}$ is calculated. The yaw angle with the lowest $e_{\mathrm{skyline}}$ in the grid is chosen as the yaw estimate. In order for this to work, the variations in the terrain need to be larger than the errors produced by the camera and by the approximations done in the algorithm. Hence the method needs unique features and is not expected to work with a flat skyline, e.g. in open waters. Note that the roll and pitch angles can still be estimated with a flat skyline.

The angle that covers one pixel in a camera system given in radians per pixel, $\alpha_{\mathrm{px}}$, can be calculated by

$$\alpha_{\text{px}} = 2\tan^{-1}\left(\frac{d_{\text{px}}}{2f}\right) \tag{3.8}$$

where $d_{\text{px}}$ is the physical dimension of the pixel in mm, and $f$ is the focal length in mm. The angle that covers one pixel describes the angle that a pixel represents in the world, i.e. if the camera turns $a$ radians, an object movement in the image can be approximated with $a \cdot \alpha_{\text{px}}$ pixels. Note that this approximation assumes that the sensor is spherical, introducing a small deviation when using a real sensor that is flat. The approximation error decreases as $a$ decreases.

The vertical angle between the mean of the camera image skyline and the synthetic skyline, $\hat{\theta}_{\text{error}}$, is thus given by:

$$\hat{\theta}_{\text{error}} = \alpha_{\text{px}} \cdot (\mu_{\text{synthetic}} - \mu_{\text{camera}}) \tag{3.9}$$

where $\mu_{\text{synthetic}}$ is the vertical mean of the pixels in the synthetic skyline, and $\mu_{\text{camera}}$ is the mean of the pixels in the camera image skyline. See figure 3.5 for an illustration. Using this, the pitch can be estimated by:

$$\hat{\theta}_i = \hat{\theta}_{i-1} - \alpha_{\text{px}} \cdot \mu_{\text{skylines error}} \tag{3.10}$$

where $\mu_{\text{skylines error}}$ is the mean of $e_{\text{skyline}}$ and $\hat{\theta}_{i-1}$ is the previous pitch estimate. The initial pitch estimate, $\hat{\theta}_0$, can be provided by e.g. an IMU estimate of the pitch or the user.

In order to estimate the roll error, the angle between the synthetic skyline and camera image skyline is calculated as the difference between the median angle between the both skylines and their vertical means, $\mu_{\text{camera}}$ and $\mu_{\text{synthetic}}$. See figure 3.5 for a visual description. The median is used rather than the mean in order to decrease the influence of outliers. The roll estimate can thus be calculated by

$$\hat{\varphi}_i = \hat{\varphi}_{i-1} - (\widetilde{\alpha}_{\text{synthetic}} - \widetilde{\alpha}_{\text{camera}}) \tag{3.11}$$

where $\widetilde{\alpha}_{\text{synthetic}}$ is the median of the angle between all the points in the synthetic skyline and its vertical mean, $\widetilde{\alpha}_{\text{camera}}$ is the median of the angle between the corresponding points on the camera skyline and its vertical mean, and $\hat{\varphi}_{i-1}$ is the previous roll estimate. The initial roll estimate, $\hat{\varphi}_0$, can be provided by e.g. an IMU estimate of the roll or the user.

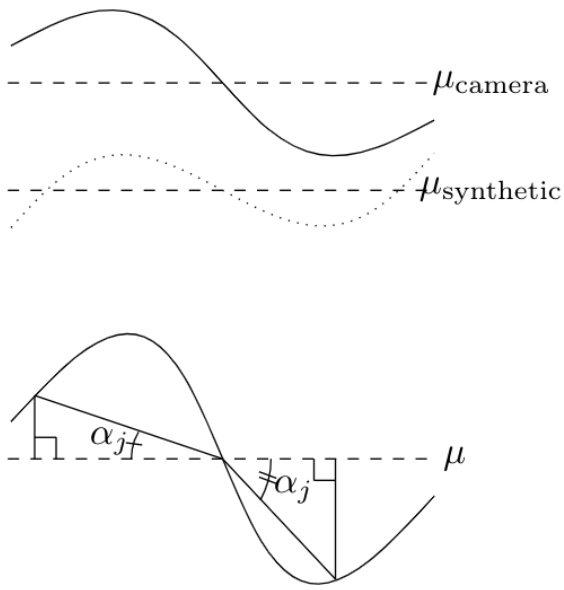


Figure 3.5: The geometry of the shape features used in the estimation algorithm. The upper figure shows the camera image skyline (solid) and synthetic skyline (dotted) and their respective vertical means $\mu_{\text{camera}}$ and $\mu_{\text{synthetic}}$ (dashed). The lower figure shows two angles $\alpha_j$ between two arbitrary points on the skyline mean $\mu$ (dashed) and the skyline (solid sinusoidal curve).

As the roll and pitch estimates will not produce an exact solution, but rather converge towards the true attitude, several iterations might be preferred in order to obtain better final estimates.

## 3.3 Experiment Setup

An experiment was performed in order to evaluate the algorithm. This section describes the hardware and data used, as well as considerations regarding the hardware and data.

### 3.3.1 Camera and PTU

The camera system used in the experiment is a Panasonic Lumix DMC-GF6 camera with a Panasonic Lumix G Vario 45-150 mm f/4.0-5.6 lens. More detailed specifications of the camera system are shown in Table 3.1.

The camera was mounted on a FLIR D48E Pan-Tilt Unit (PTU). The PTU has a resolution of 0.003° in pitch and 0.006° in yaw [41]. The geographic coordinates of

Table 3.1: Specification of the camera system

| | |
|---|---|
| **Camera** | Panasonic Lumix DMC-GF6 |
| **Resolution** | 4608 x 3464 px |
| **Pixel size** | 3.45 µm |
| **Lens** | Panasonic Lumix G Vario 45-150 mm f/4.0-5.6 |
| **Focal Length used** | 45 mm |

the camera were obtained by a GPS unit. A photo of the camera system mounted on the PTU is shown in Figure 3.6.


Figure 3.6: The camera system as mounted on the PTU in the experiment.

The camera is triggered wirelessly, and the whole process is automated. The PTU is set to rotate a certain angle in pitch and yaw, and then wait for the camera to capture a photo. The rotation of the camera for each photo taken in the experiment is shown in Figure 3.7.

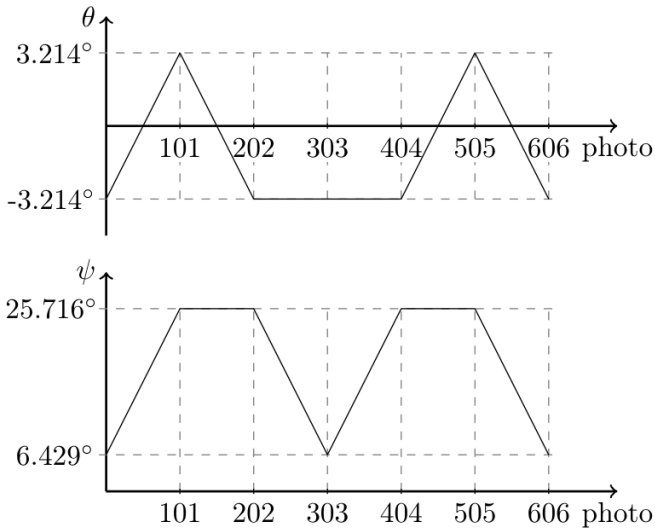


Figure 3.7: The pitch and yaw rotation of the PTU for each photo taken by the camera.

### 3.3.2 Digital Surface Model

The DSM used in the experiment was obtained from the national DOM10 dataset provided by the Norwegian Mapping Authority (Statens Kartverk) [132]. The relevant meta-data for the DSM are summarized in Table 3.2 [133].

Table 3.2: Metadata for the DOM10 (2017-03-27) dataset from the Norwegian Mapping Authority

| | |
|---|---|
| **Reference Frame** | ETRS89 |
| **Spheroid** | GRS 1980 |
| **Projection** | UTM |
| **UTM Zone** | 33N |
| **Central Meridian** | 15° |
| **Scale Factor** | 0.9996 |
| **False Easting** | 500,000 m |
| **False Northing** | 0 m |
| **Equatorial Radius** | 6,378,137 m |
| **Inverse Flattening** | 298.257222101 |
| **Grid resolution** | 10 m |

### 3.3.3 Calibration

In the experimental setup, the camera to PTU rotation ${}^{\mathrm{p}}R_{\mathrm{c}}$ is known, and the camera to world rotation ${}^{\mathrm{w}}R_{\mathrm{c}}$ will be estimated. In order to evaluate the results, the PTU to world rotation ${}^{\mathrm{w}}R_{\mathrm{p}}$ has to be estimated. The ${}^{\mathrm{w}}R_{\mathrm{c}}$ can then be obtained by

$$ {}^{\mathrm{w}}R_{\mathrm{c}} = {}^{\mathrm{w}}R_{\mathrm{p}} \, {}^{\mathrm{p}}R_{\mathrm{c}} \tag{3.12} $$

The camera system also need to be calibrated in order to eliminate deviations of the true camera parameters from the camera specification presented in the data-sheet. The unknown calibration parameters are those in the $K$ matrix in equation (3.3).

A possibility is to estimate $K$ and ${}^{\mathrm{w}}R_{\mathrm{p}}$ by modifying the attitude estimation algorithm to make ${}^{\mathrm{w}}R_{\mathrm{c}}$ fixed and the camera parameters the ones to be estimated. This approach is used by Grelsson et al, where the ground truth is available for some images in the experiment [47]. A similar but less precise method is to chose an image in the experiment, extract its skyline according to section 3.2.2, set the initial camera parameters according to the data-sheet and ${}^{\mathrm{p}}R_{\mathrm{c}}$ to the rotation used when the image was taken and then generate a synthetic skyline according to section 3.2.3. The parameters in $K$ and ${}^{\mathrm{w}}R_{\mathrm{p}}$ are manually adjusted in order to obtain the smallest error between the synthetic skyline and camera skyline. This calibration could be automated with e.g. grid search, but the number of parameters requiring adjustment would result in a long running time.

## 3.4 Results and Discussion

The data from the experiment consist of 600 photos. Six photos failed to be captured by the camera due to failures in the automated image capturing system. A sample blurred, gray-scale camera image is shown together with its full color version with the detected skyline marked in Figure 3.8.

The parameters obtained by the calibration process are summarized in Table 3.3.

The generated synthetic skyline when using the calibration parameters and the camera skyline for the first photo is shown in Figure 3.9.

For the first image the initial roll and pitch estimates, $\hat{\varphi}_0$ and $\hat{\theta}_0$, are set to zero. The grid for the yaw estimate is first set to $0^\circ$ to $360^\circ$ (global search) with an arbitrary resolution of $1^\circ$, then to $\hat{\psi} \pm 3^\circ$ with a resolution of $0.1^\circ$, and finally to $\hat{\psi} \pm 0.2^\circ$ with a resolution of $0.005^\circ$, where $\hat{\psi}$ is the yaw estimate from the previous

Figure 3.8: A sample camera image from the experiment. The top image shows the camera image converted to gray scale and with a Gaussian filter applied. The bottom image shows the camera image in its original color, with the camera skyline marked in red.

Table 3.3: Parameters from the camera intrinsic and extrinsic calibration

| | |
|---|---|
| $\phi_x$ | 43.7 mm / 3.75 µm/px |
| $\phi_y$ | 43.9 mm / 3.75 µm/px |
| $s$ | 0 |
| $\delta_x$ | 2304 px |
| $\delta_y$ | 1732 px |
| ${}^{\mathbf{w}}\varphi_{\mathbf{p}}$ | 0.23° |
| ${}^{\mathbf{w}}\theta_{\mathbf{p}}$ | -0.27° |
| ${}^{\mathbf{w}}\psi_{\mathbf{p}}$ | -41.18° |

grid. Using different grid sizes allows for a more accurate estimation at a shorter calculation time. The optimal grid size depends on the surrounding terrain, with a more varying terrain allowing the algorithm to converge for smaller grids.

Subsequent images use the roll and pitch estimate from the previous image as the initial estimates, and the two latter grid sizes and resolutions for the yaw estimate.

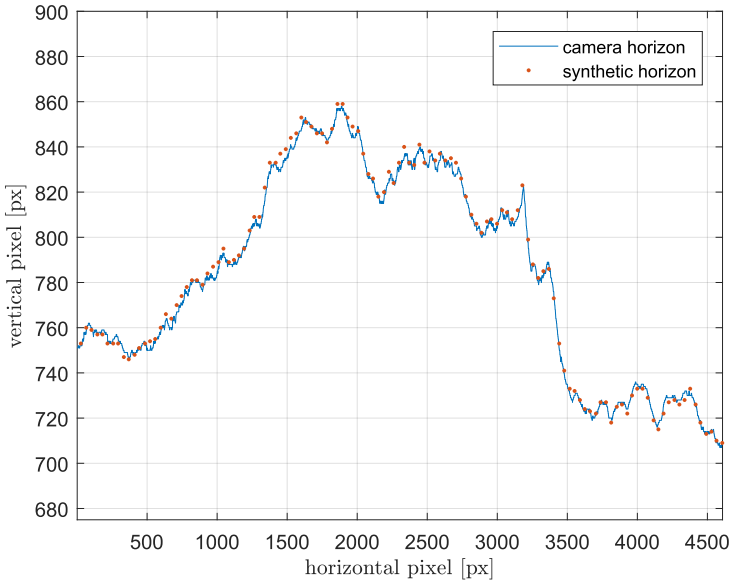


Figure 3.9: An example of a synthetic skyline (red dots) along with a skyline obtained from a camera image.

Only one iteration of the roll and pitch estimation was performed for each element in the yaw grid. A higher number might be required if the initial roll estimate has a difference from the true roll than in the experiment.

The parameters ${}^{\mathrm{w}}\varphi_{\mathrm{p}}$, ${}^{\mathrm{w}}\theta_{\mathrm{p}}$, and ${}^{\mathrm{w}}\psi_{\mathrm{p}}$ from Table 3.3 are used to convert the ground truth ${}^{\mathrm{p}}R_{\mathrm{c}}$, shown in Figure 3.7, to the ground truth ${}^{\mathrm{w}}R_{\mathrm{c}}$ according to equation (3.12). This ground truth is used to calculate the error of the estimated roll, pitch, and yaw angles for each image.

The resulting error of the algorithm is shown in Figure 3.10, and the standard deviations of the roll, pitch, and yaw angle errors are shown in Table 3.4. Only random errors – i.e. the standard deviations and not the mean errors – are reported in Table 3.4 as the mean errors depend on the accuracy of the calibrated ${}^{\mathrm{w}}R_{\mathrm{p}}$.

Table 3.4: Standard deviations of the roll, pitch, and yaw angle estimates

| | |
|---|---|
| $\sigma_{\varphi}$ | 0.037° |
| $\sigma_{\theta}$ | 0.015° |
| $\sigma_{\psi}$ | 0.018° |

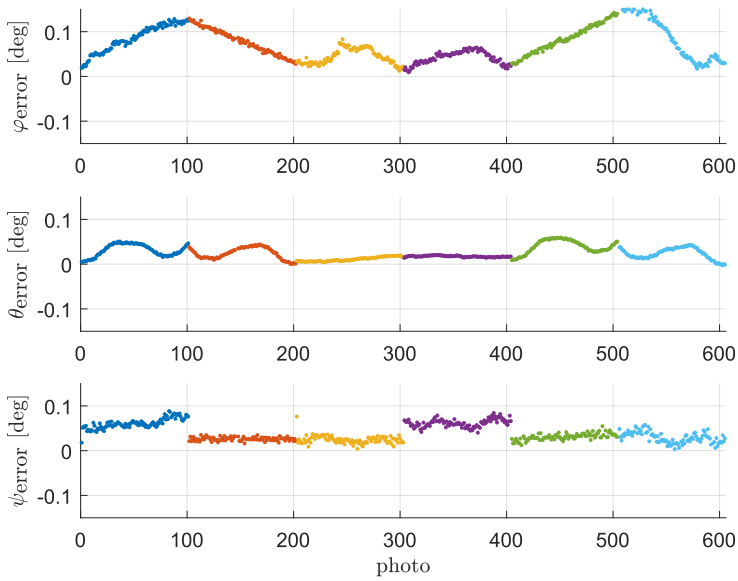


Figure 3.10: The errors of the estimated roll, pitch, and yaw angles for each photo. The different colors group together samples with the same roll and pitch steps.

The standard deviations of the errors are all low, which shows that the algorithm is able to estimate the roll, pitch, and yaw angles with a high precision. Figure 3.10 shows that the random errors are kept low, but there is a clear systematic error present. The two possible sources are the calibration procedure (section 3.3.3) and the approximation of the UTM projection as a flat grid (section 3.2.1). As can be seen, the largest error appear when the camera is moved in pitch, i.e. the error changes faster between photo 101 to 202 (where only the pitch of the PTU is varied) than between photo 202 to 303 (only the yaw is varied). During pitch movement, the skyline is only moved in the image; the DSM elements represented in the synthetic skyline changes very little. This suggests that the camera calibration, possibly the lens distortion, is a main cause of the systematic errors.

The results presented use a $n_{\text{lines}}$ of 2000 to find the skyline elements in the DSM. Using a higher $n_{\text{lines}}$ than 2000 did not result in lower standard deviations for the performed experiment, but might differ for other types of terrain and different camera system parameters.

The average running time of the algorithm, excluding the detection of the skyline in the camera images was 63 ms per image. The algorithm was implemented in Matlab, and run on a Intel Xeon E3-1535M. This indicates that it can run in real-

time if a more optimized implemention in a lower level programming language were to be used, even on an small computer such as the Nvidia Jetson.

The experimental data has shown that precision-wise, the algorithm performs similar to, or slightly better than those presented by Dumble & Gibbens [34] and Grelsson et al [47]. The benefit of the algorithm compared to the other algorithms is the global yaw angle estimate – a feature that is often required in real world applications. While the algorithm showed robustness and converged for all the images in the experimental data set, it should be noted that a proper segmentation of the sky and land or water is important in order to avoid larger errors and to ensure convergence.

## 3.5 Future Work

Future work includes a more robust skyline detection algorithm to find the skyline in the camera image, and a more accurate algorithm to generate the synthetic skyline. Additionally, the estimates provided by the algorithm can be used in an attitude estimation filter as reference vectors due to their high precision.

A more robust method for finding the skyline in the camera image could improve the robustness significantly. One example of a more robust method which uses a neural network is presented by Porzi et al [106]. This method could also be used to determine how well defined the skyline is in the camera image, which could be used as an indicator of the accuracy of the estimate. Determining the drawing distance of the synthetic skyline is a similar problem, which is particularly important in areas susceptible to fog or haze. A possible solution could be to calibrate the algorithm to see which drawing distance produces the smallest error.

In order to reduce the systematic error, a proper calibration algorithm should be used, e.g. by performing an experiment where the true angles are known, and using e.g. the method described by Grelsson et al [47]. Another method could also be used to render the synthetic skyline in order to get rid of errors associated with estimating the DSM as a flat grid. For example could a rendering engine such as OpenGL be used to do a proper conversion from the UTM projection to three-dimensional world coordinates, and then render the synthetic skyline. The method could also be used in an attitude estimation filter, e.g. to compute the reference vectors in a non-linear complementary filter, which could utilize its benefits of being highly accurate but not as robust as e.g. gyroscopes.

## 3.6 Conclusions

We have presented an algorithm for estimating the attitude of a camera using the extracted skyline from a camera image and synthetic skylines generated from a Digital Surface Model. The algorithm uses geometrical properties to estimate the roll and pitch angles, and a grid search to estimate the yaw angle.

Using experimental data, the algorithm is shown to estimate the roll, pitch, and yaw angles with standard deviations of $0.037°$, $0.015°$, and $0.018°$ respectively. The estimation algorithm is running at 16 Hz in the current implementation, which can be improved by using a lower level programming language. The results are on par with current DEM based attitude estimation algorithms, while also estimating the yaw angle globally.

# Chapter 4

# Accuracy of Sea Ice Floe Size Observation from an Aerial Camera at Slant Angles

The importance of measuring the size of ice floes in e.g. marine navigation and environmental sciences has made it a frequently performed procedure. When real-time data is required, images from a camera on-board an aerial vehicle or mounted on a marine vessel is commonly preferred over satellite images. Their lower fields of view can be improved by tilting the cameras to capture images of a larger area. However, this introduces a greatly changing ground resolution within the same camera image, which makes size estimation a more complex task. It is nevertheless performed in several methods to estimate the size of ice floes. In this chapter, ice floe size estimation is evaluated for different scenarios when using an aerial camera at slant angles. In order to reduce errors caused by automatic image segmentation and attitude estimation algorithms, the methods are aided by human input. The estimates are performed on real world data captured during the Statoil Station Keeping Trials in the Bothnian Bay during March 2017. The results conclude that the major challenge is to guarantee separation between ice floes in the camera images, which is something that requires both enough ground resolution and a suitable image segmentation algorithm.

## 4.1 Introduction

Floating ice on the ocean surface occupies approximately 7% of the surface area of the world oceans [144], and comes in different shapes such as icebergs, which have been broken off from a glacier or an ice shelf, and ice floes, which are pieces of sea ice varying in size from a few meters to tens of kilometers across [12].

Sea ice has a major effect on weather, climate, and ocean currents [126], and can also significantly obstruct navigation in places such as the Northern Sea Route [69]. Both smaller and larger ice formations undergo large changes throughout a year [103]. This makes continuously monitoring sea ice a vital part in a number of fields.

There exist a number of methods for monitoring ice floes. Haugen et al. [57] provides an overview of existing sensors and sensor platforms for ice management. Using a satellite as a sensor platform allows for very high quality sensors and a large field of view from cameras and radars carried by the satellite. However, due to their high altitude, the spatial resolution is limited. Satellite images can also be costly, and the updates are slow. Additionally, some sensors carried by satellites can not properly sample the ground environment during certain weather conditions such as clouds. An alternative to a satellite is a high-altitude aircraft. An aircraft can carry many of the same sensors as a satellite while operating at a lower altitude, which allows for higher spatial and temporal resolution than the satellite. Using a high-altitude aircraft as a sensor platform is however expensive, in particular in remote areas such as the Arctic or when continuous updates of the environment are required. If the goal is to obtain sea ice data around a marine vessel, a solution is to attach the sensors to the vessel itself. An example of this is marine radars used for navigation and collision avoidance. The data from the sensors come in real-time, and have a high spatial resolution for areas near the vessel. However, the area covered by shipborne sensors is often small. A similar scenario is when the sensors are placed on land, which has similar benefits and limitations.

In between the high-altitude airborne sensors and the shipborne sensors are the recent emergence of Unmanned Aerial Vehicles (UAVs) as a sensor platform when gathering ice data – in particular UAVs small enough to take off from and land on a marine vessel [57][81]. These UAVs often operate at a lower altitude, allowing for real-time data at a high spatial resolution, even though the weight of the sensors are limited. The sensors carried by UAVs can cover a higher area than those carried by the marine vessel, but not quite as big as a satellite or a high-altitude aircraft. A similar sensor platform is an aerostat moored to a marine vessel. The aerostat can not move around like a UAV in order to cover a larger area, but it benefits from a higher altitude than the sensors attached to a marine vessel and requires little supervision. A camera mounted to an aerostat or a small UAV is therefore suitable when monitoring sea ice in an area near a vessel, but the spatial coverage of shipborne sensors are not enough.

In order to be able to increase the field of view further of a camera attached to a marine vessel, a UAV, or an aerostat, the camera can be tilted. If the tilt angle and other camera parameters are known, the pinhole camera model [108] can be used to find the world coordinates of a pixel in the camera image. Several methods use this to rectify camera images and measure the size of ice floes [154][90]. Since the image capturing process is complex, in particular with greatly varying distances in

an outdoor environment such as that of a tilted camera measuring ice floe sizes, it is of interest to evaluate how well it is possible to estimate the ice floe size when the camera is tilted at slant angles.
A method developed by Lu & Li [90] for obtaining the size of ice floes from camera images when the camera is tilted was evaluated by the same authors against a more direct method, by comparing the ice floe distributions calculated by both methods. However, the authors only evaluates their simplified method, and not the effects of having a camera at a slant angle. A qualitative evaluation of size estimation for different camera-to-object angles (see section 4.2.3 for the definition of camera-to-object angle) could provide guidelines for when a good ice floe size estimate can be expected.
In this chapter, a method for estimating the dimensions of individual ice floes is presented. The main goal of the method is to be used in evaluating the size estimation of individual ice floes seen at different camera-to-object angles. Manual methods with high accuracy are therefore preferred over automatic methods, in order to reduce errors caused by e.g. image segmentation errors. The method is applied to experimental data from the Statoil Station Keeping Trials performed in the Bothnian Bay during the first three weeks of March 2017 [1] [86]. The results are evaluated and discussed in order to find how well the size of ice floes can be estimated for different scenarios when a large camera-to-object angle is used.

## 4.2 Methodology

This section contains explanations of the methods used in this chapter for comparing the dimension estimates of an ice floe appearing in multiple camera images at different camera-to-object angles. Figure 4.1 shows an overview of the methods. Lastly, the size estimation error that can be explained by a difference in ground resolution is explained.

### 4.2.1 Finding the pixels representing an ice floe

In order to find the pixels representing an ice floe in a camera image, the image is first segmented. Segmentation divides an image into regions or objects [45]. In this case, the ice floes are separated from other objects such as water, brash ice, and other ice floes. Errors made by the image segmentation algorithm, such as dividing one ice floe into two segments, or missing a border between two ice floes labelling them as the same segment, can have a big impact on the size estimate.
Even though image segmentation methods have existed for a long time, humans still outperform commonly available algorithms [143]. Recent algorithms have started to outperform humans [50], but human performance in image segmentation is still often used as the ground truth when evaluating algorithms. In this chapter, human

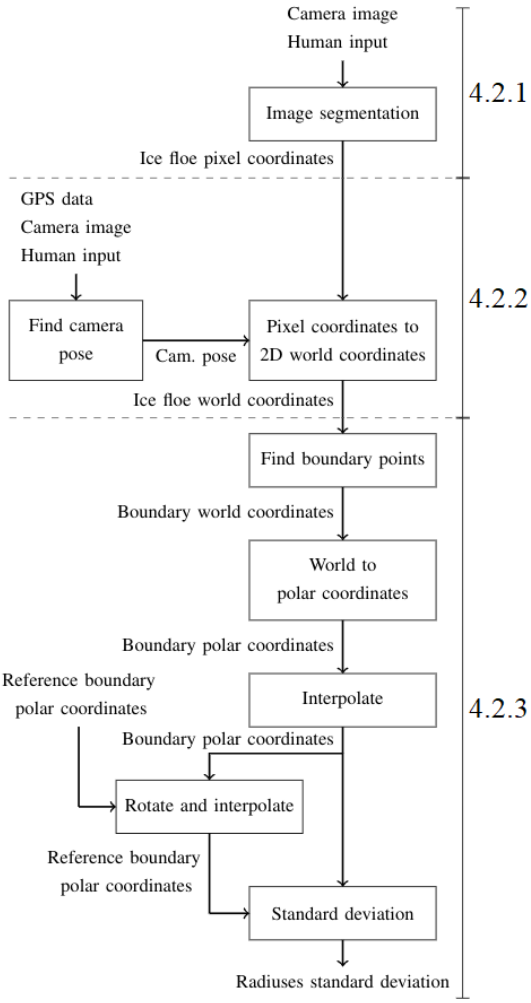


Figure 4.1: Overview of the size estimation and evaluation algorithm. The rightmost text indicates which section contain that details about each part.

image segmentation aided by flood-fill algorithm is used to find the pixels representing an ice floe in an image due to its simplicity.
The flood-fill algorithm selects connected pixels with an intensity within a certain threshold from that of the starting pixel [21], and is run after the image has been converted to gray-scale. After the flood-fill algorithm has been run, a human adjusts the segmentation by manually adding or removing regions of pixels to properly represent the area occupied by the ice floe in the camera image.

### 4.2.2 Image to world coordinates

The pinhole camera model [108] is used to model the projection of world coordinates onto an image plane. The projection is given by

$$\lambda \begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \\ 1 \end{bmatrix} = \begin{bmatrix} K \mid 0 \end{bmatrix} \begin{bmatrix} {}^{\text{c}}R_{\text{w}} & {}^{\text{c}}t_{\text{w}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\text{world}} \\ y_{\text{world}} \\ z_{\text{world}} \\ 1 \end{bmatrix} \tag{4.1}$$

where $x_{\text{image}}$, $y_{\text{image}}$, and $p_{\text{world}} = [x_{\text{world}}\ y_{\text{world}}\ z_{\text{world}}]^T$ are the pixel coordinates and world coordinates of the projected point respectively, $\lambda$ is a scaling factor, $K$ is the intrinsic camera matrix obtained by camera calibration, and ${}^{\text{c}}R_{\text{w}}$ and ${}^{\text{c}}t_{\text{w}}$ are the rotation matrix and translation vector from world coordinates to camera coordinates respectively.

The world coordinate system is in this case defined as the local tangent plane of the Earth at the location of the camera at sea level, with the $x$-axis pointing east, the $y$-axis pointing down, and the $z$-axis pointing north. By solving (4.1) for $x_{\text{world}}$ and $z_{\text{world}}$, it is possible to obtain the world coordinates of a point if its pixel coordinates and world coordinate $y_{\text{world}}$ are known. The change in $y_{\text{world}}$ due to the curvature of the Earth and the altitude of the ice floes are disregarded, and $y_{\text{world}}$ is thus set to zero.

The pinhole camera model assumes that the rotation matrix and translation vector from world coordinates to camera coordinates are known. In this chapter the exact world position of a point represented by a pixel in the camera image is not required, but rather the distance from the camera to the point. Rotating the camera around its $y$-axis (yaw) or moving the camera in the $xz$-plane will not change this calculated distance. This means that there is only a need to estimate the camera altitude, ${}^{\text{c}}t_{\text{w,y}}$, and the camera rotations around its $x$- and $z$-axis (pitch and roll).

The camera altitude is estimated using a GPS unit. The roll and pitch are estimated by projecting horizon points onto the camera image using (4.1), and then manually adjusting the roll and pitch until the horizon points match the horizon in the camera image. The method for finding the world position of horizon points is described in section 4.2.4. See figure 4.2 for an example of matching horizon points. For a more in-depth study and theory of matching a synthetic horizon with the horizon in a camera image, see [117] or chapter 3 of this thesis.

### 4.2.3 Camera-to-Object angle

The camera-to-object angle, $\alpha_{c-o}$, is defined as the angle between a vector pointing straight down from the camera, $\mathbf{a}_{\text{c-g}}$, and a vector pointing from the camera towards the object, $\mathbf{a}_{\text{c-o}}$. See figure 4.3 for a visual description.

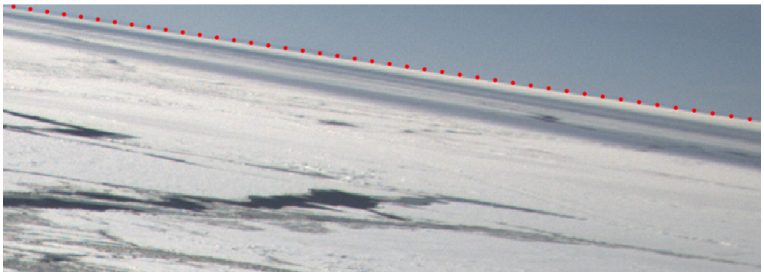

Figure 4.2: The horizon points (red dots) matched with the horizon in a camera image.

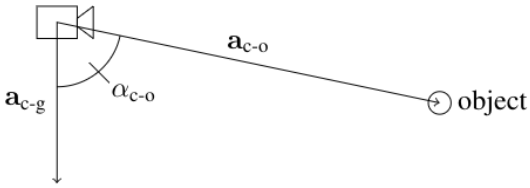


Figure 4.3: The camera-to-object angle, $\alpha_{\text{c-o}}$.

The camera-to-object angle can be calculated according to

$$\alpha_{\text{c-o}} = \cos^{-1}\left(\frac{\mathbf{a}_{\text{c-g}} \cdot \mathbf{a}_{\text{c-o}}}{||\mathbf{a}_{\text{c-g}}||\ ||\mathbf{a}_{\text{c-o}}||}\right) \tag{4.2}$$

### 4.2.4 Horizon Points

The first step in finding a horizon point is to find the angle between the camera and the horizon, $\gamma$, which is given by

$$\gamma = \cos^{-1}\left(R/(R+a)\right) \tag{4.3}$$

where $R$ is the radius of the Earth, and $a$ is the camera altitude. See figure 4.4 for the geometrical derivation of (4.3). Note that this approximates the oblate spheroid shape of the Earth with a spherical shape.

When the angle $\gamma$ has been acquired, the $y_{\text{world}}$ and $z_{\text{world}}$ position of the horizon, $y_{\text{world horizon}}$ and $z_{\text{world horizon}}$, can be acquired according to

$$\begin{aligned} y_{\text{world horizon}} &= R(1-\cos\gamma) \\ z_{\text{world horizon}} &= R\sin\gamma \end{aligned} \tag{4.4}$$

See figure 4.5 for the geometrical derivation of (4.4). The $x_{\text{world}}$-coordinate of the horizon point, $x_{\text{world horizon}}$, is simply set to zero. $x_{\text{world horizon}}$ and $z_{\text{world horizon}}$ are

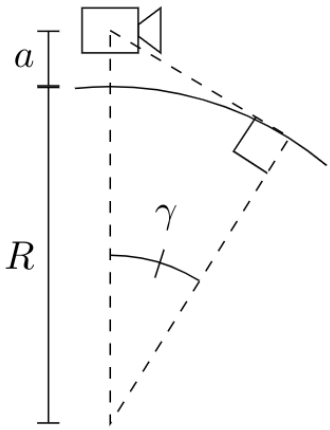


Figure 4.4: The geometric derivation of the angle between the camera and the horizon, $\gamma$.

then rotated around the $y$-axis with equidistant angles in order to obtain the world coordinates of several horizon points.

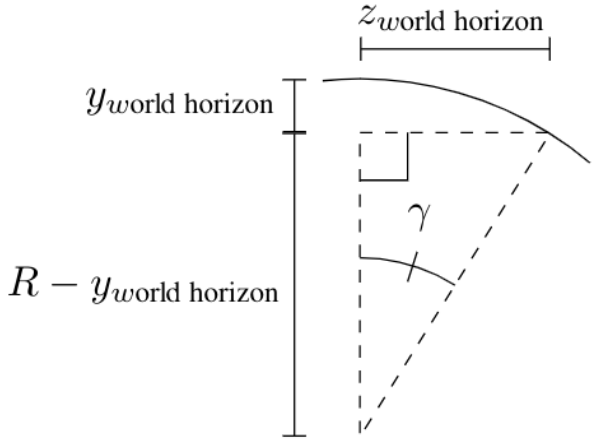


Figure 4.5: The geometric derivation of the $y_{\text{world}}$ and $z_{\text{world}}$ position of the horizon.

### 4.2.5 Comparing ice floe size estimates

The aim of this chapter is to evaluate the accuracy of the size estimates of an ice floe for different camera-to-object angles. The evaluation is done by comparing the size estimate of an ice floe with its reference. The reference is the size estimate of the

same ice floe at the lowest camera-to-object angle available in the camera image set, i.e. it is the estimate least prone to errors. In order to make a quantitative evaluation of the results easier, a single metric of the accuracy will be used. It is possible to use e.g. the volume, but this can be misleading since vastly different shapes can result in the same volume. Instead, the standard deviation of the radiuses of an ice floe in polar coordinates from the reference is calculated as the error metric, $\sigma_{\text{radiuses}}$. This guarantees a larger error when the deviation from the reference is larger overall, and can be used in e.g. statistical simulations.

The standard deviation $\sigma_{\text{radiuses}}$ will be evaluated against the camera-to-object angle, $\alpha_{\text{c-o}}$, for each ice floe. This is calculated as the mean camera-to-object angle for each $x_{\text{world}}$ and $z_{\text{world}}$ coordinate of the points representing the ice floe in the camera image.

After the $x_{\text{world}}$ and $z_{\text{world}}$ coordinates of the points representing the ice floe in the camera image have been obtained according to section 4.2.1 and 4.2.2, they are mean centered.

An alpha shape [36] is then generated for the points in order to find a natural border of the ice floe. An alpha shape is a shape that encloses a set of points by defining the boundary points as those which can be touched by empty circles (i.e. circle not containing any points) with radiuses $\alpha$. See figure 4.6 for a visual description of the alpha shape algorithm. The alpha shape boundary points are

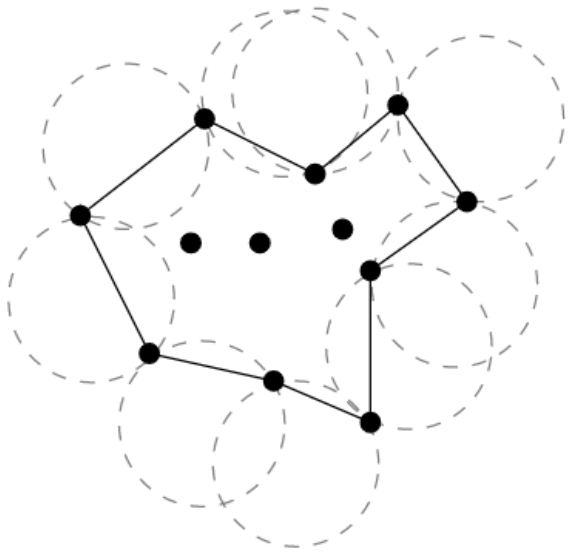

Figure 4.6: The alpha shape algorithm visualized. The filled points represent the points, the grey dashed circles represent the empty circles connected to two points each, and the lines indicate which boundary points the alpha shape goes through.

selected as the boundary points of the ice floe. The ice floe boundary points are transformed from Cartesian to polar coordinates according to

$$\begin{aligned} r_b &= \sqrt{x_{\text{world}}^2 + z_{\text{world}}^2} \\ \alpha_b &= \arctan\left(\frac{x_{\text{world}}}{y_{\text{world}}}\right) \end{aligned} \tag{4.5}$$

where $r_b$ is the radius and $\alpha_b$ is the polar angle.
Since the ice floe will be rotated relative to the camera between camera images, and the resolution of the polar angle will change, the calculated radiuses of an ice floe in one camera image can not be directly compared with the radiuses of the ice floe in another image. In order to make them comparable, radiuses for predefined equidistant polar angles are found using piecewise cubic Hermite interpolation [31]. After the interpolation, the polar angles of the reference ice floe are shifted in order to find the polar angles which produces the lowest standard deviation $\sigma_{\text{radiuses}}$. This is equivalent to rotating one ice floe to match the other ice floe.

### 4.2.6 Ground Resolution Error

The ground resolution is the number of pixels per square meter when projected onto the sea surface. In order to find what accuracy is possible, the error that can be explained by the decrease in ground resolution as the camera moves away from the object – the ground resolution error, $\sigma_{\text{radiuses gr}}$ – is calculated for each ice floe in each camera image. This is done by shifting the mean of the world points of the reference to the mean of the ice floe being evaluated; projecting it onto the image coordinate system using (4.1); rounding the image coordinates to their closest integer values to simulate the precision lost; and then calculating the standard deviation $\sigma_{\text{radiuses}}$ using the same methods as for the ice floe being evaluated.
The ground resolution error is not the theoretical minimum error for the given camera-to-object angle and ground resolution. Rather, it is the approximation of how large part of the error that can be attributed to the ground resolution for the current image.

## 4.3 Experimental Setup

Experiments were performed in order to evaluate the accuracy of size estimates of ice floes using a camera at slant angles using the method described in section 4.2. The experiments took place during the Statoil Station Keeping Trials (SKT) in the Bothnian Bay during the first three weeks of March 2017. The SKT involved two

vessels in operation, Magne Viking and Tor Viking.
The camera system was mounted on the moored balloon system OceanEye made by Maritime Robotics, which in turn was attached to Magne Viking. The OceanEye allowed for the camera to stay at an altitude of around 120 meters throughout the day.

### 4.3.1 Camera System

The camera system consisted of an EO-camera, a lens, a GPS unit, and an on-board computer. The system was designed to capture images with low distortion, high time synchronization accuracy, and to be light weight. The details of the camera unit and lens can be found in table 4.1.

Table 4.1: Specification of the camera system

| | |
|---|---|
| **Camera** | FLIR CM3-U3-31S4C-CS |
| **Resolution** | 2048 x 1536 px |
| **Pixel size** | 3.45 µm |
| **Lens** | Kowa LM8JC10M |
| **Focal length** | 8.5 mm |
| **Lens distortion** | 0.31% |

### 4.3.2 Scenarios

The camera was capturing images at 1 Hz for about eight hours per day. Since the process of estimating the dimensions of an ice floe in a camera image consists partly of manual work, the evaluation was limited to a camera image sequence of ten images, taken eight seconds apart. The sequence was chosen to be when Magne Viking was moving, in order to capture the same ice floes in multiple camera images at different camera-to-object angles. In the image sequence, five ice floes of different sizes were chosen. See figure 4.7 for a cropped camera image containing the five ice floes.

## 4.4 Results and Discussion

This section presents and discusses the experimental results together with the calculated theoretical errors. First, each step of the methods in section 4.2 is shown, followed by an analysis of the error metric, $\sigma_{\text{radiuses}}$.
The pixels are first selected using a flood-fill algorithm, and then adjusted by manually adding and removing pixel regions. See figure 4.8 for an example of this process.

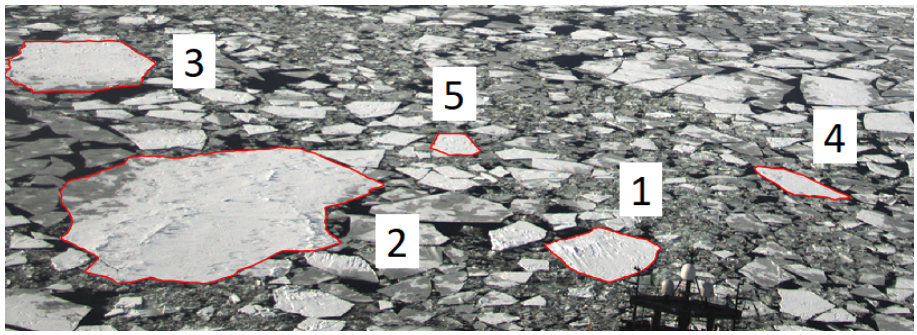


Figure 4.7: A cropped camera image containing the five ice floes being analyzed.

After the ice floe pixels have been selected, their corresponding position in the world coordinate system are found, assuming an ice floe altitude of zero. The mean is subtracted from the points in both dimensions, $x_{\text{world}}$ and $y_{\text{world}}$, and an alpha shape around the points is calculated. See figure 4.9 for the mean-centered points in the world coordinates, together with the alpha shape.

The points connected to the border of the alpha shape are then transformed into polar coordinates, and interpolated using piecewise cubic Hermite interpolation for polar angles in the interval $[0°, 360°)$, with a resolution of 1°. See figure 4.10 for the interpolated border points in polar coordinates.

The same methods were applied to the ice floe in the reference camera image. The ice floe reference was then rotated to match the current points of the ice floe being evaluated. See figure 4.11 for a comparison of the ice floe points being evaluated in Cartesian coordinates, the reference points of the ice floe, and the rotated reference points.

The distance and camera-to-object angle to the ice floe being evaluated is in this case 623 m and 81.4° respectively, and 262 m and 70.1° respectively for the reference. The standard deviation $\sigma_{\text{radiuses}}$ is 0.40 m, and $\sigma_{\text{radiuses gr}}$ is 0.33 m.
The same algorithm is then applied to the five ice floes in the selected camera image set. See figure 4.7 for the numbered ice floes in the camera image with the smallest camera-to-object angle. The resulting error metric $\sigma_{\text{radiuses}}$ and ground resolution error metric $\sigma_{\text{radiuses gr}}$ are then presented in figure 4.12. The ice floe number in the top left corner of each graph corresponds to the ice floe numbers in figure 4.7.

The errors generally increase with an increased camera-to-object angle, as was expected. There is little to be discussed for ice floe 1 – the error generally increases with an increased camera-to-object angle, and stays slightly above the ground resolution error. The difference between the error and the ground resolution error

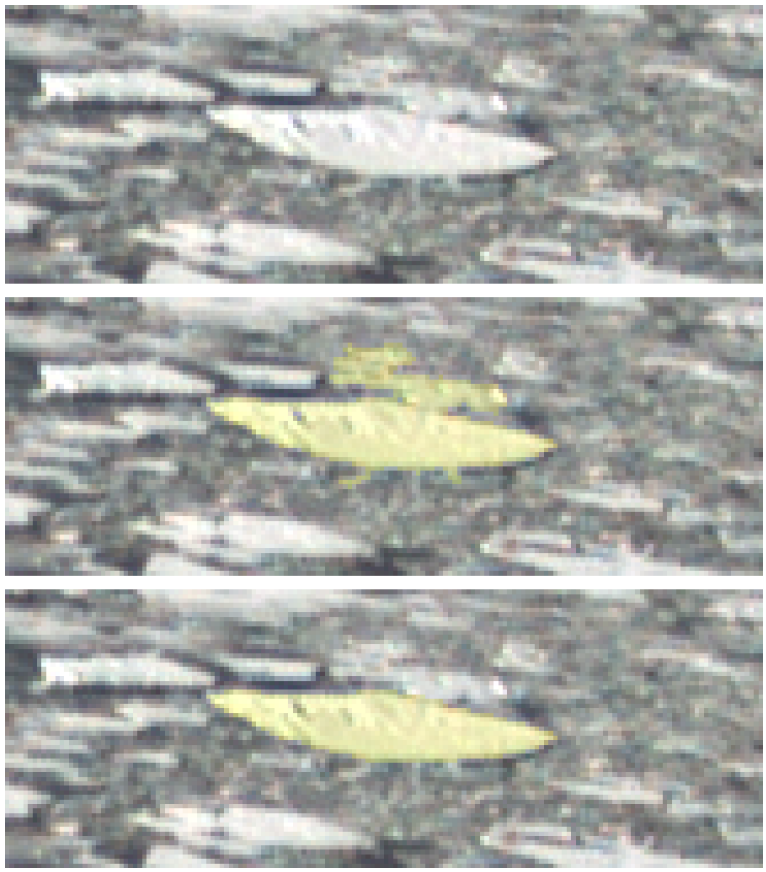

Figure 4.8: The pixel selection process. The top image shows the raw camera image. The center image shows the pixels selected by the flood-fill algorithm in yellow. The bottom image shows the final results after after manually adjusting the pixels selected by the flood-fill algorithm.

might be explained by inconsistency in the pixel selection process and remaining errors from the camera calibration. The error for ice floe 4 sometimes dips below the ground resolution error. This might happen when the ice floe in the camera images line up well with the pixel edges, or when an error has occurred when selecting the ice floe pixels in the reference image.

The error for ice floe 5 makes a sudden jump between $\alpha_{\text{c-o}}$ 76.5° and 77.8°, and a similar phenomena can be seen for ice floe 3 between 82° and 82.5°. By analyzing the camera images, it can be seen that this is because the border between nearby ice floes can no longer be seen, making them appear as one ice floe. See figure 4.13 for a comparison between the reference image and the fourth image (camera-to-object angle of 78.8°) of ice floe 5, where the loss of separation between the ice floe

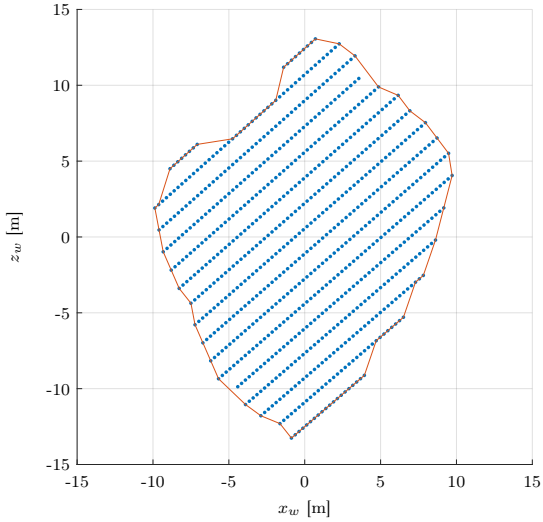


Figure 4.9: The corresponding mean centered world coordinates of the selected pixels (blue dots) and the alpha shape (red curve).

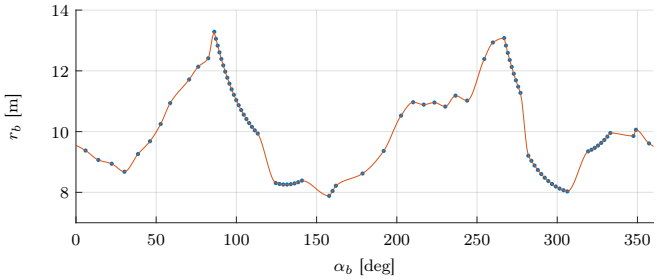


Figure 4.10: The border points in polar coordinates (blue dots), and a curve going through the interpolated points (red line).

and a nearby ice floe can be seen.

The large error produced by diminishing borders indicate that the separation between ice floes are of high importance when estimating the size of ice floes.

The errors for ice floe 2 and 3 are overall quite large. Looking at figure 4.7, it can be seen that these ice floes are larger than the others, both contain darker areas, and both are near brash ice. This pose a problem to both the flood-fill algorithm and to human segmentation since it is more difficult to manually correct minor errors for large ice floes, and ice floes with poorly defined borders.

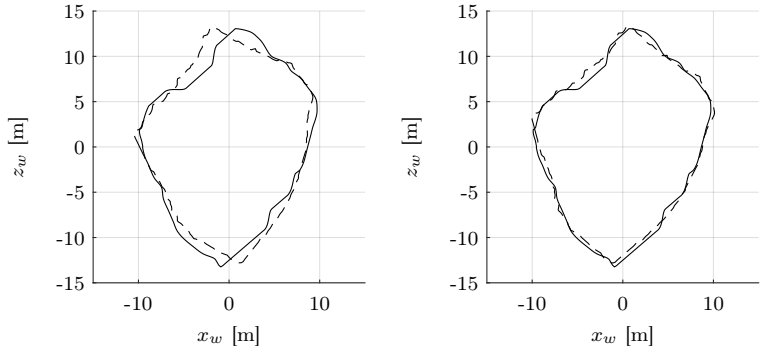


Figure 4.11: The ice floe in the camera image being evaluated versus in the reference image without any rotation (left). The ice floe in the camera image being evaluated versus in the reference image rotated to match the first (right).

## 4.5 Conclusion

In this chapter, a method for estimating the size of individual ice floes has been presented, as well as a metric to evaluate the accuracy of the size estimation. The method, which was designed to minimize the errors caused by automatic image segmentation and attitude estimation algorithms in order to find the limitations of the camera system when estimating the size of ice floes at slant angles, is applied to experimental data collected during the Statoil SKT performed in the Bothnian Bay in March 2017. During the experiment, the camera was moved in relation to the ice floes in order to capture camera images of the same ice floes at different distances, which allowed for the evaluation of ice floe size estimation for different camera-to-object angles. Evaluation of the resulting errors and the camera images conclude that the size of an ice floe can be estimated with a high accuracy if some criteria, which allows for the ice floes to be accurately segmented in the camera images, are met. This makes it important to take into account not only the size estimation accuracy of an individual ice floe when designing a camera system for a certain ground resolution, but also the separation between ice floes.
Potential future work includes an evaluation of more automatic image segmentation and size estimation algorithms for use when estimating the size of ice floes. The camera images could also be fused with e.g. Synthetic Aperture Radar (SAR) images in order to find details not visible in camera images. The geometrical calculations can also be made more rigorous by e.g. taking into account the effect of ice ridges on the ice floes, which might affect the results at a large camera-to-object angle.

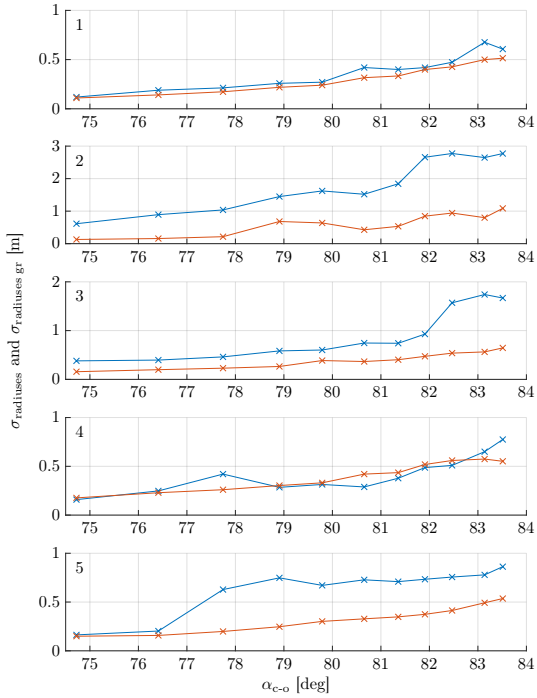


Figure 4.12: The resulting standard deviations $\sigma_{\text{radiuses}}$ (blue line) and $\sigma_{\text{radiuses gr}}$ (red line) versus the camera-to-object angle. The number in the top left corner of each graph indicates which ice floe is being evaluated, see figure 4.7.

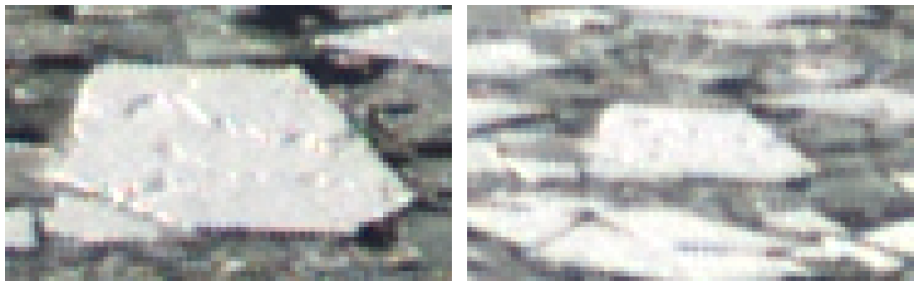

Figure 4.13: Ice floe 5 in the reference camera image (left) and with a camera-to-object angle of 77.8° (right).

# Chapter 5

# Detectability of Objects at the Sea Surface in Visible Light and Thermal Camera Images

In a number of ocean surveillance and remote sensing applications, visible light and thermal cameras are used to detect and identify objects at the sea surface. Knowing beforehand what the camera can detect or not can be important, yet highly difficult to determine. Optical models such as Modulation Transfer Functions can help in evaluating a camera system, but requires a deeper knowledge in optics, and detailed specifications of each component. The models also does not handle noise coming from the scene background, which in many cases is the major limiting factor of detectability. In this chapter, we evaluate the results of an edge detection algorithm on images from two commercial off-the-shelf camera system – one visual light and one thermal. We then draw conclusions on the detectability of objects which commonly needs to be detected at the sea surface.

## 5.1 Introduction

In ocean surveillance and remote sensing applications, the first step in various computer vision algorithms, is to detect the object in the camera image. Practical examples include sea ice detection and monitoring for situational awareness and environmental research [113], and detecting marine vessels and people in search and rescue missions [79] [135].

Two common sensors used in maritime surveillance and remote sensing missions are visible light and thermal cameras. Visible light cameras commonly provide a high spatial and temporal resolution image sequence of the scene in the visible light

spectrum, which makes it suitable to detect very small objects, and objects slightly below the sea surface since the visible light penetrates water relatively well [38]. Thermal cameras commonly provide a lower spatial and temporal resolution image sequence in the thermal spectrum. This makes thermal cameras suitable for detecting objects with a different thermal footprint than the sea surface, in particular during low visible light conditions. Since the sea surface radiance is more homogeneous in the thermal spectrum than the visible light spectrum, the segmentation of objects at the sea surface is often also simpler in thermal camera images than visible light camera images.

Determining beforehand what a camera system can detect can prove difficult. Johnson's criteria [119] is still commonly used as the theoretical basis for object detectability based on the number of line pairs (or pixels) that an object represents in a camera image. Borghgraef et al. [15] evaluated different methods for detecting objects at the sea surface, where a high detection accuracy can be obtained if the background is uniform. This can also be seen in experimental results in e.g. [79] and [152]. Some difficulties mentioned, however, are highly dynamic backgrounds and a large camera to object angle, which causes objects to partly disappear behind ocean waves. It also makes it more difficult to detect flat objects, as the area of the object projection on the image sensor decreases with an increased angle. The work by Borghgraef et al. uses a stationary camera at a low altitude, which poses different challenges than a moving airborne camera. Bloisi et al. [13] similarly uses a stationary Pan-Tilt-Zoom visual light camera in order to evaluate the detectability. The experimental data contain camera images during different lighting conditions and camera angles, however these are not evaluated in regards to the detectability of the objects.

In this chapter we aim to evaluate the detectability of common objects at the sea surface as they appear in images captured with two commercial off-the-shelf cameras – one visual light and one thermal – in order to evaluate how well objects can be detected compared to the theoretical geometric limit. Special emphasis is put on how the detectability varies with the distance and angle between the camera and objects. The detectability metric is based on the performance of an edge detection algorithm, which commonly forms the basis for more advanced computer vision algorithms [105]. Although a number of specialized edge detection algorithms have successfully been used to detect objects at the sea surface, e.g. by Can et al. [23] and Zhang and Skjetne [154], this chapter uses a Sobel filter for edge detection due to its general and common usage.

## 5.2 Methodology

The aim of the methodology is to find how well objects can be detected in the camera images. Edge detection accuracy is chosen as the detectability metric due to that many higher level algorithms depend on the accuracy of the edge detection, such as object recognition and image segmentation [105]. This section first describes three general initial steps in edge detection algorithms – converting the image to grayscale, reducing the noise, and calculating the gradient of the image. These steps are performed in e.g. the Canny edge detection algorithm, which remains one of the most commonly used algorithms for edge detection [99]. Finally the edge detection algorithm, the method for determining the position of the objects in the camera images, the camera to object distance and angle calculations, the detectability metric, and the edge detection algorithm parameter tuning are explained.

### 5.2.1 Grayscale Conversion

The images from the visible light camera are in RGB (Red-Green-Blue) format, i.e. each pixel contains intensity data for the red, green, and blue spectra separately. Since the subsequent methods work with one intensity value per pixel, the images first need to be converted to grayscale. Four different methods are considered for grayscale conversion: an average of the three color spectra, only using the red spectrum, only using the blue spectrum, and only using the green spectrum respectively. Due to the sea surface generally containing higher intensities in the green and blue spectra than the red, the choice of method can affect the result of the edge detection algorithm. The different methods are considered in the parameter tuning for the visual light camera dataset.

The thermal camera images contain only one intensity value per pixel, hence no grayscale conversion is required.

### 5.2.2 Noise Reduction

Noise can come from different sources: the image acquisition process (e.g. sensor noise), or from the scene background. At the sea, the water surface is a major source of noise, and can make it difficult to detect small objects. A common method for reducing noise is to apply a low-pass filter on the data. For images, a common low-pass filtering method is to convolve the image with a Gaussian kernel, which is a discrete approximation of a two-dimensional Gaussian function [17]. The two-dimensional Gaussian function is described in equation (5.1), and an example of a Gaussian kernel of size 3x3 is shown in equation (5.2).

$$g(x, y, \sigma) = \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5.1}$$

$$g = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{5.2}$$

The parameters used in Gaussian smoothing are $\sigma$, which determines the width and height of the peak in the Gaussian function, and the size of the Gaussian kernel, $s$. A larger $\sigma$ and size will smoothen the image more, resulting in less noise, but also less defined edges.

### 5.2.3 Edge Detection

In order to find the amplitude of the edges in an image, the gradient of the image is calculated. The method commonly used in e.g. the Canny edge detection algorithm is the Sobel operator [99]. Other methods exists, such as Robert's cross and the Prewitt operator, but they are generally outperformed by the Sobel operator [99]. Thus only the Sobel operator is considered in this chapter.

The Sobel operator consists of two kernels, one for the horizontal edges, $s_x$, and one for vertical edges, $s_y$. An $s_x$ of size 3x3 is shown in equation (5.3), and $s_y$ is the transpose of $s_x$, see equation (5.4). Larger Sobel kernels, commonly 5x5 and 7x7, can also be used.

$$s_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{5.3}$$

$$s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{5.4}$$

The kernels are convolved with the image in order to calculate an approximation of the image gradient in each direction, and finally the complete image gradient is calculated according to

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{5.5}$$

where $G_x(x, y)$ and $G_y(x, y)$ are the results of convolving $s_x$ and $s_y$ with the image, respectively.

The parameter used in edge detection is the size of the Sobel kernel, $s$. Three different sizes are considered: 3x3, 5x5, and 7x7.

### 5.2.4 Locating Objects in the Images

In order to find the location of the objects in the images, their positions were semi-automatically determined. The procedure varies between visual light images and thermal images due to the different noise characteristics – the visual light camera images are taken at an angle and at a high resolution resulting in a complex scene, making robust object detection difficult. In the method used, the location of the boundaries of the objects were manually selected by a human camera operator in every $n$th camera image, while the position of the boundaries in the intermediate images were found via linear interpolation. $n$ was chosen to be 10, which gave a good compromise between an accurate position estimate and a low manual workload.

The thermal camera images are taken close to vertically at a lower resolution resulting in a homogeneous background, which is a good precondition for robust automatic object detection. A Gaussian Mixture Model [134] (GMM) was first used to find the position of the objects in the images. The GMM segments the image into background and foreground, and is suitable for thermal images at the sea since there are two distributions present in the incoming radiance: the radiance reflected from the sky and the heat emitted from the water. The GMM algorithm used was implemented as part of the Background Subtraction Library [129]. The algorithm detected most objects in the thermal image dataset, but the detections were manually corrected by a human camera operator to improve the detection quality.

### 5.2.5 Distance and Angle

In order to evaluate the detectability of the objects in relation to the distance and angle between the camera and the objects, the distance and angle need to be calculated. The distance refers to the three-dimensional Euclidean distance between the camera and the object at a given time. The angle refers to the angle between the vector pointing from the camera towards the Earth and the vector pointing from the camera towards the object. See figure 5.1 for an illustration of the angle.

In order to calculate the distance and the angle, the world position of the camera and objects need to be determined. The world position of the camera and boat and pallet are obtained from their onboard GNSS. In order to find the world

position of the human and buoys, the attitude and position data of the camera are used together with the assumption that the object altitude is zero to project the object image position onto the world coordinate system using the pinhole camera model [108]. In order to improve the camera attitude estimate, the roll and pitch angles are corrected by solving Wahba's problem [93], where the position of the boat and pallet in the image coordinate system are aligned with their world positions projected onto the image coordinate system.

### 5.2.6 Geometric Limits

In order to find the geometric limit of whether an object is detectable or not, its apparent area (perpendicular to the camera sensor) is projected onto the camera sensor. The geometric limit for detecting an object is considered to be when an objects smallest apparent dimension (width or height) is represented by one pixel in the camera image. The following approximations are made regarding the dimensions of the objects:

- The boat has a height of 1.2 meters. Due to the complex geometry, this is a rough approximation.
- The human is 0.5 meters wide, and sticks out 0.25 meters from the water surface.
- The surface of the pallet is at the water surface.
- The buoys are floating on top of the water surface.

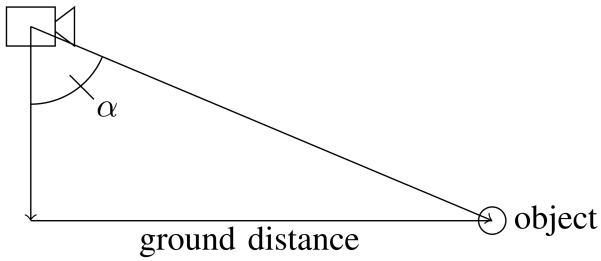


Figure 5.1: The angle between the camera and object, $\alpha$.

### 5.2.7 Error Metric and Evaluation

In order to quantitatively evaluate the detectability of objects, a metric needs to be chosen for how well the objects were detected in the camera images. The goal of the metric is to give a higher score when the difference between the object edges and background edges are larger.

The proposed method first generates a number of windows at random locations in the image. The windows are generated so to not contain any object, i.e. they represent the background. For the visual light camera images, the background windows also does not contain any part of the sky or land areas. An object is then considered detectable if its average edge value (the intensity of the image gradient) is larger than the average edge in a predefined fraction of the background windows.

### 5.2.8 Parameter Tuning

In order to obtain results which are not affected by poorly chosen parameters, the parameters of the edge detection algorithm are tuned using grid search. A random sample consisting of 20% of the images from the visual light camera and thermal camera datasets are selected to tune the parameters for each of the datasets. As the optimal parameters will differ between the visual light and thermal camera images due to their difference in resolution, scene, and noise, they are tuned independently.

A summary of the parameters and their ranges are shown in table 5.1 for the visual light camera, and table 5.2 for the thermal camera.

Table 5.1: Algorithm parameters for the visual light camera images.

| **Grayscale conversion** | |
|---|---|
| Type | average, only red, only green, only blue |
| **Blurring** | |
| Gaussian kernel size | 1x1 – 15x15 |
| Gaussian $\sigma$ | 0.5 – 7.5 |
| **Edge detection** | |
| Sobel kernel size | 3x3 – 7x7 |

Table 5.2: Algorithm parameters for the thermal camera images.

| **Blurring** | |
|---|---|
| Gaussian kernel size | 1x1 – 7x7 |
| Gaussian $\sigma$ | 0.5 – 7.5 |
| **Edge detection** | |
| Sobel kernel size | 3x3 – 7x7 |

Each combination of the parameters in table 5.1 and 5.2 are tested on the visual light and thermal camera datasets, and the parameters producing the highest detectability rate for the worst case object type (boat, pallet, human, buoy) are chosen as the best parameters, which will be used in the final evaluation of detectability.

## 5.3 Experimental Data

This section describes the datasets used in this chapter. The dataset consists of images captured by a visible light and a thermal camera, as well as the position and orientation of the cameras. The cameras were attached to UAVs and flown at varying positions relative to the objects. The subsections describe the objects and information about the visible light and thermal cameras.

### 5.3.1 Objects

The objects placed at the water surface was a 26 foot boat, an EUR-pallet, a human wearing an immersion suit, and two red buoys with a diameter of 60 cm, which were chosen as common objects to detect in maritime missions such as search and rescue and seismic operations. The objects are shown in figure 5.2.

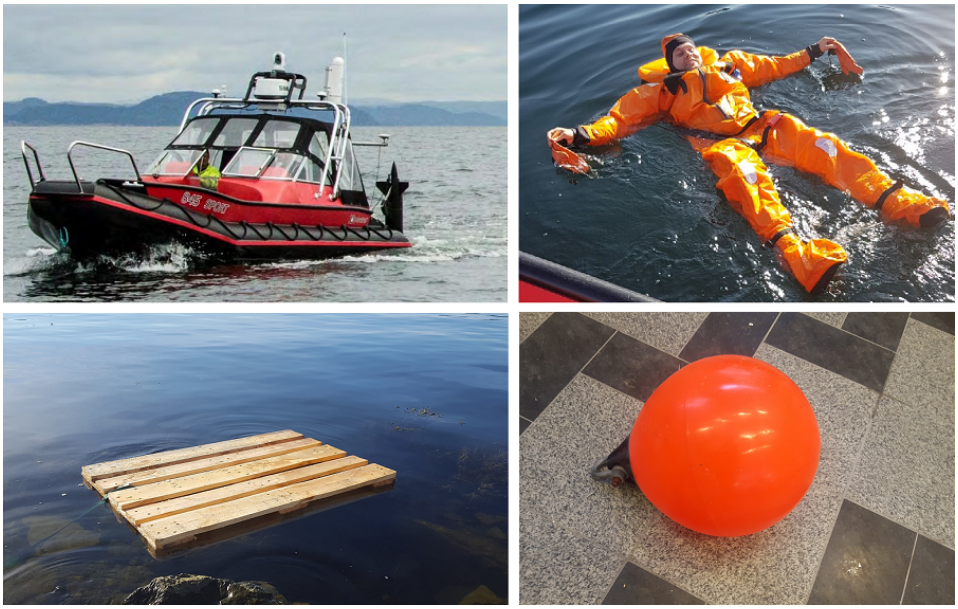

Figure 5.2: The objects present in the scene. Upper left: boat, upper right: human in immersion suit, lower left: EUR-pallet, lower right: buoy.

The boat and pallet were equipped with GNSS (Global Navigation Satellite Systems), which means that their world position data is available.

### 5.3.2 Visible Light Camera Data

The visible light camera is a DJI FC350, which provides images at a 3840 px x 2160 px resolution at 25 Hz in 24 bit H.264 format. The lens has a focal length of 3.61 mm, providing an angle of view of 83.6° x 50.2°. A subset of the whole dataset is used for the detectability evaluation – 2000 images for when the UAV is moving towards the objects, and the objects transitions from not detectable to detectable. See figure 5.3 for a sample image from the dataset. It can be seen that the lighting conditions varies throughout the image – the upper left part is brighter than the lower right part. Different lighting conditions can result in under- or overexposure of the objects, which might decrease the detectability.

Figure 5.3: A sample image from the experiment dataset captured by the visible light camera.

The position and orientation of the camera is available as metadata. This is used to calculate the distance and angle between the camera and object for each image, as well as determining the world position of some of the objects (see section 5.2.5). Figure 5.4 shows the distances and angles between the camera and the objects. The camera altitude is kept steady at 63–65 meters throughout the dataset.

### 5.3.3 Thermal Camera Data

The thermal camera is a FLIR Tau2, which provides images at a 640 x 512 pixels resolution at 9 Hz in 16 bit raw format. The lens has a focal length of 19 mm, providing an angle of view of 32° x 26°. See figure 5.5 for a sample image from

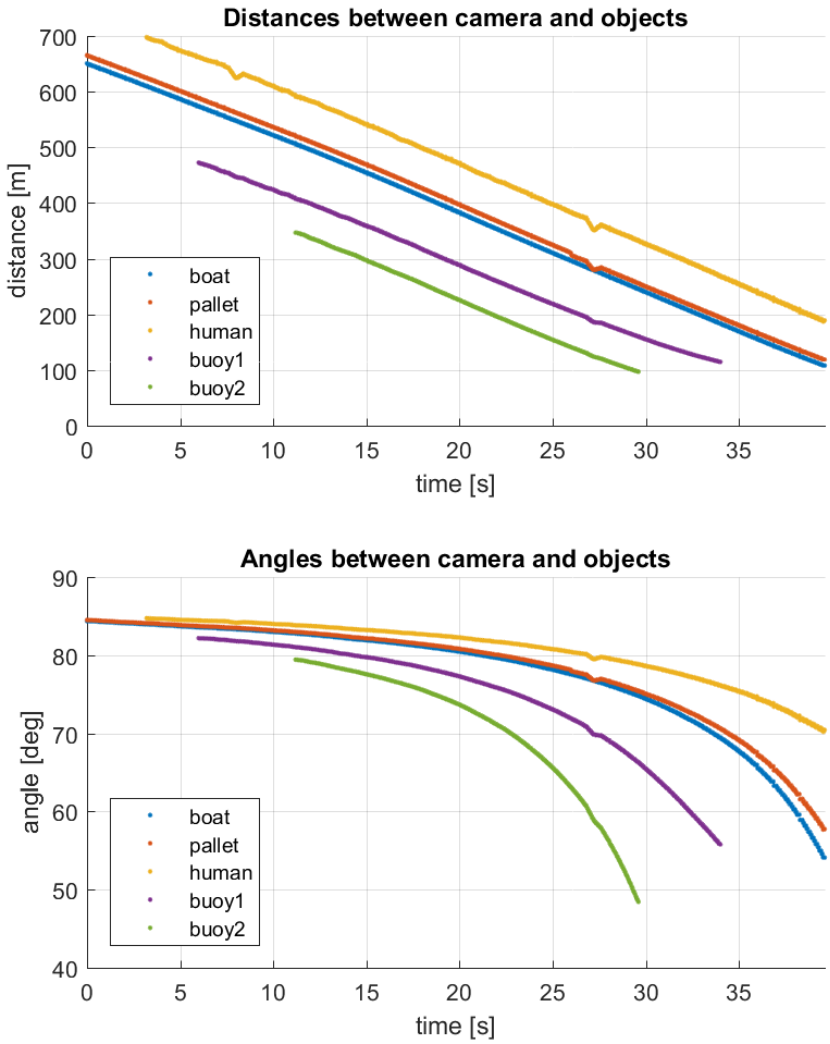


Figure 5.4: The distances and angles between the visual light camera and the objects at the sea surface.

the dataset. A total of 1974 thermal camera images, containing at least one object each, are used to evaluate the detectability of the objects.

Position and orientation data of the camera is also available as metadata. The thermal images are taken close to vertically, i.e. with the camera pointing straight down. Because of this, the total number of detections are evaluated rather than

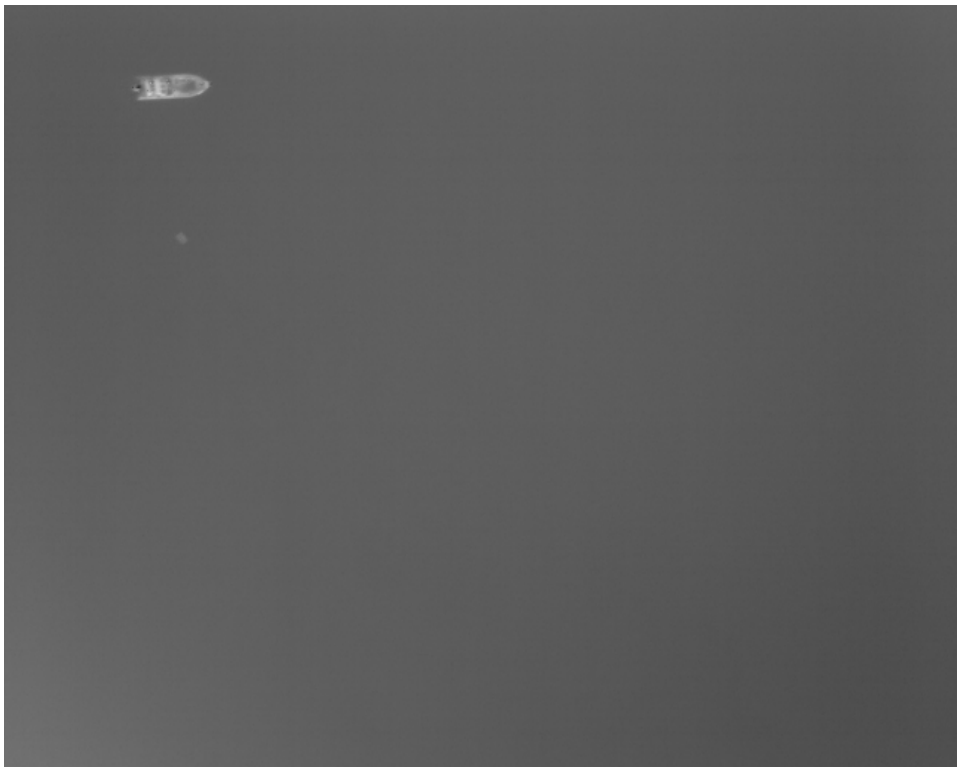

Figure 5.5: A sample image from the experiment dataset captured by the thermal camera. The boat is visible in the upper left corner, and a human or buoy is visible below the boat.

the distance and angle between the camera and objects. The altitude of the camera varies between 235 and 250 meters during the flight.

## 5.4 Experimental Results

The datasets were processed according to the methods described in section 5.2. At first the parameters were tuned for the visual light and thermal camera datasets, and the best performing parameters were evaluated. The best performing parameters were then applied on the full data sets, and the results were evaluated in relation to the camera to object distance and angle.

### 5.4.1 Parameter Tuning – Visual Light Camera

The parameter were tuned using 100 background windows with a width and height of 200 pixels for each image. An object is considered detectable when it has an

average intensity larger than 90% of the background windows.

The detection rate for each type of object for different parameters can be seen in figures 5.6–5.9.

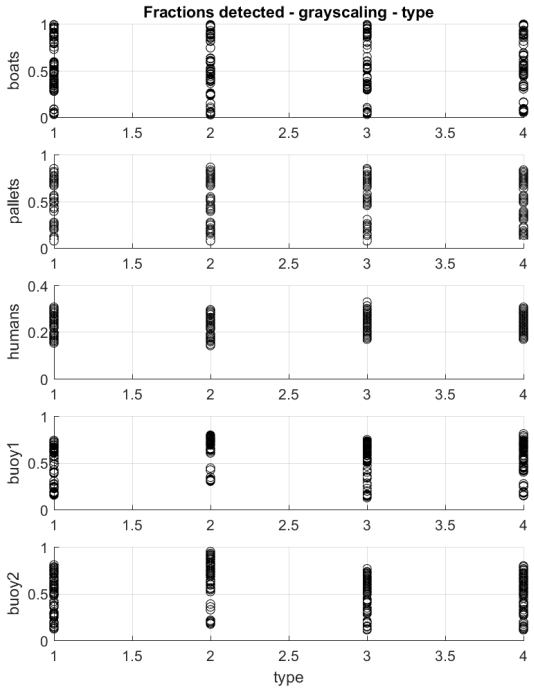


Figure 5.6: The fraction of objects detected for different methods for converting the image to grayscale in the visual light camera image test dataset. 1: RGB average, 2: only red, 3: only green, 4: only blue.

The grayscale conversion parameter influence on detectability can be seen in figure 5.6. The method of converting the RGB image to grayscale has a noticeable effect on the detectability of humans and buoys. For detecting the human, using only the green channel produces the best results, while this yields the worst detectability of the buoys. Only using the red channel produces the best detectability for buoys, while it produces among the worst detectability for the human. Since both objects are closest to red, while the sea surface is closest to blue, it would be expected that only using the red channel would produce the best results. The poor detectability for the human when only using the red spectrum might be due to image compression artifacts, rendering the colors of the human closer to its surrounding.

Since the human has the lowest detectability of all objects, this will be the deciding factor, thus only using the green channel is set as the best parameter for grayscale conversion.

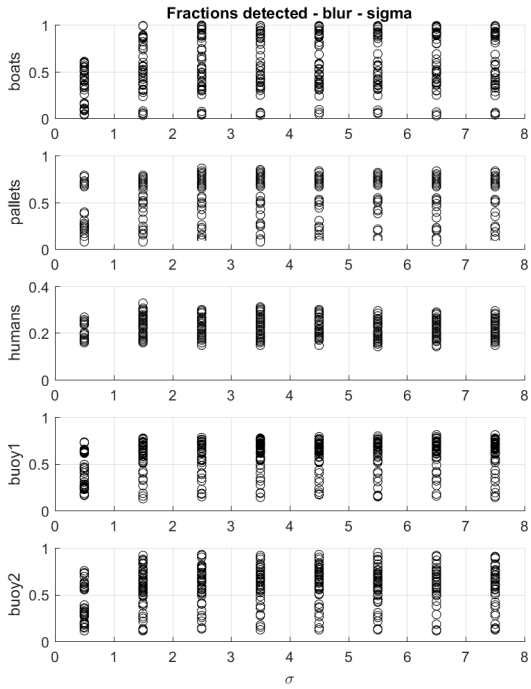


Figure 5.7: The fraction of objects detected for different noise reduction $\sigma$ in the visual light camera image test dataset.

The noise reduction parameter influence on detectability can be seen in figure 5.7 and 5.8 for $\sigma$ and size respectively. The noise reduction algorithm uses two parameters – $\sigma$ and kernel size. An increase of either parameter reduces the noise more, but produces less defined edges. It can be seen that the detectability is greatly improved for all objects when applying minor blurring. For the larger objects, in particular the boat, it keeps increasing, while for the smaller objects the detectability decreases after a Gaussian kernel size of 5x5 px, and $\sigma$ of 1.5 for the human and around 5.5 for the buoys.

Since the human has the lowest detectability of all objects, this will be the deciding factor, thus a $\sigma$ of 1.5 and kernel size of 5x5 is chosen as the best parameters for noise reduction.

The edge detection parameter influence on detectability can be seen in figure 5.9.

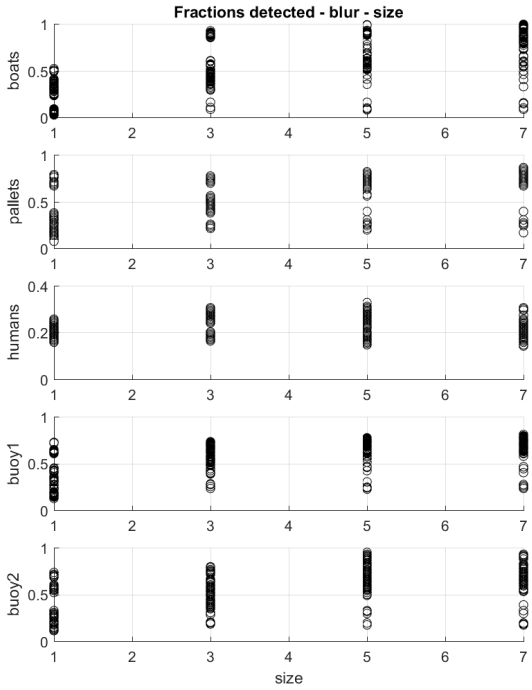


Figure 5.8: The fraction of objects detected for different noise reduction kernel sizes in the visual light camera image test dataset.

It can be seen that for all objects except the pallet, using a Sobel kernel size of 5x5 produces a better detectability. A kernel size of 7x7 seems to be able to increase the detectability of certain objects, but in general the 5x5 Sobel kernel produces the best results.

Since the human has the lowest detectability of all objects, this will be the deciding factor. The parameter set which produces the best results had a Sobel kernel size of 3x3, and is thus chosen as the best parameters for edge detection.

The parameter set which provided the best detectability for the visual light camera dataset is shown in table 5.3.

A sample image before and after applying the edge detection algorithm using the best and worst parameter set is shown in figure 5.10. It can be seen that an edge detector reducing noise rather than emphasizing the edges produces better results.

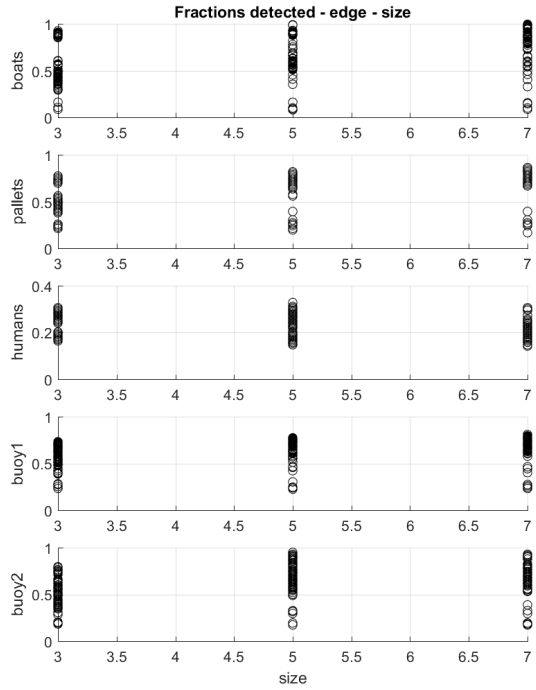


Figure 5.9: The fraction of objects detected for different edge detection kernel sizes in the visual light camera image test dataset.

### 5.4.2 Detectability – Visual Light Camera

The optimal parameters were used to run the algorithms on the full visual light camera dataset. The results are visualized in figures 5.11 – 5.15 as its detectability status (detectable by the algorithm, detectable by the operator, or not detectable) in graphs showing the distance between the camera and the objects, and the number of pixels in the smallest dimension when projected onto the camera image plane. The status ”detectable by the operator” means that the human camera operator was able to locate the object in the camera image (see section 5.2.4). The status ”not detectable” is thus only available for the boat and the pallet since their world position is available from their GNSS.

It can be seen that the boat is detectable by both the operator and edge detection algorithm throughout the image dataset. The algorithm misses the boat in two images, which can be considered outliers. This is as expected, as the number of pixels representing the smallest dimension of the boat in the image barely dips below 4.

Table 5.3: The parameters producing the best results in the thermal test dataset.

| **Grayscale conversion** | |
|---|---|
| Type | Only green |
| **Blurring** | |
| Gaussian kernel size | 5x5 |
| Gaussian $\sigma$ | 1.5 |
| **Edge detection** | |
| Sobel kernel size | 3x3 |

The pallet can be detected by the operator until its smallest dimension reaches around 1.5 pixels, while the algorithm can detect the pallet until it reaches 2.5 pixels.

The human remains very detectable by both the algorithm and operator until a camera to object distance of around 300 meters. At that distance, the human is represented by slightly above 1 pixel. The greater detectability of the human than the pallet can be explained by that the human is at a brighter part of the sea surface, creating a larger contrast between the object and the background and thus a larger edge intensity.

The first buoy is more detectable then the second buoy, both for the algorithm and the operator. Looking the the images, this is mainly caused by the lighting – the first buoy has less sun reflections, and thus has a deeper red color in the camera images. The more red color separates it better from the sunlight reflections in the sea surface. See figure 5.16. The buoys are sporadically not detectable by the algorithm, but when their size is close to 3 pixels, there are no missed detections.

### 5.4.3 Parameter Tuning – Thermal Camera

The parameter were tuned using 100 background windows with a width and height of 30 pixels for each image. An object is considered detectable when it has an average intensity larger than 100% of the background windows.

The detection rate for each type of object for different parameters can be seen in figure 5.17–5.19.

It can be seen that the Sobel kernel size has the largest effect on the detectability, with a larger kernel resulting in more objects detected. The effect on boats, which

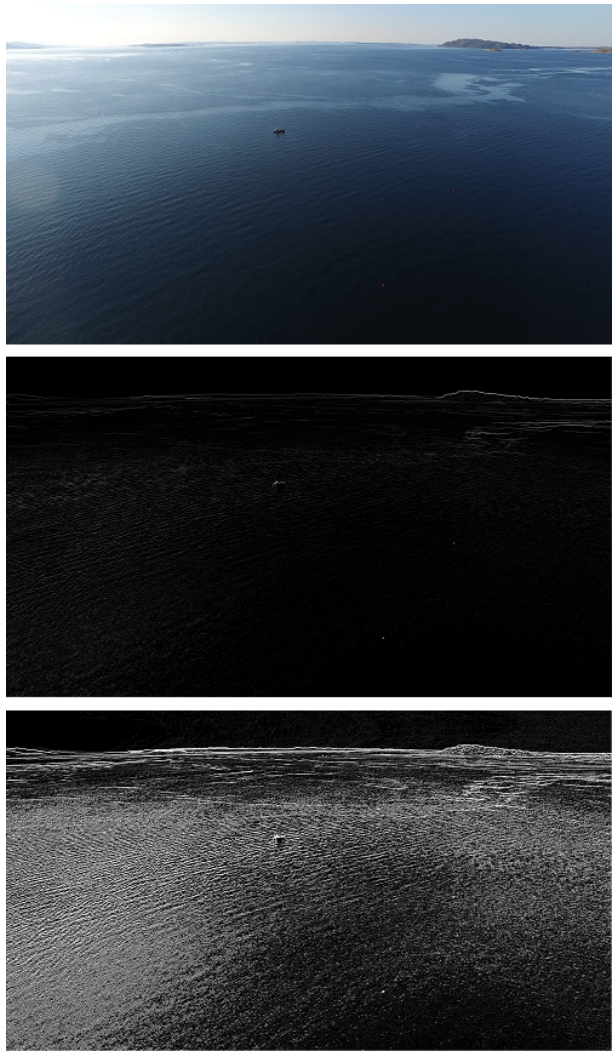

Figure 5.10: A visual light camera image (top), and its edges using the best parameters (middle) and worst parameters (bottom).

are larger, is minor, while the detectability of humans/buoys and pallets are greatly increased, and often reach 100% using a Sobel kernel size of 7x7.

The parameter set which provided the best detectability for the thermal camera dataset is shown in table 5.4.

A sample image before and after applying the edge detection algorithm using the best and worst parameter set is shown in figure 5.20. It can be seen that an edge detector emphasizing edge detection over noise reduction produces better results.

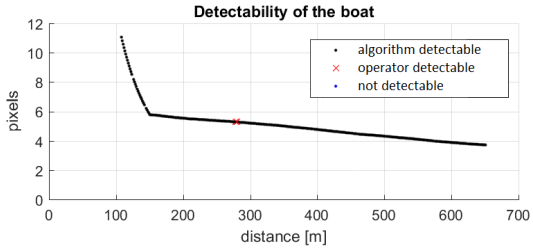


Figure 5.11: The detectability of the boat compared to the distance between the camera and the number of pixels representing the smallest dimension of the boat.

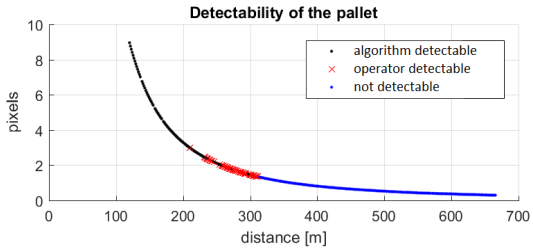


Figure 5.12: The detectability of the pallet compared to the distance between the camera and the number of pixels representing the smallest dimension of the pallet.

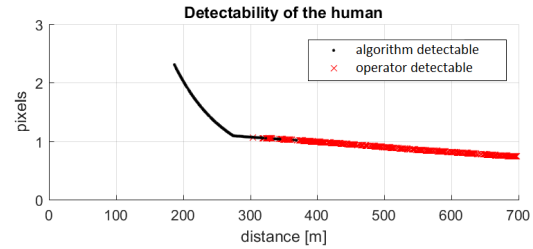


Figure 5.13: The detectability of the human compared to the distance between the camera and the number of pixels representing the smallest dimension of the human.

### 5.4.4 Detectability – Thermal Camera

Using the best parameter set from the parameter tuning, the algorithm is applied for the full thermal camera dataset. The number of detections are summarized in table5.5. In order to estimate how well the objects can be detected geometrically, the pixels representing the objects in the camera images are shown in table 5.6.

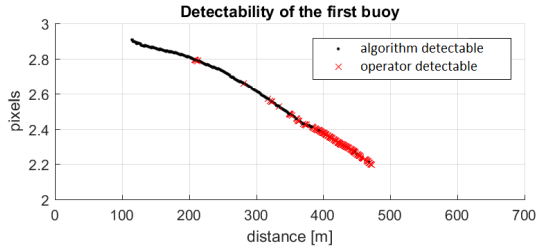


Figure 5.14: The detectability of the first buoy compared to the distance between the camera and the number of pixels representing the smallest dimension of the first buoy.

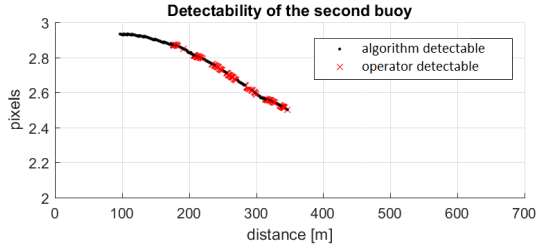


Figure 5.15: The detectability of the second buoy compared to the distance between the camera and the number of pixels representing the smallest dimension of the second buoy.

Table 5.4: The parameters producing the best results in the thermal test dataset.

| **Blurring** | |
|---|---|
| Gaussian kernel size | 1x1 |
| Gaussian $\sigma$ | 1 |

| **Edge detection** | |
|---|---|
| Sobel kernel size | 5x5 |

Table 5.6: The size of the objects in pixels, assuming a vertical photo at an altitude of 240 meters.

| **Object** | **Size [px]** |
|---|---|
| Boat | 38 x 12 |
| Pallet | 5.7 x 3.8 |
| Human | 2.4 x 1.4 – 2.4 x 8.1 |
| Buoy | 2.9 x 2.9 |

Figure 5.16: The first buoy (left) versus the second buoy (right), as they appear in a visual light camera image.

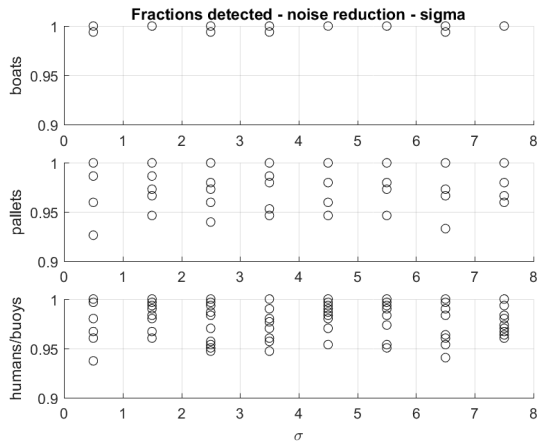


Figure 5.17: The fraction of objects detected for different noise reduction $\sigma$.

It can be seen that all boats and pallets in the dataset were detected. 6 humans/buoys were not detected, representing 0.4% of all objects of the same class. As can be seen in table 5.6, most objects are above 2 pixels in their smallest dimension, which is the general limit for detectability according to Johnson's criteria [119]. An exception is the human, which can fall below 2 pixels when standing upright (only the head and shoulders are above the water surface).

## 5.5 Conclusions

In this chapter we have evaluated the detectability of objects that commonly needs to be detected in maritime surveillance missions. The detectability metric was the

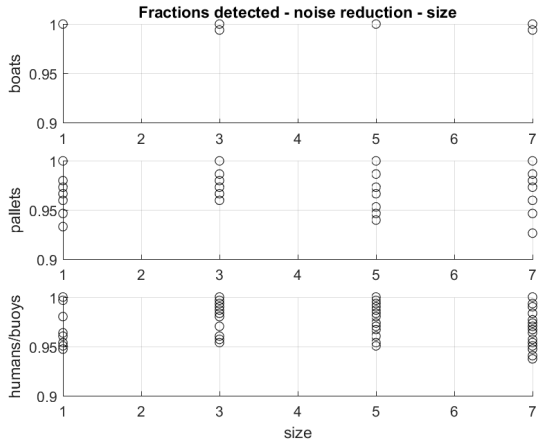


Figure 5.18: The fraction of objects detected for different noise reduction kernel sizes.

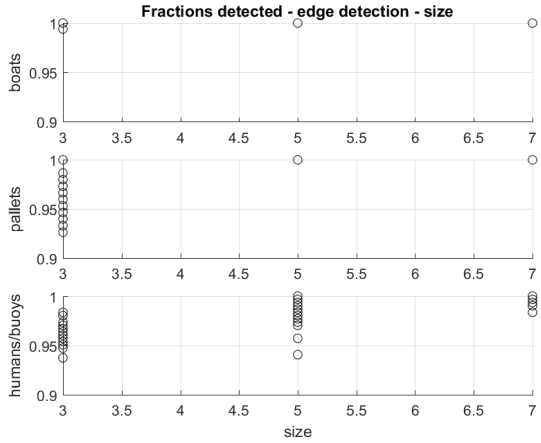


Figure 5.19: The fraction of objects detected for different Sobel kernel sizes.

average edge intensity of an object compared to 100 randomly located background windows in the same image, after applying grayscale conversion, noise reduction, and edge detection algorithms on the images. In order to avoid poorly chosen algorithm parameters, parameter tuning was performed using 20% of the full datasets. The detectability was then evaluated in relation to the number of pixels representing the smallest dimension of the objects when projected onto the camera image plane for the visual light camera images, and the total number of detections for the thermal camera images.

The results show that objects can be reliably detected when the smallest dimension

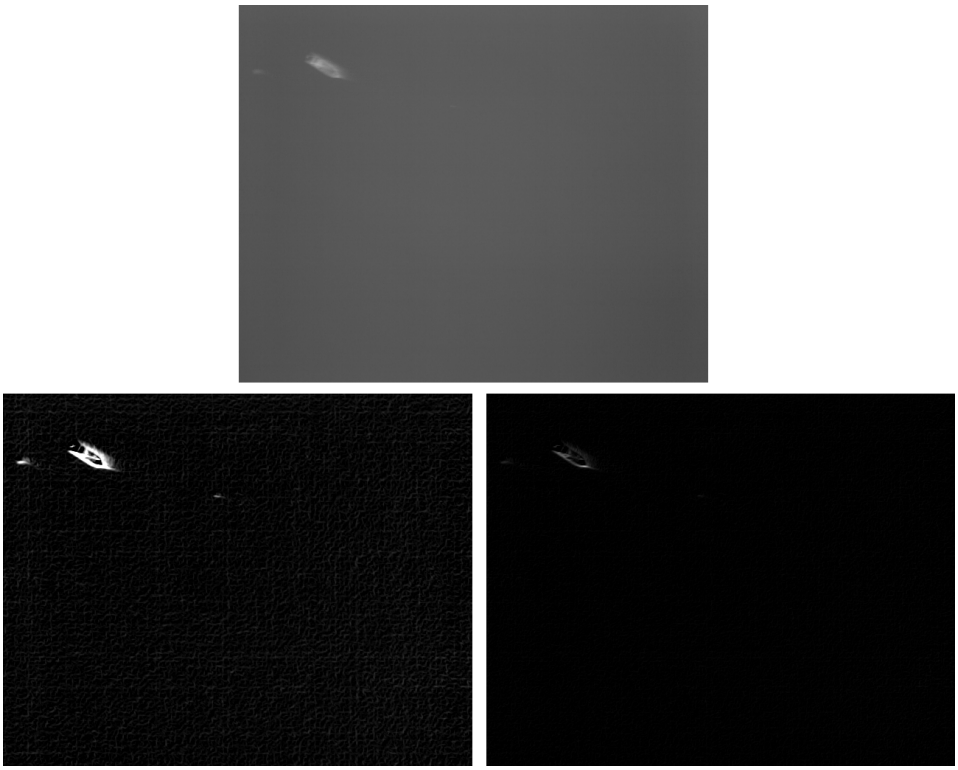

Figure 5.20: A thermal image (top), and its edges using the best parameters (bottom left) and worst parameters (bottom right).

Table 5.5: The number of objects considered detected in the thermal camera image dataset.

| **Object** | **Detected** | **Total** | **Fraction** |
|---|---|---|---|
| Boats | 869 | 869 | 100% |
| Pallets | 787 | 787 | 100% |
| Humans/buoys | 1504 | 1510 | 99.6% |

is around 3 pixels or larger. At lower pixel counts, the noise from the sea surface, e.g. waves, is too large compared to the contrast between the sea surface and the objects. The lighting is also shown to have a certain affect on the detectability, where two buoys of the same size and color have a different detectability due to their different lighting conditions.

Further work should be done to generalize detectability for different algorithms further, e.g. evaluating other noise reduction and edge detection algorithms, as well

as other object detection algorithms. Due to their good object detection and classification performance, the results of Convolutional Neural Networks could be used to determine detectability of the objects. In order to develop a more sophisticated model of object detectability at the ocean surface, more advanced camera sensor and optics models, as well as water surface models, could be used. E.g. could the modulation transfer function (MTF) be used to model how well the camera can detect the objects. The lighting conditions for the different objects should also be further investigated, since it was shown to be an important factor of detectability when the objects were represented by a very small number of pixels in the camera images.

# Chapter 6

# Object Classification in Thermal Images using Convolutional Neural Networks for Search and Rescue Missions with Unmanned Aerial Systems

In recent years, the use of Unmanned Aerial Systems (UAS) has become commonplace in a wide variety of tasks due to their relatively low cost and ease of operation. In this chapter, we explore the use of UAS in maritime Search And Rescue (SAR) missions by using experimental data to detect and classify objects at the sea surface. The objects are chosen as common objects present in maritime SAR missions: a boat, a pallet, a human, and a buoy. The data consists of thermal images and a Gaussian Mixture Model (GMM) is used to discriminate foreground objects from the background. Then, bounding boxes containing the object are defined and used to train a Convolutional Neural Network (CNN). The CNN achieves the average accuracy of 92.5% when evaluating a testing dataset.

## 6.1 Introduction

Maritime Search and Rescue (SAR) operations are usually based on the drifting trajectory, which is influenced by the water streams and winds. In such operations, it is common to estimate the drift by deploying buoys with GPS sensors to transmit their positions [66]. Since changes in the environment at the search region are common, the search parameters might change many times during the mission, leading to the necessity of the reconfiguration of the mission itself. The search is

usually performed using manned aircraft and vessels and is limited by the costs, the availability of human resources, and the mental and perception limitations of the human operators. All these limitations impose that a method for automatic classification of objects would be beneficial to the SAR mission as an additional assistance to the operators, due to its ability to process multiple inputs at higher speeds and with an invariable reliability rate, as it is not subject to exhaustion.

The use of Unmanned Aerial Systems (UAS) has grown rapidly, especially because of their high endurance, reduced cost, rapid deployment and flexibility. They also offers reduced risk for humans and impact on the environment compared to manned aircraft. Therefore, intelligent autonomous UAS equipped with image recognition capabilities to classify vessels, wrecks, people and objects pose are well suited tools to assist maritime SAR operations.

In these missions, it is fundamental to identifying key objects in aerial images using the techniques of object detection, classification and tracking. However, it might be more challenging to solve these classic computer vision problems when using UAS, especially because of real-time requirements and relatively fixed view angles. Moreover, running computationally intensive algorithms, such as image processing algorithms and deep neural networks with many filters and convolution layers, can be an additional challenge due to UAS power consumption limitations, and space and weight constraint for embedded hardware.

Leira et al. [79] used thermal camera images captured by UAS to detect, classify, and track objects at the sea. The solution presented arises as a useful tool for SAR operations. The object detection algorithm used relies on static filter parameters and thresholds, which are determined manually a posteriori. The classifier used is based on the object area, the average object thermal radiation, and its general shape. However, there are a number of scenarios where this classification would be challenging, e.g., when motion blur is present or when the object is moving across an image with varying sensor intensity, which can be caused by an uneven scene radiance or sensor noise. Therefore, a deep learning algorithm could be a more effective tool for the object classification, since it can handle variations on the images affected by environmental changes, as long as these effects are widely present in the dataset.

Convolutional Neural Networks (CNN) are the state-of-the-art deep learning tools for classification of images. Using convolution and pooling layers, it is possible to efficiently extract the most relevant features of the images. Some works were done with CNN and UAS, as in [125], where bounding boxes of images captured by a camera mounted on a UAS at a high altitude were classified in real-time into four classes: building, ground, tree and road. In [75], ground animals were detected

using CNN in aerial images captured by a camera mounted on a low-cost UAS in Namibia and the step of object detection for bounding boxes prediction was also explored in the work. Sea animals were detected in aerial images in [92], where the bounding boxes were defined by the confidence of each pixel of being the center of a window containing a mammal and then a CNN is used to classify the images. Regarding the use of CNN to classify objects in aerial images taken by a UAS in maritime environments, a work was done by [27], where RGB images were used and bounding boxes were classified into two classes: *boat* or *notboat*. SAR, CNN and UAS are used together in [8], where near real-time object detection was performed by a UAS for avalanche SAR missions. A pre-trained CNN did the object detection and a Support Vector Machine (SVM) was used to classify the proposed human bodies. All of these works were done using datasets of RGB images, but there are also some works using CNN with thermal images, as in [68], to monitor machine health and in [71], to detect pedestrians. However, there were not found any works using CNN to classify objects at the sea in aerial thermal imagery and this is particularly important in night time during low visibility SAR operations.

In this chapter, a CNN is trained to classify boats, buoys, people and pallets in images captured by a thermal camera mounted on a fixed-wing UAS. The foreground objects were detected by modeling the background as a mixture of Gaussian distributions and subtracting the foreground [134]. This method is computationally cheaper than other object proposal methods such as sliding windows [102] or selective search [139] because it is particularly suitable for thermal images at the sea, as there are two modes present in the distribution: the radiance reflected from the sky, and the heat emitted from the sea [15]. Subsequently a window was fitted around the objects and padded to ensure that the full objects were included in the window. One other novelty brought by this study is the use of the estimated observed area as an extra feature in the fully connected layer of the CNN.

## 6.2 Dataset

The dataset consists of images captured by a thermal camera mounted on a fixed-wing UAS. The thermal camera used is a FLIR Tau2, which provides analogue video data at a $640 \times 512$ pixels resolution. The lens has a focal length of 19 mm, which produces a $32^\circ \times 26^\circ$ angle of view. The analogue video data is converted to digital using a 16 bit analogue-to-digital converter. In order to create 8 bit images, the 16 bit images are normalized between 0 and 255 for the smallest and largest intensity in the full dataset. The UAS was also equipped with an Inertial Measurement Unit (IMU) and Global Navigation Satellite System (GNSS) unit, in order to find the surface area of the objects in the images (see section 6.2.3).

Four different objects were placed in the ocean: a 26 feet boat, a euro pallet, a human wearing an immersion suit, and a buoy with a 60 cm diameter. The objects were chosen as common objects present in maritime SAR missions, where e.g. pallets are a common object to search for when trying to locate fish aggregating devices. The objects can be seen in higher resolution visual light camera images in figure 6.1. The human varied between different actions during the experiment: floating horizontally on the surface (creating a large, long surface), swimming (creating a medium sized surface varying in shape), and standing vertically (creating a small surface, down to 20 cm across).

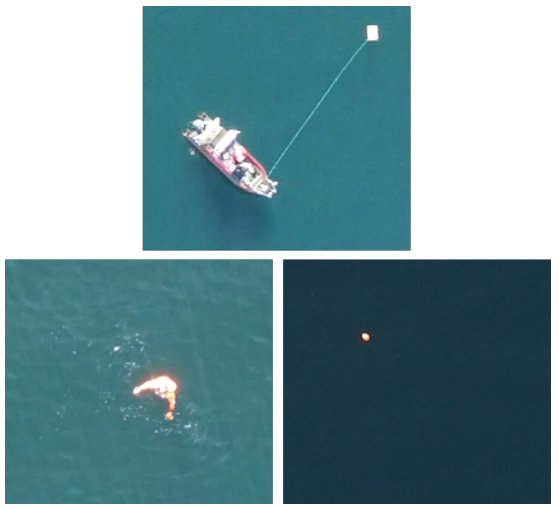

Figure 6.1: The different objects present in the scene, as captured by a higher resolution, visual light camera. Top: boat and pallet, bottom left: human, bottom right: buoy. The images were captured at different altitudes.

The total dataset consists of around 22,000 images that were captured during a time span of 50 minutes. The objects were only fully inside in the camera field of view in a limited subset of the full dataset, leading to a smaller number of images used in the CNN.

Various imperfections were present in the images. Several images contain motion blur caused by the dynamics of the UAS. This effect is minor for larger objects such as the boat, however for smaller objects such as a human head sticking out from the water, it can greatly affect the shape, size, and intensity of the object. See figure 6.2 for an example of how motion blur changes the object size and dimensions. The pixel intensity is also varying throughout each image, which makes the same object take on intensities between 97 to 110 in an example 8 bit image sequence. This might be caused by noise in the uncooled thermal image sensor, internal camera

intensity calibrations, or varying scene radiance. The background varies between 81 and 98 in the same image sequence. See figure 6.3 for an average of all images without objects, where the intensity variation can be seen.

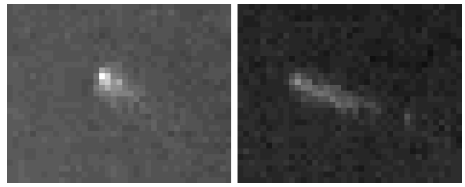

Figure 6.2: The same object (human) without motion blur (left) and with motion blur (right). The shape, size, and intensity is greatly affected.

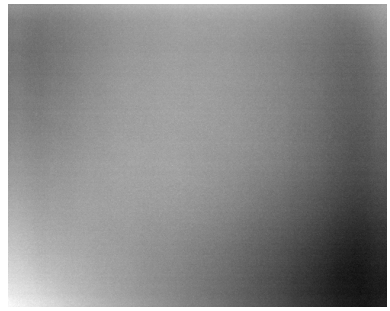

Figure 6.3: The mean image without any objects, with its intensity stretched to show the varying image intensity.

In order to find the objects in the images and label them, their boundaries were first found (section 6.2.1). The objects were then automatically labeled based on the physical area of the boundary (see section 6.2.1 for definition of the physical area) and finally manually corrected (section 6.2.2). The number of labeled objects in the dataset used in the CNN is summarized in table 6.1.

Table 6.1: Number of labeled objects in the dataset

| | |
|---|---|
| **Boats** | 620 |
| **Pallets** | 739 |
| **Humans** | 313 |
| **Buoys** | 276 |

### 6.2.1 Bounding Boxes

In order to discriminate the foreground objects from the background in the images, the background pixels were modeled using an adaptive background Gaussian

Mixture Model (GMM) [134]. The GMM provides robust foreground segmentation and is suitable for thermal images at the sea since there are two modes present in the distribution: the radiance reflected from the sky, and the heat emitted from the sea. It can also model the varying sensor noise, but might fail when the thermal camera is performing sudden noise corrections. The algorithm was implemented using the Background Subtraction Library [129]. A study by Borghgraef et al. [15] showed that more advanced algorithms, such as ViBe and the behaviour subtraction algorithm, outperformed the GMM for detecting objects at the sea surface in thermal images. However, this was for a static camera at a highly slant angle, which means that the study is not completely applicable to the scenario of this chapter. For this application, the GMM is chosen as a good balance between robustness and simplicity. The bounding box was then defined as the smallest box that encloses the boundary of the object. The bounding boxes of all objects were then padded to the size of the largest bounding box found in the dataset. See figure 6.4 for a sample boundary and bounding box.

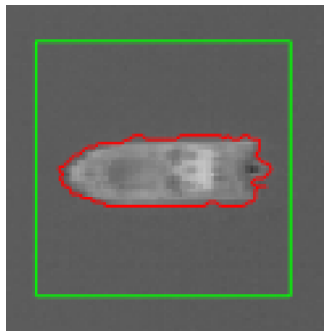

Figure 6.4: The border around the extracted foreground object (red), and the bounding box (green).

### 6.2.2 Labeling

In order to use the extracted foreground objects in the supervised learning algorithm, each object needs to be properly labeled. The objects were first assigned one of three labels based on their observed area in square meters (see section 6.2.3) - boat, pallet, or human/buoy. Each label was then manually verified and adjusted if deemed incorrect.

Due to the low ground resolution and their similar dimensions, discriminating humans from buoys was not possible only using the size as a criterion or by looking at individual images due to the varying shapes of the human and other effects, e.g., motion blur. A manual classification was therefore done by analyzing the shape of each object appearing in a sequence of images, taking into consideration that the

buoy is completely round while the human has a more elliptical and varying shape. See figures 6.5 and 6.6 for examples with a boat, a pallet, a human, and a buoy in the images.

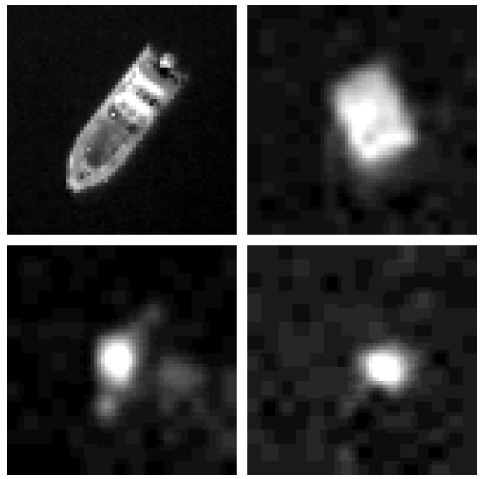

Figure 6.5: The different objects which were labeled. Top left: boat, top right: pallet, bottom left: human, bottom right: buoy. The images are scaled to show more detail.

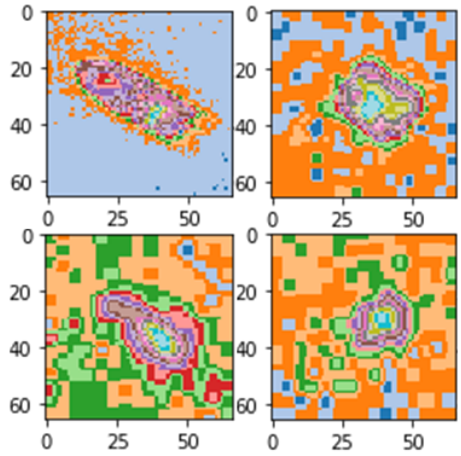


Figure 6.6: Objects in a different color map, in order to aid in manually discriminating humans from buoys. Top left: boat, top right: pallet, bottom left: human, bottom right: buoy.

### 6.2.3 Object Area

The observed area of each object in square meters is used as an extra feature in the fully connected layer of the CNN. The real observed area is defined as the area of the object as seen by the camera, when projected at the plane spanning the North and East axes (NE-plane) at an altitude of zero ($D = 0$). See figure 6.7 for a visual description of the observed area of an object.

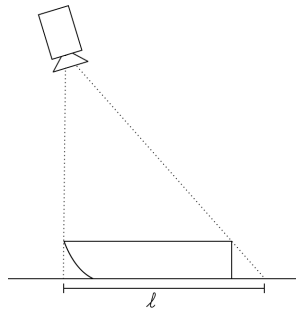


Figure 6.7: The observed length, $l$, of an object. The observed area is the corresponding feature in two dimensions.

The pinhole camera model [108] is used to calculate the observed area of the boundary of each object. First, the observed area of the center pixel within the boundary is calculated, which is then multiplied by the number of pixels within the boundary. In order to perform these calculations, it is necessary to know the roll and pitch angles and altitude of the camera. This data is obtained from the IMU and GNSS data, and is represented in the form of the extrinsic camera matrix. The intrinsic camera matrix is calculated from the camera specification. No lens distortion is considered – due to the relatively small angle of view of the lens, the distortion will likely be low and not affect the results in a significant way.

The observed area distributions for each object is shown in figure 6.8. It can be seen that boats and pallets can be almost completely classified based on their observed area (with minor overlap between pallets and humans), while there are major overlaps between humans and buoys. This is however an artifact of this dataset – in other datasets, buoys and boats can take on a variety of sizes. As previously mentioned, humans can take on a wide variety of sizes due to the different poses.

The real observed area of a buoy with a diameter of 60 cm should be 0.28 $m^2$. As can be seen in figure 6.8, the area is biased towards higher values. One reason for this is that a 60 cm circle can appear in 16 pixels (figure 6.9), when the observed area of each pixel is 17.9 cm – which is the case when flying at an altitude of 200 m with no roll or pitch using the camera system used in the experiment performed.

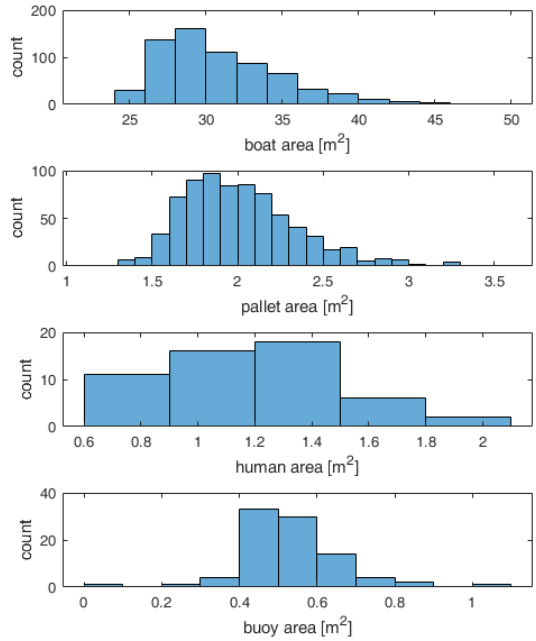


Figure 6.8: Distribution of observed areas for the objects present. From top to bottom: boats, pallets, humans, buoys.

This gives an observed area of 0.52 $m^2$. Additionally, the motion blur sometimes cause the object to appear larger than it really is.

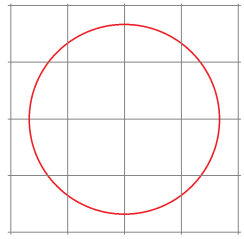

Figure 6.9: A buoy with a diameter of 60 cm can appear in 16 pixels, when the observed width and height of each pixel is 17.9 cm.

## 6.3 Convolutional Neural Network

Traditionally, supervised learning based image analysis combines feature extraction with classical machine learning methods [141]. Convolutional Neural Network (CNN) is an alternative trend for image classification that has been proven to

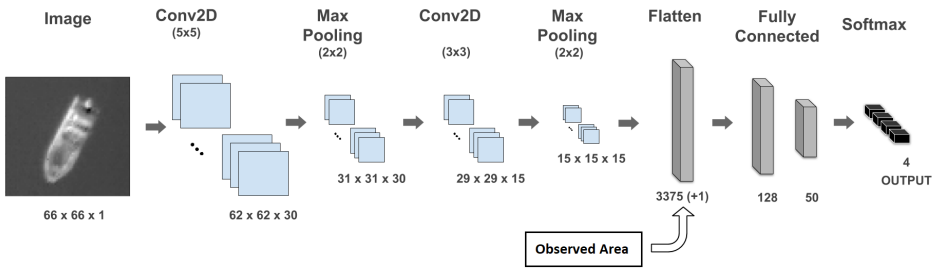


Figure 6.10: CNN architecture. If the Estimated Object Area is used, one more element is added in the flatten layer, resulting in an output of 3.376 elements.

produce high accuracy in image classification tasks [146] without requiring any task-specific feature engineering [97]. It is considered the most successful machine learning model in recent years [78] and the most eminent method in computer vision [26], in part because it consists of a powerful image feature extractor [10].

A CNN is based on neuroscience research about the processes that mammalian visual cortex uses to recognize images [46]. Typically several basic stages compose a CNN. Each stage consists of concatenation of convolution, normalization, activation (nonlinear), and pooling layers [145]. In this work, two distinct architectures were used. The difference between them was the employment of the observed object area as an input of the fully connected layer.

### 6.3.1 Architecture

Regarding the architecture (see Figure 6.10), the proposed network starts with an input layer containing the image window. This layer is followed by the convolution layer which produces 30 feature maps from filters of size 5×5. The convolutional layer has a set of learnable filters called kernels. By the convolution between one kernel and a chunk of values from the layer, a feature map is generated, which consists of a presence representation of a specific feature in the image. The next layer is a max pooling with a filter of size 2×2, whose purpose is extracting the hierarchical features of the input image [88]. It works by mapping the bigger value from a 2×2 chunk to only one value in the next layer. Pooling helps to make the representation approximately invariant to small translations of the input [46]. This function is also responsible for reducing the width and the height of the feature map. Reducing this dimension, the computational demand is reduced due to the reduction of the number of parameters, which helps to avoid over-fitting. Then, it comes another convolution layer with 15 feature maps of size 3×3, and finally one more max pooling layer composed by 2×2 filters.

The next layer is a flatten layer used to adjust the tensor dimensions to the fully connected layers. At this point, the two architectures become different. One network has the estimated object size as an input and the other one does not. Then, it follows 2 fully connected layers composed respectively by 128 and 50 neurons using the rectifier activation function. The last fully connected layer is used to provide the predicted classification, using the softmax activation function. This is the most common solution for the regulation of the output values within the range from 0 to 1 [87], which assigns a multinomial probability distribution to the output vector [153]. It enhances the discriminative modeling power of the CNN, providing the probability of the input to belong to each possible class, namely: boat, buoys, human or pallet.

### 6.3.2 Dropout

One technique widely used to improve the performance and avoid the over-fitting (which is often a serious problem for a CNN [149]) is the dropout. The term "dropout" refers to dropping out units (hidden and visible) in a neural network during the training phase. By dropping a unit out, it means temporarily removing it from the network in the current epoch, along with all its incoming and outgoing connections [130].

The dropout parameter controlled in this chapter was the independent probability of deactivating a neuron. This parameter was tested with 2 values: 0.2 and 0.5.

### 6.3.3 Cross-validation

In order to evaluate the generalization capacity of the classifiers, it is preferable that the dataset used in the evaluation process is different from the one used during the training process. Typically, the formation of the training and test set is based on non-repetitive sampling techniques, such as the $k$-fold cross validation method [118]. Cross-validation is a robust statistical technique for estimating the true risk function [5] (or the generalization error), the most important operational performance metric of a trained network [74].

In this chapter, the database is divided into five sets of equal size. During each execution of the algorithm, one set is chosen to be excluded in the training phase, which will be the corresponding test set. This process is repeated five times, and the performance metric is inferred for each of the datasets that were left out of the training process. The value of the overall performance metric will be defined by the average of the values obtained for each of the five executions.

### 6.3.4 Stopping criterion

In the neural network training phase, a stopping criterion has to be used to stop the training of the neural network. To ensure that the training was stopped in a way to provide an appropriate generalization, a validation based early stopping is used in this work [107]. A small part of the training dataset is sorted out to be used as a validation set. At each epoch, the performance index is evaluated for the new training set and for the validation set. When the performance metric of the validation set stops to decrease, i.e., when the training starts to over-fit, the training is stopped.

## 6.4 Results

After testing the convergence for different parameters, the first CNN was chosen and trained for the 5-folds of images in 8 bit format, without taking into consideration the estimated size of the objects. The maximum number of epochs was 500 and the early stopping was set to stop the training after 50 validation evaluations without improvements. The validation split was 0.18 and dropout was 0.50. After doing 10 executions to get an indicative statistical performance, the average accuracy was 92.0% with 0.50% of standard deviation. This result shows that the configuration of the training algorithm was well set, so that the performances of all executions for all folds were similar.

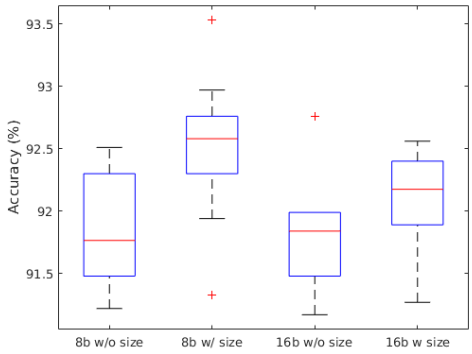


Figure 6.11: Accuracy for different configurations.

When using the estimated objects size as an extra input of the fully connected layer, the resulted accuracy is higher as shown in figure 6.11 for 8 bit and 16 bit images, achieving 92.5% and 92.1% of accuracy, respectively. Regarding the classification, it is possible to notice that classifying between buoys and humans is a challenge as seen in the Confusion Matrix (table 6.2). However, the use of the estimated

object size helps the CNN to get better results (table 6.3). When looking to the Confusion Matrix for buoys, there are fewer cases when the buoy is classified as a human. There is even a case of a boat being classified as a pallet when the estimated object size was not used (table 6.2).

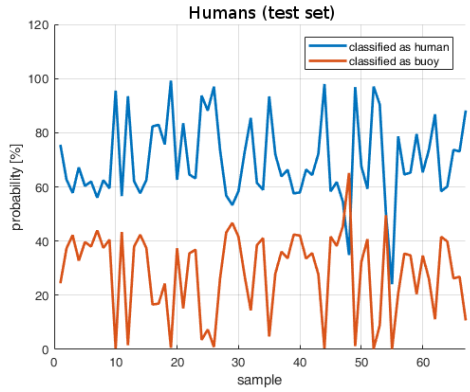


Figure 6.12: Probability that a human is either a human (blue) or buoy (red).

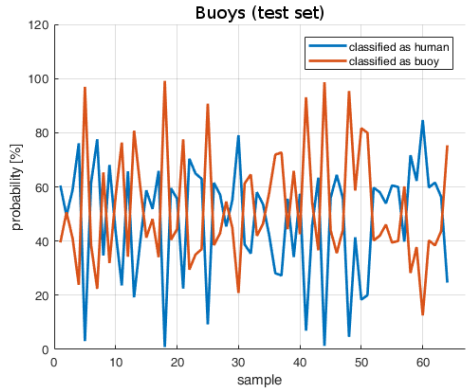


Figure 6.13: Probability that a buoy is either a human (blue) or buoy (red).

The ability of the CNN to classify humans vs. buoys is further investigated in figures 6.12 and 6.13, where the probability of each human and buoy test sample being either a human or a buoy are shown. In the humans samples, it is possible to notice that it is easier for the CNN to differentiate them from buoys. However, when analyzing the classification probabilities of the buoys samples, it shows that it is very challenging to the CNN to decide if it is a buoy or a human.

Regarding the comparison of the performance between 8 and 16 bit images, even with the fact that the 16 bit images have more detail, the accuracy was sightly higher for the configuration with 8 bit images. This might be caused by the reduc-

tion of noise when reducing the bit depth, e.g., small intensity variations in the sea surface.

Table 6.2: Confusion Matrix for 8 bit images without using object size.

| | | Predicted | | | |
|---|---|---|---|---|---|
| | | Boat | Human | Buoy | Pallet |
| True | Boat | 128 | 0 | 0 | 1 |
| | Human | 0 | 36 | 18 | 0 |
| | Buoy | 0 | 15 | 45 | 1 |
| | Pallet | 0 | 0 | 1 | 145 |

Table 6.3: Confusion Matrix for 8 bit images using object size.

| | | Predicted | | | |
|---|---|---|---|---|---|
| | | Boat | Human | Buoy | Pallet |
| True | Boat | 114 | 0 | 0 | 0 |
| | Human | 0 | 41 | 13 | 0 |
| | Buoy | 0 | 7 | 64 | 0 |
| | Pallet | 0 | 0 | 1 | 150 |

Images of three vessels from a different dataset (Figure 6.14) were used to evaluate the performance of the CNNs.

The observed areas of the vessels were estimated and chosen to be 200 $m^2$, 25 $m^2$ and 10 $m^2$ for the big, medium and small boat, respectively.

When using the CNN where the observed area is not used, all the three samples achieved 100% of probability of being a boat. Regarding the CNN where the observed area is used as an input, the big and medium vessel achieved 100% of probability of being a boat and the small vessel was classified as a pallet.

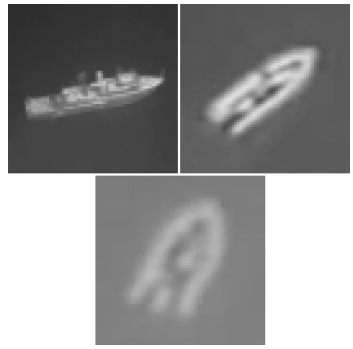

Figure 6.14: Images of vessels from an external dataset. Top left: big vessel, top right: medium vessel, bottom: small vessel.

## 6.5 Discussion

To obtain the images of the objects, the bounding boxes were defined as the smallest box that encloses the boundary of the object. Then, the bounding boxes of all objects were padded to the size of the largest bounding box found for the set of objects of the same class. However, in the SAR mission, it is not possible to know the class of the object a priori, therefore, another strategy need to be used to define the bounding boxes. One solution could be to define a specific number of pixels pad the boundaries of the detected object with, to ensure that the whole object will be inside the bounding box. This number of pixels should be defined by the altitude of the UAS when the image is being captured and also the estimated size of the object. Thus, effects by the distance to the scene would also be mitigated.

The observed area, as well as its appearance in the thermal images, are greatly affected by motion blur. For larger objects this does not pose a major problem, but for objects being just a few pixels in size, the difference can be of major concern. An actively stabilized gimbal and carefully chosen exposure times based on the UAS dynamics could prevent this. Another mitigating solution would be to collect a larger dataset in order to be able to properly classify objects even with motion blur.

The major difficulty of the CNN is to properly distinguish between humans and buoys, which is likely due to the low resolution of the thermal image sensor and relative high altitude, resulting in the objects being represented by very few pixels in the images. In real world maritime SAR missions, however, a buoy being classified as a human would not be a major issue, as the operator would still be notified, and could dismiss the notification from the CNN. Incorrectly classifying a human as a buoy could potentially cause a missed person, but could be solved by lowering the human probability threshold for notifying the operator.

In the dataset used in this work, all boat samples have similar observed areas. Therefore, when evaluating the classification performance for images of vessels from an external dataset, the result was superior when using the CNN where the object area was not considered as an input. However, the generalization power of the CNN containing the observed area can be improved by using a dataset with more samples of boats of different sizes. Also, in general, it is beneficial to have more data, especially at different angles and altitudes.

## 6.6 Future Work

In the mission carried out to gather the data used for this work, an Electro-Optical (EO) camera was also equipped in the UAS to capture RGB images. However, the

thermal and the RGB images were not obtained during the same flight, so it is not possible to use the two images together as inputs of the same CNN.

Therefore, a future step is to develop a CNN to classify the objects in the RGB images, as done by [27]. Then, investigating a method to use both datasets together, for example, trying to use the results of each independent CNN multiplying the probability of each sample to be one of the classes.

For the CNN proposed by this work, the classification of images of vessels obtained in another mission in totally different conditions was evaluated. However, it is important to evaluate the classification for images of humans, buoys and pallets as well. Thus, it would be possible to estimate how well the CNN could perform in a real mission.

Another aspect that requires evaluation is how to improve the discrimination between humans and buoys, especially in the case of buoys, where the calculated probability of a buoy sample being a buoy is very close to the probability of being a human. Examples of this approach would be to use an actively stabilized and sweeping gimbal together with a lens with higher focal length, in order to get a higher ground resolution.

## 6.7 Conclusion

In this chapter, the algorithm for detecting and classifying objects at the sea surface in thermal camera images taken by Unmanned Aerial Systems (UAS) has been discussed. The algorithm uses a Gaussian Mixture Model (GMM) in order to discriminate foreground objects from the background in the images. Then, bounding boxes around the objects are defined and used to train and test a Convolutional Neural Network (CNN). The observed area of the objects was also estimated and used as an input. The CNN was evaluated using the k-fold method with 5 folds and achieved an average of 92.5% of accuracy. Images of vessels from an external dataset were also evaluated and all of them achieved 100% of probability of being a boat when using the CNN where the observed area was not used. The results and the robustness of the CNN algorithm prove it to be a useful tool to assist maritime SAR operations, and be a central part in a future fully autonomous UAS operation in SAR missions.

# Chapter 7

# Concluding Remarks and Recommendation for Future Work

In this thesis, the accuracy of optical imaging systems mounted on sUAS have been analyzed, with a special focus on sUAS operations in maritime environments.

The thesis research body starts off by introducing problems that arises with a volatile and vibrating camera platform (i.e. a small UAS) in chapter 2. The main sources are vibration are presented, along with suggestions on how to solve them by choosing the right camera and optics, or software that is either dependent or independent of additional sensors. This research forms the foundation for selecting a suitable camera, as the camera requirements and limitations (such as weight and exposure time) are partly set by the capabilities, vibrations and movements of the sUAS platform.

Chapter 3 presents an algorithm for improving the accuracy in camera image georeferencing applications. The algorithm works by estimating the attitude of the camera by matching a horizon detected in the camera image with a horizon synthetically generated from a Digital Surface Model. It achieves record accuracy in the roll, pitch, and yaw angles, while running in near real-time. The chapter also serves as a statement of the geometrical performance and capabilities of camera systems, which aids in the camera component selection and performance estimation.

Chapter 4 uses in part the algorithm presented in the previous chapter in order to estimate the size and shape of ice-floes in a mission taking place in the Bothnian bay in Northern Europe. The chapter presents a method for estimating the size of individual ice-floes using alpha shapes, as well as a metric to evaluate the accuracy of the size estimation. The method showed that the size and shape of ice-floes can

be accurately estimated even at very steep angles, which shows the high spatial accuracy of camera systems.

In Chapter 5, data from an experimental mission is used to evaluate the detectability of objects commonly present in maritime surveillance missions. The sensors used for the evaluation was a thermal infrared camera and a visible-light camera, where the detectability was based on edge detection performance. The chapter concludes that the objects can be detected fairly robustly when the smallest dimension of the object allocates at least three pixels in the camera images. When the objects are smaller in the image plane, the scene noise makes object detection difficult. This shows the importance of not only understanding the camera system performance and noise, but also the scene properties such as noise.

Chapter 6 extends the previous chapter by attempting to classify the objects detected in the thermal infrared camera images. The method uses a Gaussian Mixture Model to first segment the image into foreground and background, and then a Convolutional Neural Network to classify the objects found. The algorithm achieved an accuracy of 92.5%. The objects that proved difficult to classify were humans and buoys, which in many cases just occupied a few pixels in the image plane, and would also be difficult for a human to classify without any additional information. This problem of low resolution could be solved by flying at a lower altitude, by using a higher resolution camera, or by using a larger focal length.

The main activities of recommended future work are grouped as the following.

- **Camera system simulations**: In order to easier predict how a camera system will perform, the camera system could be simulated beforehand. Models such as Modulation Transfer Function [14] and image sensor noise models described in e.g. [40] can be used to give a good indication of the camera system performance. An important factor in the simulation would be to find a suitable scene simulation engine. For this, an evaluation of game engines and databases of high quality (low noise and high resolution) camera images could be done to find the most accurate and flexible scene simulation method. Other factors such as blur caused by camera motion could also be included in such simulation in order to find a suitable camera stabilization platform.
- **Attitude estimation**: The attitude estimation algorithm presented in this thesis is shown to be highly accurate, and using it gives enough accuracy to e.g. estimate the shape of ice-floes at very slant camera-to-object angles. However, it lacks robustness, and low visibility or a difficult to segment camera image can render it useless part of the time. In order to circumvent this limitation, a complimentary method could be used to estimate the attitude

when the camera-based attitude estimate is not available. This could be done by e.g. using a Bayesian filter such as the Kalman filter or a particle filter together with data from an IMU. Such system could also calibrate the IMU, and provide a better starting point for the camera-based attitude estimate. Another improvement would be to use a more robust horizon detection method. This could be more recent improvements in image processing or using machine learning methods.

- **Multi-camera object classification**: In Chapter 6, research was done on the classification of objects commonly present in maritime search-and-rescue missions. It proved that object classification is fairly dependent on the objects having a high enough resolution. High resolution thermal infrared cameras can be very costly, so future work could be to solve this problem. One solution could be to reduce the altitude of the UAV when an object has been detected. This would allow the thermal infrared camera to capture higher resolution images of the object that has been detected, but may not be possible due to air space restrictions. Another solution could be to use two cameras: one thermal infrared, which has the benefit of easily detecting objects at the sea surface; and one visible-light camera which commonly has a much higher resolution.

# Appendices

# Appendix A

# Autonomous Unmanned Aerial Vehicles in Search and Rescue missions using real-time cooperative Model Predictive Control

Unmanned Aerial Vehicles (UAVs) have recently been used on a wide variety of applications due to their versatility, reduced cost, rapid deployment, among other advantages. Search and Rescue (SAR) is one of the most prominent areas for the employment of UAVs in place of a manned mission, specially because of its limitations on the costs, human resources, and mental and perception of the human operators. In this chapter, a real time path-planning solution using multiple cooperative UAVs for SAR missions is proposed. The technique of Particle Swarm Optimization is used to solve a Model Predictive Control problem that aims to perform search in a given area of interest, following the directive of international standards of SAR. The solution is able to be embedded in the UAVs on-board computer, using the Ardupilot autopilot and DUNE, an on-board navigation software. The performance is evaluated in a software-in-the-loop environment with the JSBSim flight dynamics model. Results show that when employing three UAVs, the group reaches 50% of Probability of Success 2.25 times faster than when a single UAV is employed.

## A.1 Introduction

Over the last years, Unmanned Aerial Vehicles (UAVs) are becoming an important and indispensable tool in a wide variety of applications due to their versatility, low cost, rapid deployment, among other advantages.

Search and Rescue (SAR) is one of the fields where the employment of UAVs brings many advantages over manned missions, such as its reduced costs, lower use of human resources, and mental and perception limitations of human operators. [122] was one of the first works to perform experimental tests of a complete autonomous single UAV SAR solution. A probability density function (PDF) that expressed the likelihood of the target's location was one of the main inputs of the system. Video data from the UAV was transmitted to the ground station, that processed it in real time using computer vision techniques to detect the presence of the target and update the PDF. Paths were generated by the ground station to maximize the probability of finding the targeted object. The experimental flights showed satisfactory results in searching and detecting the target. The main necessary improvements identified by the authors were to implement on-board computing and to use multiple UAVs in the future.

Search and Rescue missions with autonomous UAVs are usually defined as an exploration problem. Exploration approaches can be used in a wide range of applications. For example, ice management, such as proposed by [55], where a Centralized Model Predictive Search Software was used for surveillance and tracking of ice using multiple UAVs. In this reference, the optimization finds a set of optimal waypoints that are sent to the autopilot. The solution was tested in a Software-In-The-Loop environment and the results were evaluated for a different number of UAVs. A broad literature review about the persistent surveillance problem was done by [98] focusing on the use of multiple UAVs. Persistent surveillance is a type of exploration problem where the areas must be revisited over time. Among the many topics that the literature review covers, grid decomposition and path-planning techniques are the ones of the most interest for this work. The author reviews the most common types of grid decomposition classifying the rectangular one, that is also used by the enhanced solution proposed by this chapter, as the most popular. Regarding the path-planning techniques, the author states that the most common methods are classical search methods such as A* [67], decision theoretic methods such as Mixed Integer Linear Programming (MILP) [42], and Spanning Tree Coverage (STC) methods [43]. Model Predictive Control (MPC) is mentioned as a topic less studied compared to the other planning techniques, but with significant advantages because it directly incorporate dynamic constrains, it is less heuristic and can react to changes in the environment.

Model Predictive Control [22] is a receding horizon control technique where the motion constraints are integrated to the control problem, which is particularly interesting for problems with fixed-wing UAVs. Also, as the optimization is done for a finite time horizon, the technique is proper for real-time problems where the environment can dynamically change during the mission execution. In [128], MPC

was used for sea Search and Track (SAT) missions using an autonomous UAV. Hardware-In-The-Loop tests were performed. Waypoints were optimized and sent to the autopilot. Gimbal attitude was also optimized and sent to the servo system. The MPC optimization was not run on-board but on a dedicated computer in the ground control station. In [29], a cooperative multiple UAV solution using MPC was used to close the communication link between a moving Autonomous Surface Vehicle and the ground station. Each UAV had to minimize a local cost function that took into consideration the planned states of the adjacent UAVs. In [32], a multiple UAVs receding horizon strategy was proposed for a cooperative surveillance problem. A potential field method was used for collision avoidance and network topology control management. The cooperative searching model was established based on the detection probability of the UAVs on targets in cells. In addition, a forgetting factor was included to indicate how fast the detection efforts are forgotten, so the UAVs can revisit the areas that were searched before. Simulations for different parameters were compared. Also, the performance of the proposed method was compared to the performance of a parallel sequence search. In [138], a multi-vehicle cooperative search solution was proposed using MPC. Decoupled, centralized, cooperative and greedy approaches were compared.

In this chapter, a multiple UAVs cooperative Nonlinear Model Predictive Control solution to search a given area is proposed. The coordinated turn vehicle model is implemented considering the effects of wind. The search area is divided into cells and each cell has an associated reward, that in this work is defined according to the international Search and Rescue directives. The algorithm is fully implemented in an embedded system to run in the UAV on-board computer and interfaced to the flight controller board. A Software-In-The-Loop (SITL) environment with flight dynamics simulations is used to test the solution.

As mentioned, research about the use of receding horizon techniques for exploration is limited. Also, the solutions found in the literature are developed with simplified vehicle models, in which the effects of wind are not considered either. In addition, the solutions are only simulated in environments without embedded programming restrictions and where vehicle dynamics are not simulated. This makes results not close enough to what is expected from real-life applications. Implementing the solution in an embedded software for real-time applications brings additional challenges such as communication delays, processing time, actuator limitations, among others. This chapter contributes to the field by filling this gap in the literature.

The main contents of this chapter are the following. First, the exploration Model Predictive Control problem is detailed including a coordinated turn kinematic model that takes the wind into consideration. Second, a finite time horizon grid

search cost function with cells rewards and terminal cost is proposed. Third, the algorithm is fully implemented in an embedded software and tested in a real-time SITL environment that also simulates the flight dynamics, bringing results that are very close to reality. Forth, international SAR directives are used to define the performance indicators and the mission scenario in order to test the solution in a relevant case.

## A.2 Optimal Control Problem

### A.2.1 Coordinated Turn Model

A two-dimensional kinematic model is used based on the Coordinated Turn model [7] [120] [109] [9] [29]. In this model, the UAV turns by changing its roll angle and therefore there is no net side force acting on the UAV.

As wind is a major issue on UAV missions as it can likely reach more than half of the UAV's maximum airspeed, the Coordinated Turn model used in this project was developed to consider the influence of wind on the UAV kinematics.

For level flight in the presence of wind:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\chi} \end{pmatrix} = f(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} v_g \cos\chi \\ v_g \sin\chi \\ \frac{g}{v_g} \tan u_\phi cos(\chi - \psi) \end{pmatrix}, \tag{A.1}$$

where $g$ is the gravitational acceleration of 9.81 m/s$^2$, $\mathbf{x} = (x, y, \chi)$ are the north and east positions in the NED frame in [m] and the course angle in [rad], respectively. $\mathbf{u} = (u_v, u_\phi)$ are the airspeed control input in [m/s] and roll control input in [rad], respectively, and with the ground speed ($v_g$ in [m/s]):

$$v_g = \sqrt{(u_v \cos\psi + v_w \cos\psi_w)^2 + (u_v \sin\psi + v_w \sin\psi_w)^2}, \tag{A.2}$$

where $v_w$ is the wind speed in [m/s], $\psi_w$ is the wind heading in [rad] and with the aircraft heading ($\psi$ in [rad]) calculated using the law of sines:

$$\psi = \chi - \arcsin\Big(\frac{v_w}{u_v} \sin(\psi_w - \chi)\Big). \tag{A.3}$$

The model is discretized by the forward Euler method:

$$\mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k + T_s f(\mathbf{x}_k, \mathbf{u}_k), \tag{A.4}$$

where $T_s$ is the sampling period.

### A.2.2 Model Predictive Control Problem

To reach the mission goal, a centralized optimization approach might not be feasible because the problem would be too complex with too many control inputs. In a non-convex problem with a very long vector of variables to optimize, falling very early in a local minima is a common issue. In addition, the necessary processing power to optimize so many control inputs would be difficult to achieve by the on-board processing unit of the UAV. In the other hand, optimizing the controls of all UAVs in a ground station would not be an ideal solution, due to communication range limitations and because that in case of a communication failure, the UAVs would not receive its controls, which could compromise the mission.

Therefore, in this research, the problem is addressed as a cooperative control problem, where each UAV optimizes its own control inputs to update its state so that a local cost function is minimized. The cost function also takes into consideration the planned states of the other UAVs. As each UAV follows the same process, it is expected that the global mission goal is achieved cooperatively. Collision avoidance between UAVs is also considered.

Considering $I$ UAVs ($\mathbf{x}^i, \forall i \in \{0, ..., I-1\}$), the algorithm finds a control input sequence $U_k^i = \{\mathbf{u}_0^i, \mathbf{u}_1^i, ..., \mathbf{u}_{K-1}^i\} \in R^{2\times K}$ for the $i$th UAV, which solves the following optimal control problem:

$$\begin{aligned}
\text{minimize} \quad & \delta(\mathbf{C}\bar{\mathbf{x}}_K) + \sum_{k=0}^{K-1} L^i(\mathbf{C}\bar{\mathbf{x}}_k, \mathbf{u}_k^i) && \text{(A.5)}\\
\text{subject to} \quad & \mathbf{x}_{k+1}^i = f_d(\mathbf{x}_k^i, \mathbf{u}_k^i), && \text{(A.6)}\\
& v_{a_{min}} \leq u_{v_k}^i \leq v_{a_{max}}, && \text{(A.7)}\\
& \phi_{min} \leq u_{\phi_k}^i \leq \phi_{max}, && \text{(A.8)}\\
& |\mathbf{C}(\mathbf{x}_k^i - \mathbf{x}_k^j)| > r_c, \forall j \in \{0, ..., I-1\}\backslash\{i\}, && \text{(A.9)}
\end{aligned}$$

where

$$\delta(\bar{\mathbf{x}}_K) = F(\mathbf{C}\mathbf{x}_K) - aJ(\mathbf{C}\bar{\mathbf{x}}_K), \tag{A.10}$$

and

$$L^i(\mathbf{C}\bar{\mathbf{x}}_k, \mathbf{u}_k^i) = aJ(\mathbf{C}\bar{\mathbf{x}}_k) + b(u_{v_k}^i - u_{v_{k-1}}^i)^2 + c(u_{\phi_k}^i - u_{\phi_{k-1}}^i)^2. \tag{A.11}$$

Consider $u_{v_{-1}}$ and $u_{\phi_{-1}}$ as the commanded airspeed and roll angle, respectively, in the previous optimization loop, $\bar{\mathbf{x}}_k = [\mathbf{x}_k^0, ..., \mathbf{x}_k^{I-1}]$ as the states of all UAVs, $K$ as the number of horizon steps and $r_c$ as the minimum safe distance between the UAVs to avoid collision. $a$, $b$, $c$ are constant weighting factors and $\mathbf{C} \in R^{2\times 3}$ is used to define that only the $x$ (north) and $y$ (east) positions are used from the state vector:

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{A.12}$$

The function $J$ represents the grid search function, which is the sum of the rewards of unvisited cells. $F$ is the terminal cost (cost-to-go) function, which is the distance from the terminal position to the unvisited cell with highest reward. Both functions are described in detail in Section A.4.

## A.3 SAR directives applied to UAVs equipped with remote sensing

The Search and Rescue (SAR) consists, according to the Department of Defense (DoD) of the United States of America, in "the use of aircraft, surface craft, submarines, and specialized rescue teams and equipment to search for and rescue distressed persons on land or at sea in a permissive environment" [101]. This work focuses on the sea cases, therefore, the following description emphasizes sea SAR missions. Also, as only Unmanned Aerial Vehicles (UAVs) are used in this work, only the directives for aircraft facilities are studied.

### A.3.1 Search Area

According to the International Aeronautical and Maritime Search and Rescue (IAMSAR) Manual [65], the Total Adjusted Search Area ($A_t$), which is the mission's actual search area, is calculated based on the Total Available Search Effort ($Z_{ta}$), the Optimal Search Area ($A_o$) and the targeted Probability of Detection (POD). The first is a measure of the total area that a set of search facilities can effectively search within limits of search speed, endurance, and sweep width. The second is the search area which will produce the highest probability of success

when searched uniformly with the search effort available and is essentially calculated based on the leeway and the Datum probable position error. Leeway is the the movement of a search object through water caused by winds blowing against exposed surfaces and Datum is a geographic point, line, or area used as a reference in search planning, such as the "Last Known Position" or the "Estimated Incident Position".

If the Total Available Search Effort ($Z_{ta}$) is smaller than the Optimal Search Area ($A_o$), a strategy must be chosen to balance the Probability of Detection (POD) and the Total Adjusted Search Area ($A_t$). Usually, the chosen strategy is to fly on higher altitudes, increasing the sensor's footprint or the crew's field of view while decreasing the POD. However, in this work, as UAVs equipped with automated remote sensing are assumed to be used, resolution requirements usually can not be relaxed. Therefore, no trade-off between the POD and the search area is made and the POD is set to its maximum value of one, which makes the Total Adjusted Search Area ($A_t$) equal to the Total Available Search Effort ($Z_{ta}$).

In order to calculate the Total Available Search Effort ($Z_{ta}$), the sweep width ($W$) must be defined. When employing UAVs equipped with automated remote sensing in such missions, the sensor being used has a direct influence on this parameter. Altitude, view angle and image quality may affect the capability of identifying a survivor or an object on the sea. This is specially important to be taken into account because if the image does not contain the object of interest properly recorded, the computer vision algorithm will not identify it, independently of the ability that the algorithm has on identifying an important occurrence on an image. This can occur due to low image quality or too long sensing distances, making the object of interest imperceptible.

In the IAMSAR manual, the sweep width is calculated based on the altitude of the aircraft, the visibility and the sensor system specifications. In SAR sea missions with aircraft facilities and visual search, the Corrected Sweep Width ($W$) is adjusted regarding the weather, velocity and crew fatigue correction factors. However, these factors can be excluded from the equation when automated remote sensing systems are used and the system's velocity constraints are respected. Therefore, in this work, the Corrected Sweep Width ($W$) is considered equal to the original Uncorrected Sweep Width ($W_u$).

Finally, the Search Effort ($Z_a$), which represents the area which can be covered by a specific facility, is calculated by:

$$Z_a = V \times T \times W, \tag{A.13}$$

where $V$ is the Search Facility Speed (average speed) in [m/s], $T$ is the Search Endurance in [s] and $W$ is the Sweep Width in [m].

Note is that the Search Endurance is the time available for the facility to fly looking for the survivors. The IAMSAR manual considers this time as 85% of the lower value between the Daylight Hours Remaining and the On-Scene Endurance. This is due to the fact that human crew is often only able to search with visible light. Despite of it, UAVs are often capable to equip sensors that are not affected by that, such as infrared cameras, which allows the task to be held even along the night. This is a considerable advantage of using UAVs equipped with remote sensing systems.

By summing the Search Effort of all facilities, the Total Available Effort ($Z_{ta}$ in [$m^2$]) can be found:

$$Z_{ta} = \sum_{f=1}^{F} Z_{af}, \tag{A.14}$$

where $F$ is the number of facilities.

As described above, in this work the Total Adjusted Search Area ($A_t$) is equal to the Total Available Effort ($Z_{ta}$). Therefore, for Single Point Datum, the Length and the Width of the search area are given by the square root of the Total Available Effort ($Z_{ta}$) as defined by the IAMSAR manual.

### A.3.2 Probability Map

The Probability of containment (POC) distribution in the search area is very important to guarantee an efficient employment of the SAR facilities. When the initial indications do not provide enough information about the area, a standard distribution is assumed. The two most used types of standard distributions are the standard normal distribution and the uniform distribution, according the nature of the datum. For datum point and lines, the standard normal distribution is used. For datum areas, the uniform distribution is the most used. In this work, only the single point datum is studied. Single point datum occurs, for example, when there is no significant leeway (e.g. when the target is a person in water [19]).

The probability map is a set of grid cells where each cell is labelled with the probability of containing (POC) the search object in that cell. As the the probability map follows a probability distribution function, the total sum of all cells should be equal to 100%. An example of probability table for single point datum with $12 \times 12$ cells is shown in Figure A.1:


Figure A.1: Initial probability table. Source: IAMSAR Manual

## A.4 Cost Function

An exploration cost function was developed based on [138] to search a given area.

The region of interest is divided into $M \times N$ square cells of a width ($r_e$ in [m]), which value must be chosen to be smaller than the optical imaging sensor's footprint radius ($R_e$ in [m]) times the square root of 2. The sensor radius is equal to the radius of the circle inscribed in the sensor's footprint. Figure A.2 shows an example of a $4 \times 4$ grid with 100 m of cell width ($r_e$) and a UAV at position $\mathbf{Cx}$ equipped with a sensor with 100 m of radius ($R_e$).

The matrix $\mathbf{B}^i \in R^{M \times N}$ is used to identify if a cell was visited by the $i$th UAV. The matrix $\mathbf{b}^i \in R^{M \times N}$ is used to identify if a cell is planned to be visited by the $i$th UAV in the MPC horizon. In every $M \times N$ matrix used to identify if the cells are visited, each element has an associated value of 1 if the referring cell is visited or 0 if it is unvisited. Each cell has also an associated reward, given by $\phi \in R^{M \times N}$.

The function $J(\bar{\mathbf{x}}_k)$ is the sum of all cells associated value (1 or 0) in the step $k$ times the correspondent reward:

$$J(\bar{\mathbf{x}}_k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \phi_{mn} y_{mn_k}(\bar{\mathbf{x}}_k), \tag{A.15}$$

with

$$\begin{aligned} y_{mn_k}(\bar{\mathbf{x}}_k) =& (\|\mathbf{Cx}_k - r^1_{mn}\| < R_e \wedge \|\mathbf{Cx}_k - r^2_{mn}\| < R_e \wedge \\ & \|\mathbf{Cx}_k - r^3_{mn}\| < R_e \wedge \|\mathbf{Cx}_k - r^4_{mn}\| < R_e) \vee y_{mn_{k-1}}, \end{aligned} \tag{A.16}$$

where $r^1_{mn}$, $r^2_{mn}$, $r^3_{mn}$ and $r^4_{mn}$ are the four vertices of the cell (Figure A.2) and $y_{mn_{k-1}}$ is the associated value of the cell in the previous horizon step.

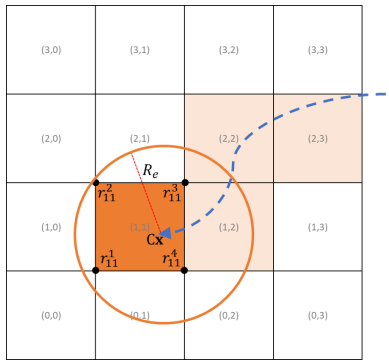


Figure A.2: Cells grid example.

The starting value of $y_{mn_0}$ is given by the logical sum of the matrices of already visited cells of all UAVs and the matrices of cells planned to be visited by other UAVs:

$$y_{mn_0} = \mathbf{B}^i_{mn} \vee \mathbf{B}^j_{mn} \vee \mathbf{b}^j_{mn}, \forall j \in \{0, ..., I-1\} \backslash \{i\}. \tag{A.17}$$

Finally, $F(\mathbf{x}_K)$ is the terminal cost. This function is necessary for the algorithm to consider the search beyond the prediction horizon by having a *cost-to-go* term. It is given as the minimum euclidean distance from the latest state of the UAV in the horizon, to the center of the closest unvisited cell, weighted by the correspondent reward, in the end of the horizon:

$$F(\mathbf{Cx}_K) = \min_{\forall m \in O, \forall n \in P} \frac{\|\mathbf{Cx}_K - \mathbf{r}_{mn_K}\|}{\phi_{mn}}, \tag{A.18}$$

where $O \subseteq M$ and $P \subseteq N$ are subsets of all unvisited cells and $\mathbf{r} = [x, y]$ are the north and east positions of the cell's center.

## A.5 Embedded System

The path-planning algorithm was implemented as a task in DUNE: Unified Navigation Environment [104]. DUNE is a software framework that allows the operation of a wide variety of robots using the same environment. This facilitates the development because the communication between DUNE and the different control units is transparent to the user.

Regarding the communication with the UAV control unit, it connects DUNE and the Ardupilot [2] autopilot via MAVLink Micro Air Vehicle Protocol [3]. To command the Ardupilot, the task must dispatch a message with the desired command, which will be interpreted by DUNE's Ardupilot control task and then sent to the Ardupilot via MAVLink.

The communication between DUNE tasks is done via the IMC: Intermodule Communication API protocol [104], which is also part of LSTS' toolchain. This protocol basically works by dispatching and consuming messages. So, if a message is dispatched by a task, another task that is waiting for that message will consume it.

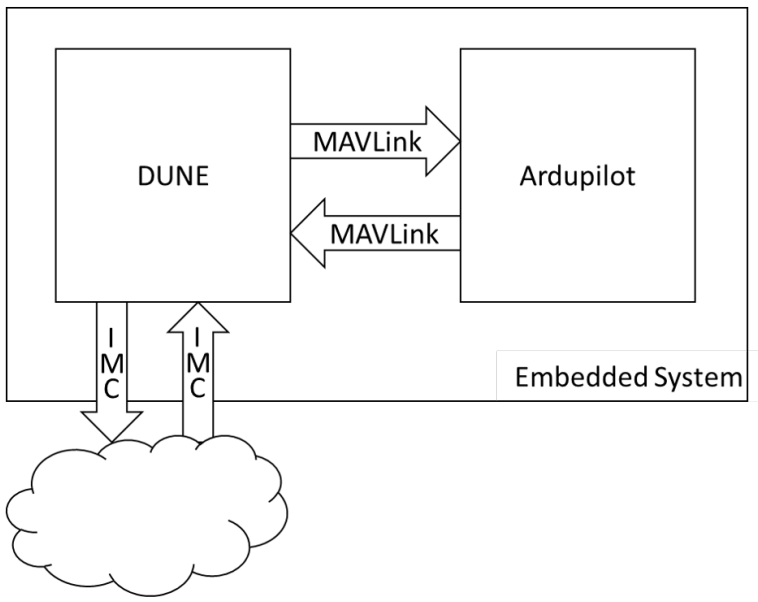


Figure A.3: Simplified embedded system block diagram.

Figure A.3 shows a simplified block diagram describing how the embedded system blocks are associated. DUNE communicates with the Ardupilot via MAVLink and with other UAVs and ground station via IMC. The systems must be in the same network and IP addresses and TCP and UDP ports must be configured.

Inside DUNE resides the MPC task, which is outlined at Figure A.4. The commands to control the UAV are given by the DesiredSpeed and DesiredRoll IMC messages, which carry, respectively, the airspeed and roll control inputs given by

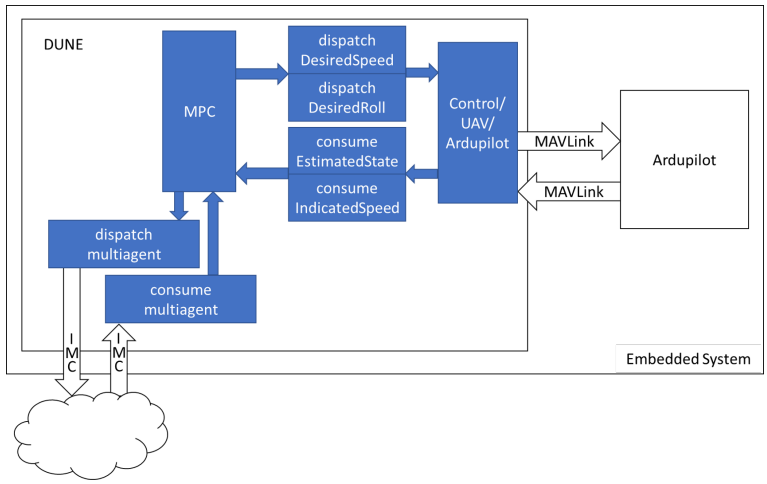


Figure A.4: Simplified DUNE block diagram.

the optimization. These messages are interpreted by the Ardupilot control task, that sends the correspondent MAVLink message to the Ardupilot. The Ardupilot control task is also responsible for receiving the pose and attitude information of the UAV and to dispatch it in the IMC messages EstimatedState and IndicatedSpeed. These messages are consumed by the MPC task to be used as the current state of the UAV.

The communication between the UAVs is done by the multiagent message, which was created and included in the IMC messages list specifically for this application. This message carries the information that the UAVs need to share, such as planned control inputs and current state.

Each UAV waits for the multiagent messages from all other UAVs before running the MPC optimization. Once all messages are received, the optimization is done and then the UAV dispatches its multiagent message containing all information that need to be shared with the other UAVs. This flow is described in Figure A.5. It is important to mention that the UAV states are predicted according to the communication and MPC delay and to the planned control inputs before being used by the MPC algorithm. Therefore, the control inputs that are obtained by the MPC optimization and sent to the Ardupilot correspond to the predicted state that is expected to be close to the real state of the UAV after the MPC optimization.

Note that, in an extension of the method, measures can be employed to protect the system from communication failures, so that the UAV does not wait for delayed messages for too long.

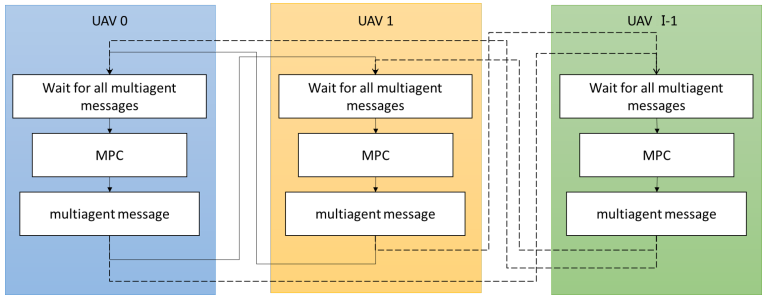


Figure A.5: UAV agents flowchart.

### A.5.1 Optimization Technique

In this application, the Particle Swarm Optimization (PSO) [35] technique is used to find an optimal set of airspeed ($u_v$) and roll angle ($u_\phi$) that minimizes the cost function. PSO, which is described in the Appendix, is a meta-heuristic optimization method where the particles (solutions) are updated every iteration based on the best global and local solutions. In this application, a standard PSO algorithm was implemented using CUDA C programming language in order to benefit from the parallelism of the Nvidia Graphics Processing Unit that is assumed to be used in the UAV on-board computer.

The algorithm was set to run a fixed number of iterations on every loop. In addition, the number of particles must be defined. These two parameters affect the processing time and need to be fine-tuned according to the requirements.

The initial solutions are initiated with random values following the uniform distribution, where the minimum and maximum values are the defined boundaries of the airspeed and roll angle control inputs.

## A.6 Software-In-The-Loop environment

To evaluate the proposed solution, a Software-In-The-Loop (SITL) environment was set up. In this environment, the original Ardupilot and DUNE embedded software are used. The aircraft platform is simulated by JSBSim [11], an open source Flight Dynamics Model. Therefore, it is able to compute the UAV dynamics according to the actuator controls of the Ardupilot.

The JSBSim was modified to use in its calculations the same wind map that is used by the MPC optimization. Also, an aircraft platform model must be chosen for JSBSim flight dynamics calculations. In this work, the X8 UAV (Figure A.6) model [48] was used.

Figure A.6: X8 UAV. (Source: NTNU)

Figure A.7 shows the interconnection between modules. For each UAV, an Ardupilot SITL instance must be started linked to a JSBSim module. Each Ardupilot instance uses a different TCP port. Therefore, one DUNE module must be started for each UAV, set with the correspondent TCP port. Finally, Neptus [104], a command and control software which is also part of the LSTS' toolchain, is used to visualize the UAVs telemetry and location and to give commands to the UAVs, such as take off, loiter and to start/stop the Search and Rescue mission.

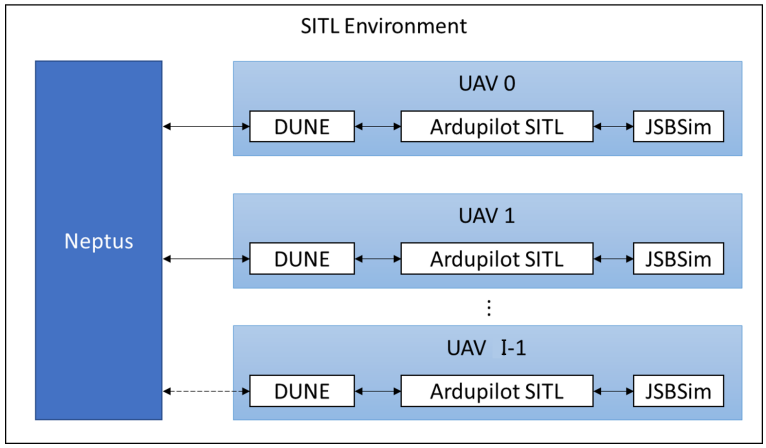


Figure A.7: Software-In-The-Loop setup.

## A.7 Mission Simulation Scenario and Parameters

In this section, the parameters that define the mission scenario are described.

### A.7.1 Aircraft platform

In this work, the Skywalker X8 UAV (Figure A.6) was chosen to be the aircraft platform. The X8 is a battery powered small UAV which can fly for around 80 min with the automated remote sensing payload and proper battery. The radius of the remote sensor is equal to 200 m, which is half of the width of the sensor's footprint. This footprint was chosen assuming that a computer vision algorithm,

such as the one described in the Appendix (same as in [116]), can detect the target in images captured at 400 m of altitude by an infrared camera with 7.5 mm of lens focal length, 640 × 480 pixels of resolution and 17 µm of pixel size.

### A.7.2 Search Domain

The reference search area used in this work is equivalent to the Search Effort ($Z_a$) of one X8 UAV, calculated by Equation A.13. Considering the total endurance of 80 min, the On-Scene Endurance ($T$) is equal to 60 min (85% of the total endurance). The Search Facility Speed ($V$) is equal to the average airspeed of the aircraft, in this case 16 m/s. The Sweep Width ($W$) is equal to 400 m, which is the lateral length of the required sensor footprint. Therefore, the search area is equal to 23.04 km$^2$, which gives a length and width equal to 4.8 km as the area has a squared shape because the datum is a single point.

### A.7.3 Cells Grid

The grid was built with a cell width of 100 m. Therefore, the 23.04 km$^2$ were divided into 48 × 48 cells.

A two dimensional normal distribution curve was fitted to the single point datum reference table provided by the IAMSAR manual (Figure A.1).

The fitted curve of the probability (Figure A.8), that gives the reward of each cell is given by:

$$\phi_{mn} = 0.002946 \exp\Big( - \Big( \frac{(m - 23.5)^2}{108.28} + \frac{(n - 23.5)^2}{108.28} \Big) \Big) \tag{A.19}$$

where $m$ and $n$ are the horizontal and vertical indexes of the cell, respectively.

### A.7.4 MPC Parameters

The boundaries and constraints of the control problem were chosen as follows.

The airspeed range was chosen to be between 12 to 22 m/s. The reason for this choice was to keep the airspeed around the cruise speed, so that the battery consumption does not get too high. The roll angle range was chosen to be between -45 and 45 degrees so that the aircraft performs smoother maneuvers but still with freedom. The safe distance between UAVs was chosen to be 100 m and a wind of 9.9 m/s pointing to 45 deg was considered both in the optimization and in the flight dynamics simulation.

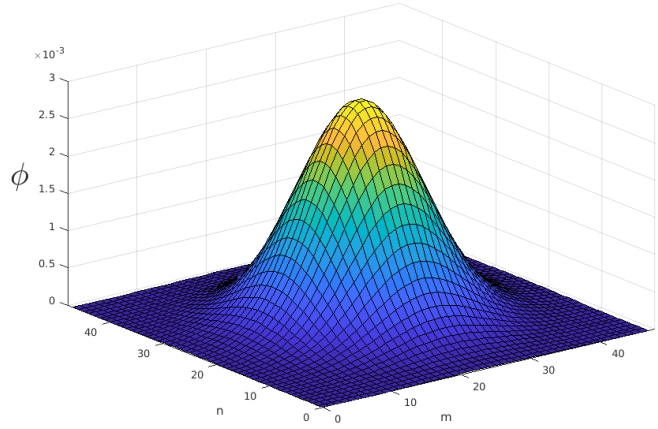


Figure A.8: Reward of cells.

The time horizon of 20 s and 20 horizon steps were the parameters chosen for the MPC problem. With, for example, a ground speed of 17 m/s, this means 340 m of straight distance, or a 180 deg turn. The weighting factor $a$ was chosen to be 10000, because the rewards are of a very low value (the sum of all cells rewards is equal to one). The weighting factors $b$ and $c$ were chosen to be 0, so the algorithm does not consider the airspeed and roll angle constraints for aggressive maneuvers. Therefore, the algorithm is free to only consider the search performance in the optimization.

Regarding the PSO parameters for the optimization, a total of 384 particles was used and the algorithm runs 35 iterations with local and global coefficients of 1.

### A.7.5 Simulation platform

With these parameters, each UAV is able to run the optimization in around 400 ms when 3 UAVs perform the optimization at the same time. The optimization algorithm was written in CUDA C programming language in order to benefit from the parallelism, with the goal to embedded it on a NVIDIA Jetson board in the future for field tests.

The simulations were run in a laptop with the NVIDIA 940MX graphics card, which has around five times processing power more than the NVIDIA Jetson TX2, according to performance tests available in the manufacturer's website. Therefore, it is expected that the embedded board will also be able to run the optimization at 400 ms with just a fine tunning of the parameters. Another possibility is to

implement a optimization stopping feature that will run as many iterations as possible within a given time, instead of a fixed number of iterations.

It is also relevant to mention that the optimization time was adjusted to 400 ms also when using only one or two UAVs. This was done by inserting a delay, so the optimization time in each UAV is the same for simulations when one, two or three UAVs are used. Therefore, this gives a fair comparison of the results.

## A.8 Results and Discussion

Three operational profiles were evaluated for the mission scenario: employing only one UAV; employing two UAVs; or employing three UAVs. Five missions were executed for each one of the profiles in order to obtain the average performance.

The reference search area was the Total Adjusted Search Area ($A_t$) for one UAV facility and Probability of Detection (POD) equal to 1, as described in the previous sections.

The area was kept the same when employing two or three UAVs in order to allow a proper performance comparison between the profiles. Figure A.9 illustrates one mission with three UAVs being monitored by the Command and Control software Neptus. The light red area is the search area and the dark red cross in the middle is the single point datum.

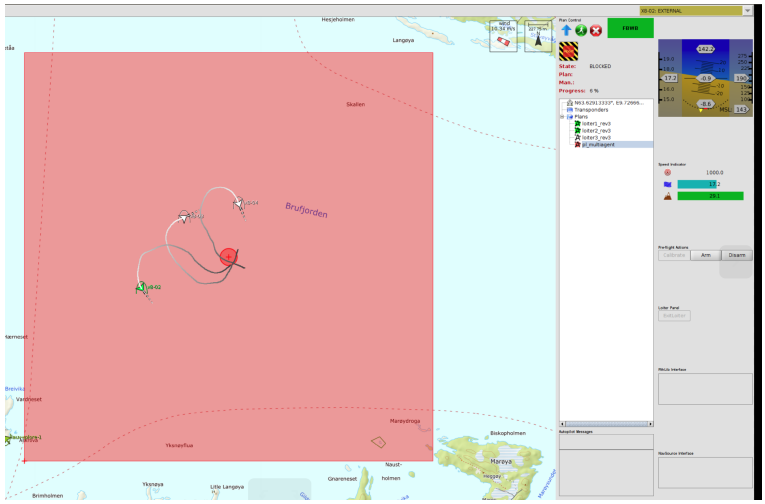


Figure A.9: Snapshot of a mission with 3 UAVs being monitored with Neptus.

In all missions the UAVs departed from the same region (southeast of the search area as shown in Figure A.10) where they were loitering and waiting for the com-

mand. After receiving the command, the UAVs departed to the search area and the mission time started to count from when the first UAV collected the first reward.

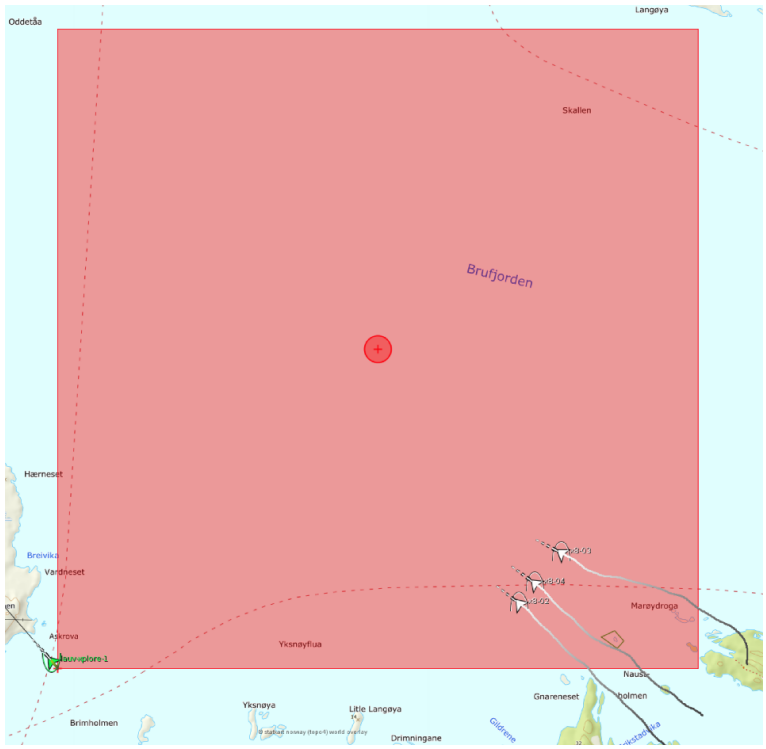


Figure A.10: Snapshot of the beginning of a mission.

The IAMSAR manual describes the Probability of success (POS) as the probability of finding the search object with a particular search. For each sub-area searched, $POS = POD \times POC$. It is therefore the way to measure search effectiveness. As the Probability of detection (POD) is kept at 1, the POS is equal to the POC of the searched area, which in this work is the sum of all rewards collected by the group of UAVs.

The boxplot of the time to reach 50% of POS is shown in Figure A.11 for the three operational profiles: employing one; two; or three UAVs. It is possible to notice that the gain when a pair of UAVs is used is very significant when compared to the single UAV profile, reaching 50% of POS 75% faster. When adding a third UAV, the gain was less significant: on average, the group reached 50% of POS 28% faster than when employing a pair of UAVs. The decrease on the gain is probably due to the fact that the UAVs are often flying over areas that have already been flown. A possible solution to avoid this situation is to reduce the width of the cells, increasing the resolution of the grid. Therefore, the UAVs would better tune their

maneuvers and still have the cells inside the UAVs' sensor radius. However, this will increase the required computational power. This issue could be mitigated by optimizing the algorithm, for example.

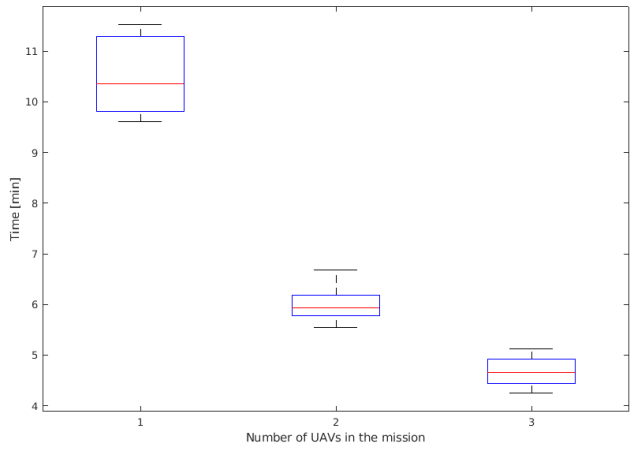


Figure A.11: Time to reach 50% of Probability of success (POS).

Figure A.12 shows the average POS during 20 min of mission. It is possible to notice that the results match the observed behavior when the missions were monitored. From the mission start to around 4 min the UAVs fly to the area close to the datum, where the reward (Probability of containment) is higher. When two UAVs are employed, they fly parallel so that they cover more cells than when employing a single UAV. However, when three UAVs are employed, even if they form a parallel path, they fly close to each other and, therefore, do not visit more cells than the pair of UAVs. This happens because in case the three UAVs get far enough from each other to not visit the same cells, they would take a longer path to arrive at the central area (highest rewards), not being a cost beneficial solution.

After reaching the area close to the datum at around 4 min, the curve of rewards collection grows steadily and the difference between the three operational profiles is clear. When three UAVs were employed, the group reached 50% of POS 2.25 times faster than the single UAV. The gain, however, reduces over time. For example, to reach 65% of POS, the group of three UAVs did it 2.04 times faster than the single UAV. The reason of the decrease on the gain is that the more cells are already visited and rewards collected, further the UAVs have to fly to visit new cells and collect new rewards (that are also lower in value). Therefore, the closer it is to the end of the missions, smaller is the difference between the performance of the different operational profiles.

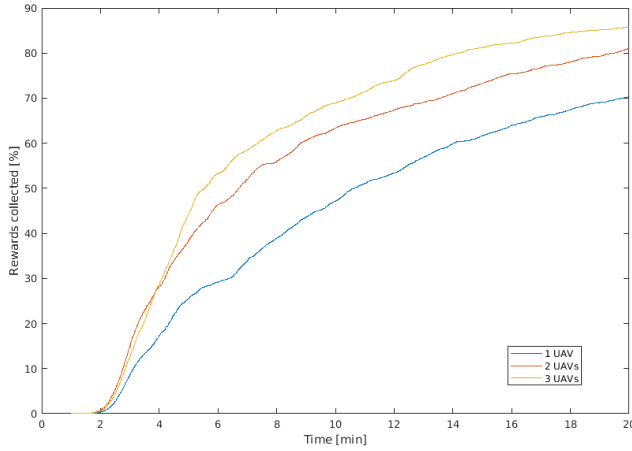


Figure A.12: Average Probability of Success in time.

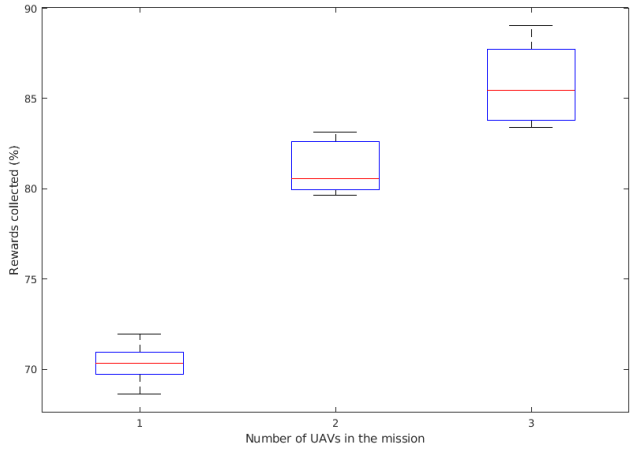


Figure A.13: Probability of success (POS) in 20 minutes of mission.

Figure A.13 shows the boxplot of the POS after 20 min of mission for the three operational profiles. According to the IAMSAR Manual calculations, the single UAV is expected to reach 100% of POS in 60 min. It is possible to notice that the group of three UAVs is able to reach close to 90% of POS in 20 min, showing that the improvement of adding extra UAVs is approximately linear.

Finally, a pre-made path where the UAV flies from the origin to the grid midpoint then flies a spiral path was created (Figure A.14) in order to compare to the

performance of the single UAV with the real-time MPC optimization. This spiral path is close to the standard path suggested by the IAMSAR Manual.

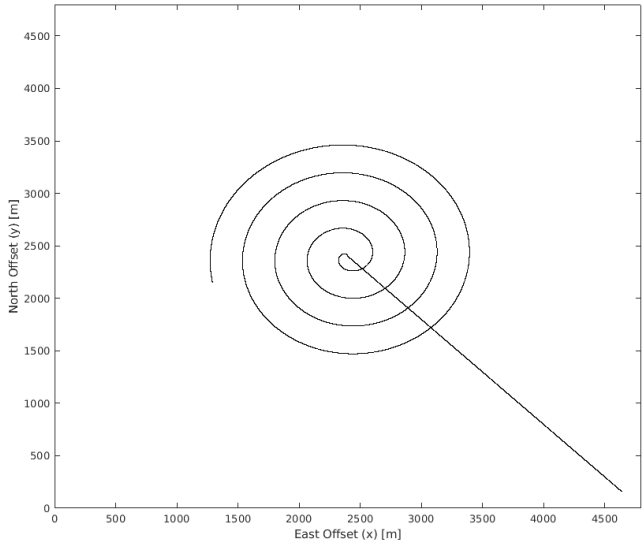


Figure A.14: Spiral path.

In the spiral path, the lanes are equally spaced allowing the best coverage by the sensor's footprint. This would be the best possible simple standard path for the mission scenario being investigated. Also, the UAV is assumed to first fly to the center of the area and then start the spiral path.

Figure A.15 shows that for the same average airspeed of 15.5 m/s, the performance of the MPC path-planning was superior in the first 20 minutes of mission. Also, in the spiral path, 50% of POS was reached in around 13 min, while it took less than 11 min when the MPC optimization was used.

In the spiral path, wind was not considered and the UAV keeps the ground speed constant, while in the MPC path-planning the UAV optimizes its speed to reach higher coverage, for example reducing the airspeed to achieve a steeper turn when needed. Another advantage, that is perhaps the most important, is that the MPC solution has the capability to deal with dynamic changes in the environment and mission parameters during the mission, as it is a real-time optimization. These changes can be wind variations, updated search and rescue reports or even the lost of one UAV in the middle of the mission due to technical problems or the addition of extra UAVs that arrived later when the mission had already started.

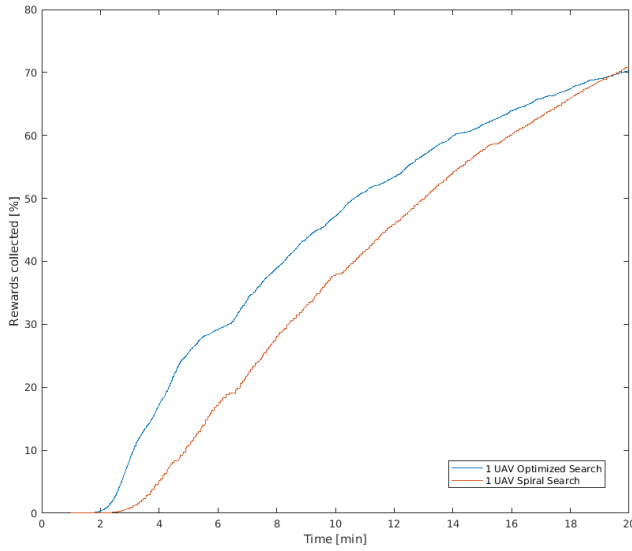


Figure A.15: Probability of Success.

## A.9 Conclusions

In this chapter, a real-time path-planning for search and rescue with Model Predictive Control solved by Particle Swarm Optimization was proposed. The solution was implemented on a fully embedded software and tested in a Software-In-The-Loop environment with flight dynamics simulations. The search area was defined using the International Aeronautical and Maritime Search and Rescue (IAMSAR) directives. Also, the area was divided into a grid of cells, where each cell had a correspondent reward, referred to the IAMSAR's Probability of containment. Results were analyzed for missions where one, two or three Unmanned Aerial Vehicles (UAVs) were employed. To reach 50% of Probability of success, the performance of the group of three UAVs was on average 2.25 times faster than the single UAV search. The performance of the single UAV was also compared to a standard search pattern based on the IAMSAR's suggested pattern. The search using the proposed solution outperformed the standard search pattern in the first 20 min, with the additional advantage of being a real-time method that can deal with environmental dynamic changes and new mission directives.

# References


[1] *Station-Keeping Trials in Ice: Test Scenarios*, volume Volume 8: Polar and Arctic Sciences and Technology; Petroleum Technology of *International Conference on Offshore Mechanics and Arctic Engineering*, 06 2018. doi: 10.1115/OMAE2018-78587.

[2] Ardupilot. ardupilot.org, 2019. [Online; accessed 2019-07-10].

[3] Mavlink micro air vehicle protocol. mavlink.io/en, 2019. [Online; accessed 2019-07-10].

[4] S. Albrektsen and T. Johansen. User-configurable timing and navigation for uavs. *Sensors*, 18(8):2468, 2018.

[5] S. Amari, N. Murata, K. R. Muller, M. Finke, and H. H. Yang. Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks*, 8(5):985–996, Sep 1997.

[6] P. Bajorski. *Statistics for Imaging, Optics, and Photonics*. SPIE monograph. Wiley, 2012. ISBN 9781118303603.

[7] R. W. Beard and T. W. McLain. *Small unmanned aircraft: Theory and practice*. Princeton university press, 2012.

[8] M. B. Bejiga, A. Zeggada, and F. Melgani. Convolutional neural networks for near real-time object detection from uav imagery in avalanche search and rescue operations. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.

[9] R. Bencatel, M. Faied, J. Sousa, and A. R. Girard. Formation control with collision avoidance. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 591–596. IEEE, 2011.

[10] C. Bentes, D. Velotto, and B. Tings. Ship classification in terrasar-x images with convolutional neural networks. *IEEE Journal of Oceanic Engineering*, 43(1):258–266, Jan 2018.

[11] J. Berndt. Jsbsim: An open source flight dynamics model in c++. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 4923, 2004.

[12] J. Bischof. *Ice drift, ocean circulation and climate change*. Springer, 2000.

[13] D. Bloisi, L. Iocchi, M. Fiorini, and G. Graziano. Automatic maritime surveillance with visual target detection. In *Proceedings of the International Defense and Homeland Security Simulation Workshop (DHSS)*, pages 141–145, 2011.

[14] G. D. Boreman. *Modulation transfer function in optical and electro-optical systems*, volume 4. SPIE press Bellingham, WA, 2001.

[15] A. Borghgraef, O. Barnich, F. Lapierre, M. Van Droogenbroeck, W. Philips, and M. Acheroy. An evaluation of pixel-based methods for the detection of floating objects on the sea surface. *EURASIP Journal on Advances in Signal Processing*, 2010(1), Feb 2010.

[16] A. Bosco, A. Bruna, S. Battiato, G. Bella, and G. Puglisi. Digital video stabilization through curve warping techniques. *Consumer Electronics, IEEE Transactions on*, 54:220 – 224, 06 2008. doi: 10.1109/TCE.2008.4560078.

[17] R. Bourne. *Fundamentals of Digital Imaging in Medicine*. Springer London, 2010. ISBN 9781848820876.

[18] N. J. Brake. Control system development for small uav gimbal. Master's thesis, California Polytechnic State University, San Luis Obispo, August 2012.

[19] Ø. Breivik, A. A. Allen, C. Maisondieu, and J. C. Roth. Wind-induced drift of objects at sea: The leeway field method. *Applied Ocean Research*, 33(2): 100–109, 2011.

[20] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965. ISSN 0018-8670. doi: 10.1147/sj.41.0025.

[21] W. Burger and M. Burge. *Digital Image Processing: An Algorithmic Introduction Using Java*. Springer London, 2016. ISBN 9781447166849.

[22] E. F. Camacho, D. R. Ramirez, D. Limón, D. M. De La Peña, and T. Alamo. Model predictive control techniques for hybrid systems. *Annual reviews in control*, 34(1):21–31, 2010.

[23] T. Can, A. O. Karali, and T. Aytaç. Detection and tracking of sea-surface targets in infrared and visual band videos using the bag-of-features technique with scale-invariant feature transform. *Applied Optics*, 50(33):6302–6312, Nov 2011.

[24] Z. D. Caratao, K. F. Gabel, A. Arun, B. Myers, D. L. Swartzendruber, and C. W. Lum. Micasense aerial pointing and stabilization system: Dampening in-flight vibrations for improved agricultural imaging. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, page 1239. 2018.

[25] J. Chen, J. Wu, G. Chen, W. Dong, and X. Sheng. Design and development of a multi-rotor unmanned aerial vehicle system for bridge inspection. In *International Conference on Intelligent Robotics and Applications*, pages 498–510. Springer, 2016.

[26] K. Chen, K. Fu, M. Yan, X. Gao, X. Sun, and X. Wei. Semantic segmentation of aerial images with shuffling convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):173–177, Feb 2018.

[27] G. Cruz and A. Bernardino. Aerial detection in maritime scenarios using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 373–384. Springer, 2016.

[28] C. Dahlin Rodin, F. A. de Alcantara Andrade, A. R. Hovenburg, and T. A. Johansen. A survey of practical design considerations of optical imaging stabilization systems for small unmanned aerial systems. *Sensors*, 19(21): 4800, 2019.

[29] F. A. de Alcantara Andrade, C. D. Rodin, A. R. Hovenburg, T. A. Johansen, and R. Storvold. Path planning of multi-UAS communication relay by decentralized MPC. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pages 1–8. IEEE, 2018.

[30] F. A. de Alcantara Andrade, A. Reinier Hovenburg, L. Netto de Lima, C. Dahlin Rodin, T. A. Johansen, R. Storvold, C. A. Moraes Correia, and D. Barreto Haddad. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. *Sensors*, 19(19):4067, 2019.

[31] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag New York, 1978.

[32] B. Di, R. Zhou, and H. Duan. Potential field based receding horizon motion planning for centrality-aware multiple uav cooperative surveillance. *Aerospace Science and Technology*, 46:386–397, 2015.

[33] M. J. Doty, C. R. Fuller, N. H. Schiller, and T. L. Turner. Active noise control of radiated noise from jets. *NASA, Langley Research Center, Hampton, Virginia, USA*, 2013.

[34] S. J. Dumble and P. W. Gibbens. Efficient Terrain-Aided Visual Horizon Based Attitude Estimation and Localization. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 78(2):205–221, May 2015. ISSN 15730409. doi: 10.1007/s10846-014-0043-8.

[35] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE, 1995.

[36] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 29(4):551–559, 1983.

[37] B. Eggleston, B. McLuckie, W. R. Koski, D. Bird, C. Patterson, D. Bohdanov, H. Liu, T. Mathews, and G. Gamage. Development of the brican td100 small uas and payload trials. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W4:143–149, 2015.

[38] H. Egna and C. Boyd. *Dynamics of Pond Aquaculture*. Taylor & Francis, 1997. ISBN 9781566702744.

[39] A. El-Rabbany. *Introduction to GPS : the Global Positioning System*. Artech House, 2002. ISBN 1580531830.

[40] FLIR. How to evaluate camera sensitivity. flir.com/discover/iis/machine-vision/how-to-evaluate-camera-sensitivity/, . [Online; accessed 2019-08-29].

[41] FLIR. Pan-tilt unit-d48e. flir.com/mcs/view/?id=53670, . [Online; accessed 2017-03-22].

[42] E. J. Forsmo, E. I. Grøtli, T. I. Fossen, and T. A. Johansen. Optimal search mission with unmanned aerial vehicles using mixed integer linear programming. In *2013 International conference on unmanned aircraft systems (ICUAS)*, pages 253–259. IEEE, 2013.

[43] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of mathematics and artificial intelligence*, 31(1-4): 77–98, 2001.

[44] K. Gade. The Seven Ways to Find Heading. *Journal of Navigation*, 69(05): 955–970, 2016. ISSN 0373-4633. doi: 10.1017/S0373463316000096.

[45] R. Gonzalez and R. Woods. *Digital Image Processing*. Pearson/Prentice Hall, 2008. ISBN 9780131687288.

[46] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[47] B. Grelsson, M. Felsberg, and F. Isaksson. Highly accurate attitude estimation via horizon detection. *Journal of Field Robotics*, 33(7):967–993, 2016. ISSN 1556-4967. doi: 10.1002/rob.21639.

[48] K. Gryte, R. Hann, M. Alam, J. Roháč, T. A. Johansen, and T. I. Fossen. Aerodynamic modeling of the skywalker x8 fixed-wing unmanned aerial vehicle. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 826–835. IEEE, 2018.

[49] S. Gudmundsson. *General aviation aircraft design: Applied Methods and Procedures*. Butterworth-Heinemann, 2013.

[50] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48, 2016. ISSN 0925-2312.

[51] V. Gupta and S. Brennan. Terrain-based vehicle orientation estimation combining vision and inertial measurements. *Journal of Field Robotics*, 25(3): 181–202, 2008. ISSN 1556-4967. doi: 10.1002/rob.20233.

[52] A. Gurtner, R. Walker, and W. Boles. Vibration compensation for fish-eye lenses in uav applications. In *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*, pages 218–225. IEEE, 2007.

[53] A. Gurtner, D. G. Greer, R. Glassock, L. Mejias, R. A. Walker, and W. W. Boles. Investigation of fish-eye lenses for small-uav aerial photography. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3):709–721, 2009.

[54] J. Haghshenas. Effects of satellite platform's vibrations on the image quality of a remote sensing payload: system level design and challenges. In *Optical Systems Design 2015: Optical Design and Engineering VI*, volume 9626, page 96262P. International Society for Optics and Photonics, 2015.

[55] M. Hals and M. Skjønhaug. Optimization of coordinated path planning for autonomous vehicles in ice management. Master's thesis, NTNU, 2017.

[56] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition, 2004. ISBN 0521540518.

[57] J. Haugen, L. Imsland, S. Løset, R. Skjetne, et al. Ice observer system for ice management operations. In *Proc. International Offshore and Polar Engineering Conference*, Maui, Hawaii, USA, 2011.

[58] J. Hilkert. Inertially stabilized platform technology concepts and principles. *IEEE Control Systems Magazine*, 28(1):26–46, Feb 2008.

[59] L. L. Hill. *Georeferencing: The geographic associations of information*. Mit Press, 2009.

[60] E. Honkavaara, H. Saari, J. Kaivosoja, I. Pölönen, T. Hakala, P. Litkey, J. Mäkynen, and L. Pesonen. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. *Remote Sensing*, 5(10):5006–5039, 2013.

[61] A. R. Hovenburg, F. A. A. Andrade, R. Hann, C. D. Rodin, T. A. Johansen, and R. Storvold. Long range path planning using an aircraft performance model for battery powered sUAS equipped with icing protection system. Submitted for publication.

[62] A. R. Hovenburg, F. A. de Alcantara Andrade, C. D. Rodin, T. A. Johansen, and R. Storvold. Contingency path planning for hybrid-electric UAS. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 37–42. IEEE, 2017.

[63] A. R. Hovenburg, T. A. Johansen, and R. Storvold. Mission performance trade-offs of battery-powered suas. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 601–608. IEEE, 2017.

[64] A. R. Hovenburg, F. A. de Alcantara Andrade, C. D. Rodin, T. A. Johansen, and R. Storvold. Inclusion of horizontal wind maps in path planning optimization of UAS. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 513–520. IEEE, 2018.

[65] I. IAMSAR. International aeronautical and maritime search and rescue manual. *Mission coordination*, 2, 2007.

[66] International Maritime Organization and International Civil Aviation Organization. *IAMSAR Manual: Vol. 2: Mission Co-Ordination*. International Maritime Organization, 2007.

[67] M. Jakob, E. Semsch, D. Pavlicek, and M. Pechoucek. Occlusion-aware multi-uav surveillance of multiple urban areas. In *6th Workshop on Agents in Traffic and Transportation (ATT 2010)*, pages 59–66, 2010.

[68] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke. Deep learning for infrared thermal image based machine health monitoring. *IEEE/ASME Transactions on Mechatronics*, 2017.

[69] O. Johannessen, V. Alexandrov, I. Frolov, S. Sandven, L. Pettersson, L. Bobylev, K. Kloster, V. Smirnov, Y. Mironov, and N. Babich. *Remote Sensing of Sea Ice in the Northern Sea Route: Studies and Applications*. Springer Praxis Books. Springer Berlin Heidelberg, 2006.

[70] D. Johansen, J. Hall, R. Beard, and C. Taylor. Stabilization of video from miniature air vehicles. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.

[71] V. John, S. Mita, Z. Liu, and B. Qi. Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In *Machine Vision Applications (MVA), 14th IAPR International Conference on*, 2015.

[72] A. Johnson. *Plane and geodetic surveying: the management of control networks*. Spon Press, 2004. ISBN 0415320038.

[73] C. F. F. Karney. Transverse mercator with an accuracy of a few nanometers. *Journal of Geodesy*, 85(8):475–485, 2011. ISSN 1432-1394. doi: 10.1007/s00190-011-0445-3.

[74] G. N. Karystinos and D. A. Pados. On overfitting, generalization, and randomly expanded training sets. *IEEE Transactions on Neural Networks*, 11 (5):1050–1057, Sep 2000.

[75] B. Kellenberger, M. Volpi, and D. Tuia. Fast animal detection in uav images using convolutional neural networks. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[76] P. J. Kennedy and R. L. Kennedy. Direct versus indirect line of sight (los) stabilization. *IEEE Transactions on control systems technology*, 11(1):3–15, 2003.

[77] M. Kriss. *Handbook of Digital Imaging*. Wiley, 2015. ISBN 9780470510599.

[78] S. J. Lee, T. Chen, L. Yu, and C. H. Lai. Image classification based on the boost convolutional neural network. *IEEE Access*, 6, 2018.

[79] F. S. Leira, T. A. Johansen, and T. I. Fossen. Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera. In *IEEE Aerospace Conference*, Big Sky, Montana, 2015.

[80] F. S. Leira, K. Trnka, T. I. Fossen, and T. A. Johansen. A ligth-weight thermal camera payload with georeferencing capabilities for small fixed-wing UAVs. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 485–494. IEEE, Jun 2015.

[81] F. S. Leira, T. A. Johansen, and T. I. Fossen. A uav ice tracking framework for autonomous sea ice management. In *Proc. IEEE International Conference on Unmanned Aircraft Systems (ICUAS'17)*, Miami, Florida, USA, 2017.

[82] G. J. Leishman. *Principles of helicopter aerodynamics*. Cambridge university press, 2006.

[83] C. Li and F. Tan. Effect of uav vibration on imaging quality of binary optical elements. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1693–1698. IEEE, 2018.

[84] T.-H. S. Li, C.-C. Chen, and Y.-T. Su. Optical image stabilizing system using fuzzy sliding-mode controller for digital cameras. *IEEE Transactions on Consumer Electronics*, 58(2):237–245, 2012.

[85] Z. Li, Y. Yan, Y. Jing, and S. Zhao. The design and testing of a lidar platform for a uav for heritage mapping. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1):17, 2015.

[86] P. Liferov, T. McKeever, F. Scibilia, S. H. Teigen, A. Kjøl, E. Almkvist, and J. K. Lindvall. Station-keeping trials in ice: Project overview. In *ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering*, pages V008T07A029–V008T07A029. American Society of Mechanical Engineers, 2018.

[87] S. Lim and D. Lee. Stable improved softmax using constant normalisation. *Electronics Letters*, 53(23):1504–1506, 2017.

[88] N. Liu, J. Han, T. Liu, and X. Li. Learning to predict eye fixations via multiresolution convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(2):392–404, Feb 2018.

[89] S. Long. Drones and precision agriculture: The future of farming. microdrones.com/en/content/drones-and-precision-agriculture-the-future-of-farming, 2017. [Online; accessed 2019-08-29].

[90] P. Lu and Z. Li. A method of obtaining ice concentration and floe size from shipboard oblique sea ice images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(7):2771–2780, July 2010. ISSN 0196-2892. doi: 10.1109/TGRS.2010.2042962.

[91] H. Ma and J. Wu. Analysis of positioning errors caused by platform vibration of airborne lidar system. In *Instrumentation and Control Technology (ISICT), 2012 8th IEEE International Symposium on*, pages 257–261. IEEE, 2012.

[92] F. Maire, L. Mejias, and A. Hodgson. A convolutional neural network for automatic analysis of aerial imagery. In *Digital lmage Computing: Techniques and Applications (DlCTA), 2014 International Conference on*, 2014.

[93] F. L. Markley. Equivalence of Two Solutions of Wahba's Problem. *Journal of the Astronautical Sciences*, 60:303–312, Dec. 2013. doi: 10.1007/s40295-015-0049-x.

[94] M. Masten. Inertially stabilized platforms for optical imaging systems. *IEEE Control Systems Magazine*, 28(1):47–64, Feb 2008.

[95] R. Miller, G. Mooty, and J. M. Hilkert. Gimbal system configurations and line-of-sight control techniques for small uav applications. In *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications X*, volume 8713, 2013. doi: 10.1117/12.2015777.

[96] M. Mizui, I. Yamamoto, and R. Ohsawa. Effects of propeller-balance on sensors in small-scale unmanned aerial vehicle. *IOSRJEN*, 2(8):23–27, 2012.

[97] T. Nakazawa and D. V. Kulkarni. Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing*, 2018.

[98] N. Nigam. The multiple unmanned air vehicle persistent surveillance problem: A review. *Machines*, 2(1):13–72, 2014.

[99] M. Nixon and A. S. Aguado. *Feature Extraction & Image Processing for Computer Vision*. Academic Press, 3rd edition, 2012. ISBN 0123965497, 9780123965493.

[100] M. Odelga, N. Kochanek, and H. H. Bülthoff. Efficient real-time video stabilization for uavs using only imu data. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 210–215. IEEE, 2017.

[101] U. D. of Defense". Department of defense dictionary of military and associated terms. *Joint Publication*, 2007.

[102] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[103] D. K. Perovich and K. F. Jones. The seasonal evolution of sea ice floe size distribution. *Journal of Geophysical Research: Oceans*, 119(12):8767–8777, 2014. ISSN 2169-9291.

[104] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa. The lsts toolchain for networked vehicle systems. In *2013 MTS/IEEE OCEANS-Bergen*. IEEE, 2013.

[105] K. Plataniotis and A. Venetsanopoulos. *Color Image Processing and Applications*. Digital Signal Processing. Springer Berlin Heidelberg, 2000. ISBN 9783540669531.

[106] L. Porzi, S. Rota Bulò, and E. Ricci. A Deeply-Supervised Deconvolutional Network for Horizon Line Detection. In *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pages 137–141, New York, New York, USA, 2016. ACM Press. ISBN 9781450336031. doi: 10.1145/2964284.2967198.

[107] L. Prechelt. Early stopping-but when? *Neural Networks: Tricks of the trade*, 1998.

[108] S. Prince. *Computer Vision: Models, Learning, and Inference*. Computer Vision: Models, Learning, and Inference. Cambridge University Press, 2012. ISBN 9781107011793.

[109] I. Prodan, S. Olaru, R. Bencatel, J. B. de Sousa, C. Stoica, and S.-I. Niculescu. Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Control Engineering Practice*, 21(10):1334–1349, 2013.

[110] D. Raymer. *Aircraft Design: A Conceptual Approach 5e and RDSWin STUDENT*. American Institute of Aeronautics and Astronautics, Inc., 2012.

[111] K. Reddy, H. Lokesha, and S. Akhila. Video stabilization for micro air vehicles. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(4):442–445, 2015.

[112] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi. Uav photogrammetry for mapping and 3d modeling–current status and future perspectives. *International archives of the photogrammetry, remote sensing and spatial information sciences*, 38(1):C22, 2011.

[113] C. D. Rodin and T. A. Johansen. Accuracy of sea ice floe size observation from an aerial camera at slant angles. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, Oct 2017. doi: 10.1109/RED-UAS.2017.8101669.

[114] C. D. Rodin and T. A. Johansen. Accuracy of sea ice floe size observation from an aerial camera at slant angles. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 216–221. IEEE, 2017.

[115] C. D. Rodin and T. A. Johansen. Detectability of objects at the sea surface in visible light and thermal camera images. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*. IEEE, 2018.

[116] C. D. Rodin, L. N. de Lima, F. A. de Alcantara Andrade, D. B. Haddad, T. A. Johansen, and R. Storvold. Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018.

[117] C. D. Rodin, T. A. Johansen, and A. Stahl. Skyline based camera attitude estimation using a digital surface model. In *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, pages 306–313. IEEE, 2018.

[118] L. Rokach. *Pattern classification using ensemble methods*, volume 75. World Scientific, 2010.

[119] F. A. Rosell and R. H. Willson. Basics of detection, recognition and identification in electro-optical formed imagery. In *Proceedings of SPIE*, volume 33, 1973.

[120] A. Rucco, A. P. Aguiar, F. L. Pereira, and J. B. de Sousa. A predictive path-following approach for fixed-wing unmanned aerial vehicles in presence of wind disturbances. In *Robot 2015: Second Iberian Robotics Conference*, pages 623–634. Springer, 2016.

[121] Y. G. Ryu, H. C. Roh, S. J. Kim, K. H. An, and M. J. Chung. Digital image stabilization for humanoid eyes inspired by human vor system. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2301–2306, Dec 2009. doi: 10.1109/ROBIO.2009.5420460.

[122] R. Sengupta, J. Connors, B. Kehoe, Z. Kim, T. Kuhn, and W. Jared. Final report - autonomous search and rescue with scaneagle, 2010.

[123] M. Shahbazi, G. Sohn, J. Théau, and P. Menard. Development and evaluation of a uav-photogrammetry system for precise 3d environmental modeling. *Sensors*, 15(11):27493–27524, 2015.

[124] H. Shen, Q. Pan, Y. Cheng, and Y. Yu. Fast video stabilization algorithm for uav. In *IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 4, pages 542–546, Nov 2009. doi: 10.1109/ICICISYS.2009.5357609.

[125] C. Sheppard and M. Rahnemoonfar. Real-time scene understanding for uav imagery based on deep convolutional neural networks. In *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[126] M. Shokr and N. Sinha. *Sea Ice: Physics and Remote Sensing*. Geophysical Monograph Series. Wiley, 2015. ISBN 9781119027966.

[127] E. Skjong and S. Nundal. Tracking objects with fixed-wing uav using model predictive control and machine vision. Master's thesis, Norwegian University of Science and Technology (NTNU), 09 2014.

[128] E. Skjong, S. A. Nundal, F. S. Leira, and T. A. Johansen. Autonomous search and tracking of objects using model predictive control of unmanned aerial vehicle and gimbal: Hardware-in-the-loop simulation of payload and avionics. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 904–913. IEEE, 2015.

[129] A. Sobral. BGSLibrary: An opencv c++ background subtraction library. In *IX Workshop de Visão Computacional (WVC'2013)*, Rio de Janeiro, Brazil, Jun 2013. URL https://github.com/andrewssobral/bgslibrary.

[130] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[131] Statens Kartverk. Euref89 ntm (norsk transversal mercator) sone 5 - 30. kartverket.no/globalassets/posisjonstjenester/euref89ntmbeskrivelse.pdf. [Online; accessed 2017-03-20].

[132] Statens Kartverk. Høydedata dom10 2017-03-27. hoydedata.no/LaserInnsyn/Home/DownloadFile?filename=DOM10_20170327.zip, 2017. [Online; accessed 2017-03-28].

[133] Statens Kartverk. Høydedata nhm metadata 2017-03-27. hoydedata.no/LaserInnsyn/Home/DownloadFile?filename=NHM_Metadata_20170320.zip, 2017. [Online; accessed 2017-03-28].

[134] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[135] T. Sumimoto, K. Kuramoto, S. Okada, H. Miyauchi, M. Imade, H. Yamamoto, and T. Kunishi. Machine vision for detection of the rescue target in the marine casualty. In *Industrial Electronics, Control and Instrumentation, IECON '94., 20th International Conference on*, volume 2, pages 723–726, 1994.

[136] R. Thillainayagi and K. S. Kumar. Video stabilization technique for thermal infrared aerial surveillance. In *2016 Online International Conference on*

*Green Engineering and Technologies (IC-GET)*, pages 1–6, Nov 2016. doi: 10.1109/GET.2016.7916630.

[137] M. Tico. *Recent Advances in Signal Processing: Digital Image Stabilization*, chapter 1. 11 2009. ISBN 978-953-307-002-5. doi: 10.5772/7458.

[138] P. Trodden and A. Richards. Multi-vehicle cooperative search using distributed model predictive control. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[139] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[140] B. Uragun and I. N. Tansel. The noise reduction techniques for unmanned air vehicles. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 800–807. IEEE, 2014.

[141] M. Valkonen, K. Kartasalo, K. Liimatainen, M. Nykter, L. Latonen, and P. Ruusuvuori. Dual structured convolutional neural network with feature augmentation for quantitative characterization of tissue histology. In *International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017.

[142] J. Verbeke and S. Debruyne. Vibration analysis of a uav multirotor frame. In *Proceedings of ISMA 2016 International Conference on Noise and Vibration Engineering*, pages 2401–2409, 2016.

[143] R. Vidal, Y. Ma, and S. Sastry. *Generalized Principal Component Analysis*. Interdisciplinary Applied Mathematics. Springer New York, 2016. ISBN 9780387878119.

[144] P. Wadhams. *Ice in the Ocean*. Taylor & Francis, New York, 2000.

[145] Y. Wang, J. Lin, and Z. Wang. An energy-efficient architecture for binary weight convolutional neural networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(2):280–293, Feb 2018.

[146] H. Wieslander, G. Forslid, E. Bengtsson, C. Wahlby, J.-M. Hirsch, C. Runow Stark, and S. Kecheril Sadanandan. Deep convolutional neural networks for detecting cellular changes due to malignancy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 82–89, 2017.

[147] D. Wulich and N. Kopeika. Image resolution limits resulting from mechanical vibrations. *Optical engineering*, 26(6):266529, 1987.

[148] X. Xue, W. Qin, Z. Sun, S. Zhang, L. Zhou, P. Wu, et al. Effects of N-3 UAV spraying methods on the efficiency of insecticides against planthoppers and cnaphalocrocis medinalis. *Acta Phytophylacica Sinica*, 40(3):273–278, 2013.

[149] N. Yang, H. Tang, H. Sun, and X. Yang. Dropband: A simple and effective method for promoting the scene classification accuracy of convolutional neural networks for vhr remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 15(2):257–261, Feb 2018.

[150] T. Yechout. *Introduction to Aircraft Flight Mechanics*. American Institute of Aeronautics and Astronautics, Inc., 2014.

[151] D. Yeom. Optical image stabilizer for digital photographing apparatus. *IEEE Transactions on Consumer Electronics*, 55(3), 2009.

[152] J. Yong-Xin, Y. Qun-Zhe, S. Cheng-Yong, Y. Chang-Qin, and Y. Xi-Yong. Study on detection method of small object on sea based on kernel-mrf foreground segmentation model. In *International Conference on Computer Science and Service System*, pages 1869–1872, 2012.

[153] D. Yu and L. Deng. *Automatic speech recognition: A deep learning approach*. Springer, 2014.

[154] Q. Zhang and R. Skjetne. Image processing for identification of sea-ice floes and the floe size distributions. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2913–2924, May 2015. ISSN 0196-2892. doi: 10.1109/TGRS.2014.2366640.

[155] Z. Zhenming Li, M. Lao, S. K. Phang, M. R. A. Hamid, K. Z. Tang, and F. Lin. Development and Design Methodology of an Anti-Vibration System on Micro-UAVs. In *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, Toulouse, 2017.

ISBN 978-82-326-4252-6 (printed version)
ISBN 978-82-326-4253-3 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2019:328

Christopher Dahlin Rodin

# Applications of High-Precision Optical Imaging Systems for Small Unmanned Aerial Systems in Maritime Environments

**NTNU**
Norwegian University of
Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

Doctoral Thesis
Christopher Dahlin Rodin
Doctoral theses at NTNU, 2019:328
NTNU

NTNU
Norwegian University of
Science and Technology