ISBN 978-82-326-4344-8 (printed version)
ISBN 978-82-326-4345-5 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Håkon Hagen Helgesen

Doctoral Thesis

Doctoral theses at NTNU, 2019:373

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

NTNU

Doctoral theses at NTNU, 2019:373

Håkon Hagen Helgesen

Detection and Tracking of Floating Objects using UAVs with Optical Sensors

NTNU
Norwegian University of
Science and Technology

Håkon Hagen Helgesen

Detection and Tracking of Floating Objects using UAVs with Optical Sensors

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 978-82-326-4344-8 (printed version)
ISBN 978-82-326-4345-5 (electronic version)
ISSN 1503-8181

ITK-report: 2019-10-W

Doctoral theses at NTNU, 2019:373

MILJØMERKET 241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

To mom and dad

Summary

This thesis focuses on detection and tracking of floating objects using fixed-wing unmanned aerial vehicles (UAVs) equipped with a monocular thermal camera. UAVs with optical sensors are useful in a vast number of applications such as search and rescue, inspection, target tracking and surveillance. The usefulness of UAVs in remote sensing applications is going to increase in the future so further research on these topics is needed.

The main motivation behind this work was to identify and develop suitable real-time algorithms for detection and tracking of objects located on the sea surface. Moreover, real-time performance on small embedded computers was desired and used as a guideline. Much work has been carried out to study detection and tracking on stationary platforms, but less work has been conducted to find methods that fit the operating envelope of fixed-wing UAVs. This thesis aims to provide new insight into detection and tracking, and handle issues that arise from rapid camera motion.

Another objective was to provide new insight through analysis of experimental data. The majority of the results presented in this thesis are based on experimental data that have been collected in several independent field experiments. A huge effort was made to plan and conduct these experiments, both with regard to payload development and mission design. It is hard to simulate detection and tracking in a realistic manner because some of the most prominent issues are difficult to recreate in simulations. Therefore, a goal was to collect much experimental data and process the data using different methods. Obviously, this has influenced the direction of this work because issues revealed along the way have been prioritized and investigated later in the research period.

Tracking and georeferencing of floating objects have been the main tasks studied in this thesis. Consequently, some of the methods are tailored for these applications. Nevertheless, the methods are also applicable for complementary missions with changes to smaller parts of the system. The experimental analysis has focused both on empirical and theoretical aspects. Improving the accuracy of existing systems has been desirable, but concepts such as optimality and consistency of tracking filters have also been central.

Target tracking is usually divided into three subproblems, namely detection, data or measurement association, and filtering (state estimation). Detection concerns identification of objects within a sensor scan. Object detection in images involves identifying pixels that belong to objects. It is traditionally solved with image processing techniques and more recently with machine learning. The filtering part of target tracking deals with estimation of target states, typically the position and velocity. The filtering part utilizes image detections to improve and correct the predicted states. Measurement association is needed when multiple targets are present in the surveillance region or when clutter and false detections are expected. New detections are related to existing tracks through data association. This thesis looks into every part of the tracking system, but focuses particularly on filtering and state estimation.

The first part of this thesis consists of three chapters which give a fundamental introduction to remote sensing applications, UAVs, optical sensors and target tracking. Research objectives and goals are stated together with description of related and existing work. Chapter 4 to Chapter 7 investigate detection in thermal images, georeferencing of thermal images, and target tracking. Navigation uncertainty influences georeferencing and target tracking in a negative way so one chapter is devoted to tracking in the presence of navigation uncertainty. These four chapters consist of several methods in addition to numerous case studies that investigate the effectiveness of these methods. The thesis also touches upon navigation for UAVs because errors in the UAV pose is a bottleneck with respect to the accuracy and performance of the tracking system. The final part of the dissertation concludes the results and discusses future possibilities.

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 work that forms the foundation of this thesis has been carried out at the Department of Engineering Cybernetics and the NTNU Centre for Autonomous Marine Operations and Systems (NTNU AMOS).

Professor Tor Arne Johansen has been the main supervisor of this work and Professor Thor Inge Fossen the co-supervisor. The Research Council of Norway has co-funded this project through the Centres of Excellence funding scheme with grant number 223254.

Acknowledgments

I am very grateful for the support from my supervisors who entrusted me with the position as a PhD candidate. First of all, I would like to thank my main supervisor Tor Arne for answering e-mails instantly, especially before deadlines. I thought people were joking when they mentioned your response time, but you constantly proved me wrong. I have really appreciated our discussions regarding this work and the support I have received near the end of this period. I would also like to thank my co-supervisor Thor Inge for his support. My work as a teaching assistant in guidance and control of vehicles has been very fulfilling, and I have enjoyed our cooperation in the course. I would also like to thank associate professor Edmund Brekke for including me in supervision of master students on autonomous ships and for showing interest in my work. Your expertise has helped me numerous times during this period.

I would also like to mention the technical staff at the department. People at the electrical and mechanical workshop have been helpful in the payload development. The experimental data used in this work would be impossible to gather without the effort of the skillful UAV operators at the UAV lab. Pål Kvaløy and Lars Semb have been supportive and sacrificed evenings with their families to prepare for experiments. Thank you for a great collaboration. I would also like to thank the administration at the department. You have helped several times with questions related to my work as a teaching assistant, and you ensure that PhD candidates are able to focus solely on their own project.

I have been fortunate enough to cooperate closely with Frederik S. Leira throughout my time here. Initially, you provided me data to work with. Later, we have organized experiments, and developed hardware and software together. I am very grateful for our cooperation and this thesis would not be possible without your effort. I would also like to thank Erik Wilthil for numerous discussions on different topics. You have saved me multiple times. I would also like to thank my officemate Kristoffer Gryte for his support. We moved into the same office as master students and we are still going strong. You have inspired me and your work ethic is astonishing (which can be annoying on days I would like to go home early without feeling bad).

I would also like to thank Torleiv Bryne for helping me in the final two years of my work. I have learned much from you and is thankful for your ability to help colleges. I would also like to mention Robert Rogne and Sigurd Albrektsen who contributed in the payload development.

Mathias, Bjørn-Olav, Andreas, Erik and Kristoffer, we have known each other from our days as master students. At some point during our PhD, you were replaced by the new lunch room and sent to the fourth floor. I like that the distance to the coffee machine is shorter, but you are still missed on the second floor. Erik, Kristoffer, Glenn, Inger and Gunhild, you have supported me so much these years so thank you for that.

I would also like to thank Bugge for keeping me sane. You have been very important for me in my spare time. I look forward to play (win) many tennis matches against you and watch a lot of football in the future.

Finally, I would like to thank my closest family for their unconditional support. Hilde, Eirik and Elisabeth, you have really inspired and supported me these years, and I am proud to see how much you have achieved. Mom and dad, you have always been there for me. I will never forget how much time and effort you have sacrificed to help me in different parts of life, particularly in school/studies and cross-country skiing.

November 2019, Trondheim
Håkon

Contents

Abbreviations

AGL above ground level

ARS angular rate sensor

BLOS beyond line of sight

BRLOS beyond radio line of sight

CV constant velocity

DOF degrees of freedom

ECEF earth centered earth fixed

ECI earth centered inertial

EKF extended Kalman filter

EO electro optical

GNSS global navigation satellite system

GPS global positioning system

HSV hue, saturation and value (color space)

IMM interacting multiple models

IMU inertial measurement unit

IPDA integrated probabilistic data association

JIPDA joint integrated probabilistic data association

JPDA joint probabilistic data association

KF Kalman filter

LWIR long-wave infrared camera

MAE mean absolute error

ME mean error

MEKF multiplicative extended Kalman filter

MTT multiple target tracking

MWIR mid-wave infrared

NE north-east (coordinates)

NED north east down

NEES normalized estimation error squared

NIR near infrared

NIS normalized innovation squared

PDA probabilistic data association

PDAF probabilistic data association filter

PDF probability density function

RGB red, green and blue (color space)

RMS root mean square

RPAS remotely piloted aircraft system

RTK real time kinematic

SD standard deviation

SKF Schmidt-Kalman filter

SWIR short-wave infrared

UAV unmanned aerial vehicle

UKF unscented Kalman filter

VLOS visual line of sight

Chapter 1

Introduction

This thesis is about detection and tracking of floating objects using unmanned aerial vehicles (UAVs) with optical sensors. UAVs with the ability to sense the environment and obtain situational awareness are useful in numerous applications. Moreover, UAVs with remote sensing capability is an interesting research field that is expected to gain increased attention with new technology and easier availability. Topics related to aerial mapping and surveillance using fixed-wing UAVs are of interest in this research. This chapter aims to introduce the reader to mapping and surveillance and set the scope for this thesis. The following sections are a part of the opening chapter:

- Section 1.1 introduces detection and tracking and explains why research on these topics is useful and needed.
- Section 1.2 briefly presents a few research articles concerning UAV applications.
- Section 1.3 states the scope of this thesis and the main research objective.
- Section 1.4 lists the publications that have been written as a part of the work related to this thesis. It also states other contributions.
- Section 1.5 presents the outline for the rest of this thesis and summarizes the contents of each chapter.

1.1 Background & Motivation

A large part of Norway is covered by sea. The coastline stretches from southern parts of Norway to the northern parts. The interior of Norway also includes large areas with water, such as lakes, rivers and fjords. A significant part of Norwegian economy is tied to interests in maritime environments. This includes huge industries such as offshore oil and gas technology, fisheries and salmon farming. Therefore, research that concerns the maritime environment is important for Norway and studied at research institutions such as the center for autonomous marine operations and systems (NTNU AMOS).

There are several important research fields that fall within the maritime domain. Ferries that transport passengers and cars are an important part of the infrastructure in Norway. They are often manually controlled and expensive to operate with human operators. Therefore, research on autonomous ferries is popular both in academia and in the industry. Topics like situational awareness, path planning and collision avoidance are key in the process of facilitating for autonomous operations at sea.

Autonomous ships need to obey the International Regulations for Preventing Collisions at Sea (COLREGS) [17]. The main challenge related to COLREGS is collision avoidance. A similar challenge is ice navigation in ice-infested waters. Therefore, a system for detecting and keeping track of obstacles near the planned path of the ship is required. On-board sensors such as a radar, lidar and camera can be used to detect objects in the proximity of the ship [25, 54, 120]. However, it might not be sufficient to only place sensors on-board the ship. Limited range and resolution as well as objects hidden in waves can make it challenging to detect objects. Fixed-wing UAVs can be used to overcome some of the challenges and make a robust system in combination with sensors on-board the ship. By monitoring the planned path of the ship with a UAV, obstacles that are difficult to find with ship sensors can be located [56].

Monitoring the sea surface is important in several additional applications and industries. Restricting access for foreign objects to certain areas is crucial in operations with high demands for safety. This can for instance be in operations where autonomous underwater vehicles need to approach the sea surface without the risk of facing other vessels or obstacles on the sea surface. It is also important for seismic surveys where evasive maneuvering should be minimized because they often carry large structures behind the ship. These operations require situational awareness in a large area. Consequently, an aerial view of the environment is of great benefit in these circumstances.

Fixed-wing UAVs are an excellent tool in the maritime domain because they can carry multiple sensors for remote sensing, while covering relatively large geographical areas in a short amount of time. This is more efficient than using marine vessels for the same purpose. UAVs have been used in military applications for several years. More recently, UAVs have been used in commercial and civil applications due to increased availability and lower prices. Moreover, their capabilities grow rapidly. UAVs can be used for surveillance of the coastline, which is important to protect national interests. UAVs can also be used to detect environmental threats, such as oil spills in the sea. Search and rescue operations, looking for missing people or vessels, are also missions where UAVs can serve as a critical tool.

The applications mentioned so far fall within the field of situational awareness, which is a broad term covering many research topics. UAVs can be tailored to carry different sensors based on the need. Optical sensors, radars, lidars and hyperspectral imaging sensors are all examples of sensors that fit in UAVs. Information gathered from these sensors can be stored on-board the UAV or broadcasted to other users. Broadcasting raw data is challenging because of bandwidth limitations. Therefore, on-board processing of sensor data is often a vital part of the system so that only the desired and interesting parts of the sensor data are sent to the ground.

This dissertation concerns detection and tracking of floating objects using fixed-wing UAVs, which are useful in many of the applications mentioned already. Detection and tracking are key tasks in mapping and surveillance. Optical sensors are useful for these operations because of the large amount of information that are contained in images. Images can be used to detect undiscovered objects and gather aerial information about the proximity of the UAV. A thermal camera is for example a suitable sensor at sea because floating objects usually have a distinct thermal signature that is distinguishable from the homogeneous sea surface.

Figure 1.1 and Figure 1.2 show two examples of aerial images captured with a visual spectrum and thermal camera, respectively. Both images contain a huge amount of information that illustrate the remote sensing capabilities of optical sensors. The visual spectrum image displays Agdenes airfield close to Trondheim. A blue tarp is detected in this image, which was captured as part of a precision drop mission [79]. The thermal image contains a car road, houses and the sea surface on the left side.

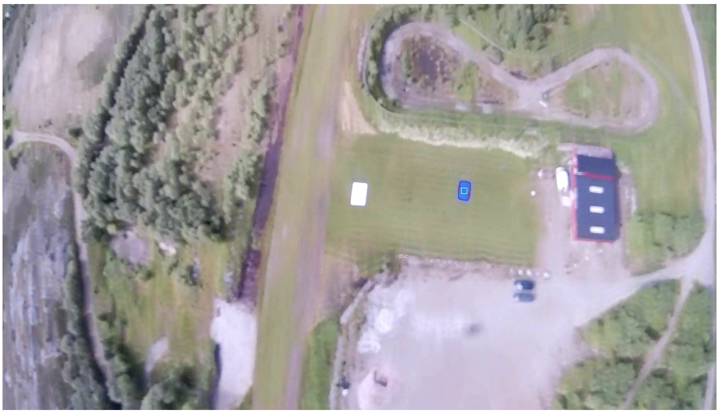


Figure 1.1: Example of visual spectrum image captured by a UAV

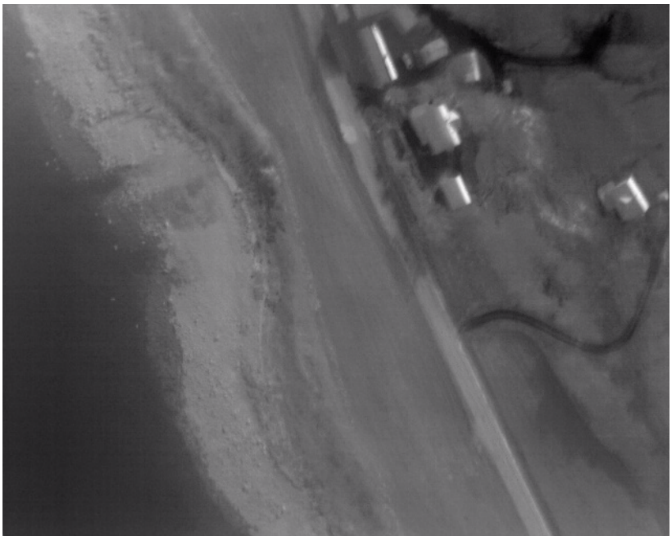


Figure 1.2: Example of thermal image captured by a UAV

1.2 Existing Work

This section describes a small portion of existing research in remote sensing applications. The works mentioned here are primarily meant to give an overview of the diversity of applications where UAVs are useful. Articles that are relevant for topics covered later in this thesis are described in their respective chapters.

One can distinguish research on UAVs into two categories. The first category covers challenges related to the UAV itself and concerns topics that are vital for the UAV to operate in a safe and robust manner. Navigation, guidance, path planning, control design, and sense and avoid technology are examples of research fields that fall within this category. The second category covers situations where UAVs are used as a tool in a larger operation. This is typically related to remote sensing applications where UAVs are used to gather data about the environment.

Optical sensors are useful in both categories. Visual navigation is a popular research field and utilizes images in the navigation system [30, 51, 131]. Moreover, horizon detection is useful for attitude estimation and is described in [23, 50]. Sense and avoid systems can also be based on optical sensors [127]. Optical sensors are also useful for autonomous landing on a moving platform [102].

Most research concern the second category, where remote sensing is the purpose. UAVs are used in various remote sensing applications as described in several surveys [16, 89, 112]. The main motivation for using UAVs in remote sensing is to collect information which is hard to gather without UAVs. It can be because it is difficult to access an area for people or because an aerial view is necessary. Moreover, areas where it is too dangerous to send people can be explored by UAVs. Disaster management is such an example where UAVs are useful because multiple vehicles can be used as first responders and provide an overview of the situation [122]. Earthquakes, floods and hurricanes are threats where UAVs are needed because areas might be unavailable for emergency personnel. Inspection of power lines and industrial facilities are other examples where UAVs are useful [20, 85]. Target tracking of ground objects [100], search and rescue missions [22, 99], and iceberg monitoring [66] are also applications where UAVs are convenient.

1.3 Objective

This thesis aims to provide new and useful insight into detection and tracking of floating objects using fixed-wing UAVs with optical sensors. The thesis takes an experimental approach, which means that the majority of the results are based on experimental data. This data were gathered in several independent flight experiments. Hence, a significant amount of time and effort were devoted to design and develop a payload with sensors capable of collecting relevant data.

The main objective in this research is to find feasible solutions for detection and tracking that can be executed in real-time on small embedded computers. Moreover, identifying and validating solutions that can mitigate the most prominent issues and challenges in target tracking from a moving platform are important. Real-time performance is desired in every part of the pipeline. This means that real-time feasibility for detection, georeferencing and target tracking is desired. The real-time requirement has served as a guideline and obviously affected many of the choices throughout the research period.

The experimental focus has to some degree restricted the theoretical contribution. It has been desirable to identify strengths and weaknesses with this type of mapping and surveillance system. Moreover, it has also been of interest to investigate what kind of accuracy and performance one can achieve with an aerial mapping and surveillance system based on a monocular optical sensor. Consequently, existing methods that have not previously been used for tracking of floating objects have been studied in this thesis.

The scope of this research covers everything from detection to georeferencing and target tracking. System integration and description of practical challenges related to field experiments are also within the scope.

1.4 Publications and Contributions

This thesis is based on the following list of articles published in peer-review international journals and conferences:

Journal publications and book chapter:

- H. H. Helgesen, F. S. Leira, T. A. Johansen and T. I. Fossen, Detection and Tracking of Floating Objects using a UAV with Thermal Camera, *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, pages 289-316, 2017. doi: 10.1007/978-3-319-55372-6_14, [43]
- H. H. Helgesen, F. S. Leira, T. I. Fossen and T. A. Johansen, Tracking of Ocean Surface Objects from Unmanned Aerial Vehicles with a Pan/Tilt Unit using a Thermal Camera, *Journal of Intelligent and Robotic Systems*, Volume 91, Issue 3-4, pages 775-793, 2018. doi: 10.1007/s10846-017-0722-3, [44]
- H. H. Helgesen, F. S. Leira, T. H. Bryne, S. M. Albrektsen and T. A. Johansen, Real-time Georeferencing of Thermal Images using Small Fixed-Wing UAVs in Maritime Environments, *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 154, pages 84-97, 2019. doi: 10.1016/j.isprsjprs.2019.05.009, [46]
- H. H. Helgesen, T. H. Bryne, E. Wilthil and T. A. Johansen, Camera-based Tracking of Floating Objects using Fixed-wing UAVs, *IEEE Transactions on Aerospace and Electronic Systems*, 2019 (submitted), [45]
- F. S. Leira, H. H. Helgesen, T. A. Johansen and T. I. Fossen. Object Detection, Recognition and Tracking from UAVs using a Thermal Camera. *Journal of Field Robotics*, 2019 (submitted), [67]
- S. G. Mathisen, F. S. Leira, H. H. Helgesen, K. Gryte and T. A. Johansen, Autonomous Ballistic Airdrop of Objects from a Small Fixed-Wing Unmanned Aerial Vehicle *Autonomous Robots*, 2019 (conditionally accepted), [79]

Conference publications:

- H. H. Helgesen, F. S. Leira, T. A. Johansen and T. I. Fossen, Tracking of Marine Surface Objects from Unmanned Aerial Vehicles with a Pan/Tilt Unit using a Thermal Camera and Optical Flow, *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 107-117, 2016. doi: 10.1109/ICUAS.2016.7502581, [42]
- H. H. Helgesen, F. S. Leira, and T. A. Johansen, Colored-Noise Tracking of Floating Objects using UAVs with Thermal Cameras, *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, [47]

In addition to the publications listed above, I have also contributed to the following publications:

Publications not included in this thesis:

- J. Hosen, H. H. Helgesen, L. Fusini, T. I. Fossen and T. A. Johansen, Vision-Aided Nonlinear Observer for Fixed-Wing Unmanned Aerial Vehicle Navigation, *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 8, pages 1777-1789, 2016. doi: 10.2514/1.G000281, [51]

- Ø. K. Helgesen, E. F. Brekke, H. H. Helgesen and Ø. Engelhardtsen. Sensor Combinations in Heterogeneous Multi-Sensor Fusion for Maritime Target Tracking. *International Conference on Information Fusion (FUSION), 2019*, [48]

Other contributions

The following list describes other contributions that have been made as a part of this project:

- Development of a payload for the NTNU Cruiser-mini fixed-wing unmanned aerial vehicle together with other researchers at the department. The payload was designed specifically for georeferencing and target tracking. It is described in Section 2.3.
- Organized field experiments and gathered experimental data.
- Contributed in development of a software framework in Matlab for detection and tracking with different tracking filters.

1.5 Outline

This thesis is organized in eight different chapters and one appendix. The contents of the chapters are shortly summarized in the following list:

- Chapter 2 presents fundamentals and necessary information about unmanned aerial vehicles, payloads for unmanned aerial vehicles and different sensors that are relevant for this work. Camera calibration and time synchronization are also described.
- Chapter 3 gives an introduction to target tracking. A literature review is presented in Section 3.1. Bayesian tracking of a single target is described in Section 3.2, and data association and multi-target tracking are described in Section 3.3. Finally, Section 3.4 describes track maintenance, and consistency of tracking filters is described in Section 3.5.
- Chapter 4 concerns detection of ground objects using optical sensors. Object detection and related work are discussed initially before methods for detection of floating objects are presented. The final part of the chapter contains two independent case studies investigating detection of floating objects (first case study) and ground objects (second case study).
- Chapter 5 concerns georeferencing of optical images captured with a monocular camera. Georeferencing equations for a camera mounted in a fixed-wing UAV are derived and compensation of mounting misalignments is described. A navigation system based on a nonlinear observer is also presented because

the accuracy of georeferencing strongly depends on precise navigation estimates. It also discusses optical flow and relates that to the velocity of the UAV. The final part of the chapter includes three independent case studies where georeferencing and calculation of target velocity using optical flow are investigated.

- Chapter 6 concerns tracking of floating objects when it is assumed that the pose of the optical sensor is known perfectly. That is obviously not correct for experimental data, but the assumption simplifies the tracking design. Moreover, it works well in practice when considering the accuracy of the target estimates. The tracking system utilizes techniques described in previous chapters and ties it all together. Both linear and nonlinear designs are investigated and compared. Four different case studies are presented in the final part of the chapter where several different topics are studied experimentally.
- Chapter 7 concerns tracking of floating objects in the presence of navigation uncertainty, in contrast to the previous chapter. Potential uncertainty in the pose of the thermal camera is considered and modeled in the tracking system. These solutions extend the architectures presented in Chapter 6. Four case studies end the chapter and analyze these methods experimentally.
- Chapter 8 concludes this research and discusses future work that can or should be conducted within the field of target tracking from UAVs with monocular optical sensors.
- Appendix A describes the sensors mounted in the thermal camera payload. Technical specifications for the thermal camera, on-board computer, gimbal, inertial measurement units and global navigation satellite system receivers are given.

Chapter 2

UAV and Sensor Suite

Unmanned aerial vehicles are used as a remote sensing platform for gathering data in this thesis. Thus, UAVs are merely utilized as a tool for solving other tasks. Nevertheless, it is important to understand limitations, possibilities and challenges that arise when UAVs are used in remote sensing operations. This includes safety, technical and ethical considerations. A thermal camera is the main sensor in this research. Optical sensors and other sensors that are necessary in mapping and surveillance are discussed in this chapter, which includes the following sections:

- Section 2.1 describes UAVs and is meant to give basic knowledge to readers who are unfamiliar with UAV operations.
- Section 2.2 describes optical sensors, which is the main sensor used for situational awareness and remote sensing in this thesis.
- Section 2.3 presents a UAV payload developed during this research and the sensors within the payload.
- Section 2.4 discusses how UAVs can be used as a tool in remote sensing applications and touches upon ethical issues that arise in research concerning UAVs.
- Section 2.5 covers time synchronization and calibration of different sensors. This is essential when observations in the sensor frame are related to other coordinate systems.

2.1 Unmanned Aerial Vehicles

Unmanned aerial vehicles or drones is a term describing aerial vehicles without humans on-board the vehicle. A pilot can still be in control of the vehicle and such an operation is called a remotely piloted aircraft system (RPAS). Therefore, an important distinction is the difference between a fully autonomous operation without human interaction and operations where a pilot or UAV operator is in control.

UAV missions today are typically a combination of the two. A pilot controls the UAV manually during take-off and landing, and the UAV operates autonomously during the main part of the flight.

UAVs can be divided into two main classes. Fixed-wing UAVs are vehicles with wings that produce lift due to airspeed. The NTNU Cruiser-mini fixed-wing UAV is displayed as an example in Figure 2.1. Multicopters are rotor-crafts with hovering capacities such as helicopters and have two or more rotary wings. Multicopters are typically used for short-duration missions (less than an hour) at lower altitudes and driven by electrical propulsion. Fixed-wing UAVs are used for longer missions (duration of several hours) and often operate at greater altitude and speed. Fixed-wing UAVs can both be driven by electrical and petrol propulsion. They are typically heavier and can carry more additional weight than multicopters.

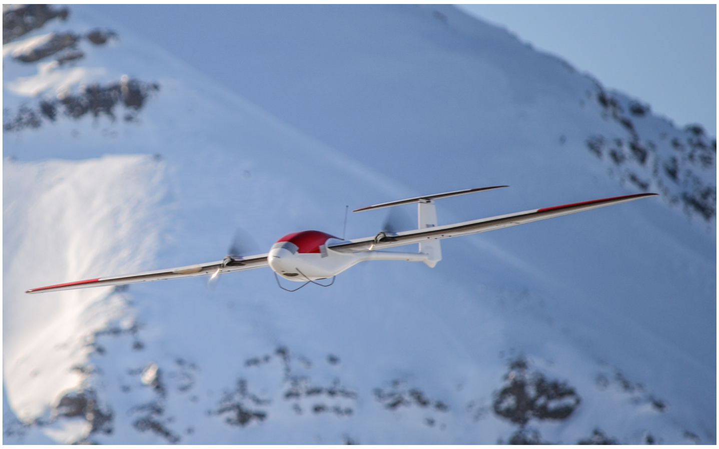


Figure 2.1: The NTNU Cruiser-Mini fixed-wing UAV

The outdoor use of UAVs is governed by the Civil Aviation Authority and regulates how and where UAVs can be used. Airworthiness is the most important consideration in the use of UAVs and describes the ability to maintain safe conditions during flight. Airworthiness is especially vital in operations near people or settlements, and when expensive payloads are used in the operation. Remote sensing operations are typically in geographical areas where the UAV is outside of the visual line-of-sight (VLOS) for the UAV operator. These operations are called beyond-line-of-sight (BLOS) operations and require a special permit. VLOS missions are restricted to 400 feet above ground level (AGL) and obviously within line of sight of the operator. BLOS operations have stronger requirements related to robustness and airworthiness because the operator cannot intervene using manual remote control in emergency situations. UAV missions can also be beyond radio line of sight (BRLOS) and concerns situations where a direct link between the ground station and the UAV is unavailable. This can also be relevant for remote sensing operations where the UAV needs to investigate areas far away from the ground station, for example in the arctic or at sea.

2.2 Optical Sensors

Optical sensors transform light into electronic signals, which are further transformed into a digital representation. Electro-optical (EO) and infrared sensors are examples of passive optical sensors, where the term passive means that the sensor depends on external illumination of the scene (e.g. sunlight or artificial light) or a scene with targets that have their own emission of light [62]. This is opposite to active sensors such as radars which send out a radio signal from the sensor itself and wait for a response. Passive optical sensors are sensitive to external factors such as weather conditions, light conditions, temperature and the characteristics of the scene. Therefore, these factors must be assessed when new missions are planned, and a sensor that fits the requirements of the mission should be chosen.

An object will either reflect, absorb or transmit incoming light. Reflected light is incoming light that is re-emitted from the surface of the object. Light that is absorbed is transformed into heat. Transmitted light is light that goes through transparent objects and is emitted on the opposite side. The radiation of an object depends on the combination of reflected light and light emitted by the object itself. The amount of reflection depends on the surface material, color and temperature. Emitted light depends on the emissivity of an object in combination with the temperature of the object.

The most common passive optical sensor is EO cameras sensitive to the visual spectrum, which ranges from 380 nm to 750 nm. They can be found in many different devices such as computers, smart phones and autonomous vehicles. The wavelengths in the visual spectrum depend mostly on reflected light and the target emissivity is not important for the end result. This is why visual-spectrum cameras require sunlight or artificial light for a scene to be visible. They typically have high resolution and frame rate at a reasonably low cost. In machine vision, visual spectrum cameras are used in everything from object detection to recognition and classification.

Infrared cameras are also passive optical sensors, but sensitive to another part of the electromagnetic spectrum. The infrared spectrum covers wavelengths from the visual spectrum (750 nm) to the far infrared spectrum (15 µm), and is typically divided into the following bands:

- The near infrared (NIR) band covers wave lengths from 0.7 µm to 1.0 µm and is located close to the visual spectrum. Therefore, near-infrared images require illumination that can be reflected and look quite similar to (grayscale) visual spectrum images.
- The short-wave infrared (SWIR) band covers wave lengths from 1 µm to 3 µm. This band is a combination of emitted and reflected radiation. Thus, the reflection properties, temperature and emissivity of the scene are important for the resulting image.
- The mid-wave infrared (MWIR) band covers wave lengths from 3 µm to 5 µm.

This band is also a combination of emitted and reflected radiation. Emitted radiation is more important for objects with higher temperature and large emissivity.

- The long wave infrared (LWIR) band covers wave lengths from 8 µm to 12 µm. Emitted radiation is dominating this part of the infrared band and images captured in the LWIR spectrum are not as similar to visual spectrum images as NIR images. The temperature of objects influences the emitted radiation. Thermal cameras are sensitive to the LWIR part of the electromagnetic spectrum to perceive differences in temperature. An example of a thermal image captured during a flight experiment is shown in Figure 2.2.

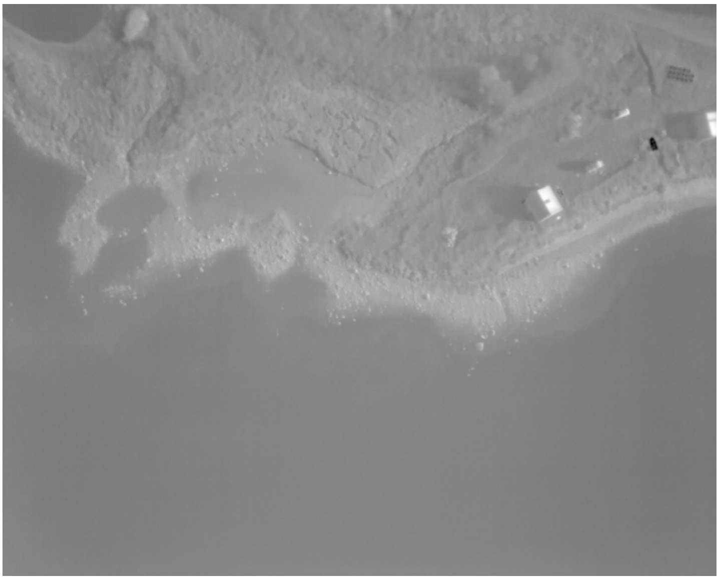


Figure 2.2: Example of image in the long wave infrared spectrum captured with a thermal camera. The image shows areas with sea, land, two buildings and a couple of cars. The emissivity of the ground is stronger than the emissivity of the sea surface and explains why it is possible to observe where the sea surface ends.

Utilizing the LWIR spectrum is beneficial in situations with reduced light and thermal cameras work at night. It is also less sensitive to poor weather conditions and more robust in general. However, the atmosphere absorbs energy throughout the infrared spectrum and can influence the noise level in SWIR images in the same sense as for visual spectrum images. Moreover, conducting recognition and classification in thermal images can be harder because information about color and texture are lost in the LWIR band. Thus, properties such as temperature, shape and size are more prominent in LWIR images.

The sensitivity of an optical sensor decides the smallest amount of change in radiation the sensor is able to perceive. For a thermal camera (LWIR), the sensitivity can be interpreted as the smallest temperature change of an object the camera can detect. The spatial resolution of an optical sensor describes the number of pixels in the image sensor and the physical area each pixel is covering. Temporal resolution describes the rate at which the sensor can acquire measurements, and is described by the frame rate. Thermal cameras typically have lower spatial resolution than EO cameras. Moreover, thermal sensors are more expensive.

2.3 UAV Payload

A number of sensors is needed when UAVs are used in remote sensing operations. It is common to distinguish between sensors that are critical for the airworthiness of the UAV and sensors that are used to gather information about the surroundings for remote sensing. On-board systems vital for the airworthiness are often called avionics and include (see Figure 2.3):

- A navigation system, which include navigation sensors such as inertial measurement units (IMUs), a global navigation satellite system (GNSS), an altimeter and a heading reference (e.g. a magnetic compass). IMUs usually measure specific force and angular rate. GNSS receivers measure position, course and speed over ground. Altimeters measure the altitude and magnetic compasses are used to measure the heading of the UAV.
- An autopilot responsible for giving low-level commands to UAV actuators based on the current and desired state. This includes changes in the course and altitude of the UAV. A guidance system is often used to keep the UAV on the desired path.
- A radio link used to communicate with the UAV from a ground control station.

The ground station is used to communicate with the UAV when it is airborne. It can also be used to change the path of the UAV or give instructions to sensors on-board the UAV. This can for instance be to instruct the camera to turn storage on and off to save memory in long-duration missions. The ground station can also be used to visualize data that are sent from the UAV.

Sensors dedicated to remote sensing are often mounted together in a payload and should not affect the systems related to airworthiness. Typical sensors in remote sensing payloads are

- Optical sensors (visual spectrum or infrared cameras). This also includes hyperspectral and multispectral cameras.
- A gimbal used to direct a camera towards a specific area
- Lidars
- Radars

Common for fixed-wing UAVs is limitations related to power consumption and available weight and space. This is especially relevant for UAVs driven by an electrical engine. UAVs often need to be light weight and have an aerodynamic shape that maximizes the available flight time. Thus, sensors that consume power are directly reducing the duration of a mission. That is why sensors with low power consumption are needed. Moreover, they should ideally also be small to fit within the restricted space available in the UAV. Consequently, it is not desirable to develop a generic payload with many different sensors for remote sensing.

A payload with georeferencing and tracking capabilities has been developed and assembled as a part of this research. A thermal camera is the core component in the payload, but several other sensors are also included in the payload. A tailor-made payload is necessary for several reasons. The thermal camera must be interfaced to capture and store images on-board the UAV. This can be achieved with the thermal camera and an on-board payload computer driven by an external battery. However, for georeferencing and target tracking, it is necessary to know the camera pose (orientation and position) whenever an image is captured. Conventional autopilots are able to estimate the pose of the UAV, but not necessarily with a sufficient update rate or the required precision. Moreover, it is not easy to synchronize the camera and the autopilot without additional sensors. For these reasons, a payload with several sensors are needed. The payload is able to store images, control a gimbal to point the camera in a specific direction and estimate the navigation states of the UAV. In addition, all sensors are synchronized through a custom-made synchronization board. This means that both georeferencing and tracking can be achieved solely by the sensors in the payload. The navigation data from the autopilot are not needed, but are of use to verify and validate the data from the payload.

Thermal camera payload

This section lists the sensors that are included in the thermal camera payload developed as a part of this research. Technical specifications are listed in Appendix A. A diagram showing the sensors and information flow is displayed in Figure 2.3. The payload consists of the following sensors:

- FLIR Tau2 thermal camera used to monitor the sea surface and search for floating objects.
- TEAX ThermalGrabber used to capture the digital image from the thermal camera.
- MicroUAV R-BTC88 retractable pan/tilt gimbal that can be used to point the thermal camera in a specific direction. It is also used to retract the camera during take-off and landing.
- SenTiBoard [1] used to synchronize different sensors in the payload.
- Analog Devices ADIS 16490 IMU measuring linear acceleration and angular rate in three orthogonal axes.

- Sensonor STIM 300 IMU measuring linear acceleration and angular rate in three orthogonal axes.
- 2x Real-time kinematic (RTK) Global Positioning System (GPS) based on uBlox NEO-M8T receivers.
- Odroid XU4 as on-board payload computer for storing sensor data.

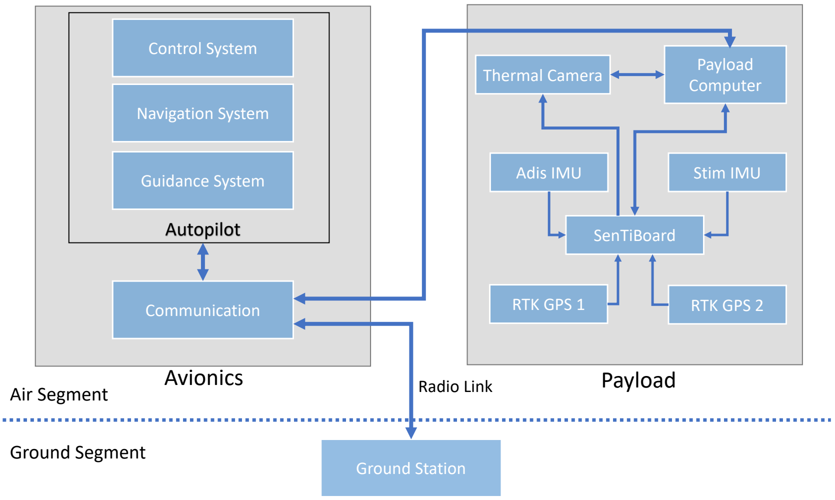
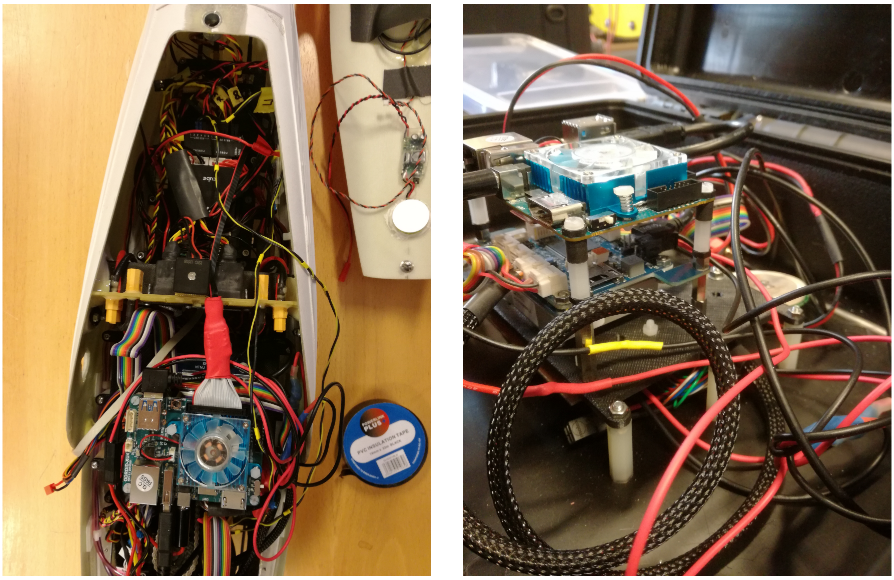


Figure 2.3: Payload illustration. The arrows show the information flow.

The GPS receivers and the IMUs are used to gather independent navigation data as an alternative to the navigation data from the autopilot. Low-cost conventional autopilots use cheap sensors so it is desirable to have an alternative in case of dubious estimates from the autopilot. The thermal camera payload mounted in the NTNU Cruiser-mini is displayed in Figure 2.4.

The payload also consists of wiring and several voltage transformers used to get the correct voltage level for each sensor. An independent payload battery has also been added to the system to prevent a short-circuit or other issues in the payload from affecting the airworthiness of the UAV.



(a) (b)

Figure 2.4: The thermal camera payload mounted in the NTNU Cruiser-mini (a) and in a suitcase (b)

2.4 Remote Sensing

A fixed-wing UAV is a convenient platform for remote sensing as described in Section 1.1. They can carry numerous sensors and cover a relatively large geographical area. Moreover, areas that are dangerous or unaccessible for people can be investigated with UAVs. The number of applications where UAVs are utilized is growing. UAVs are excellent tools in search and rescue operations where missing people need to be located quickly or in disaster areas where it is critical to get an overview of the situation. An example is a landslide that blocks all roads where the only safe access is through the air.

The usefulness of UAVs as a remote sensing platform is not going to be smaller in the future. This raises some interesting questions regarding safety and ethical prospects that should be considered. Safety is imperative in situations where UAVs operate in proximity to wildlife, settlements or people. Maintaining safety requires research on robust navigation, control and fault-tolerant systems. A significant amount of research is directed to achieve safe operations and is driven forward by the strong regulations from national authorities. Missions in proximity to humans are desirable in many applications and could have acceptable risk in the near future.

Some ethical considerations also arise with the use of UAVs in remote sensing applications. This includes protection of privacy for individuals and awareness about the possibility of new technology being used for other purposes than the intention. Privacy issues are important to maintain and should be achievable as long as rules and regulations are respected. This is for example related to photography of private property or of people without their permission. Perhaps the most important consideration is awareness about UAVs being used as a tool to threaten national interests and people. It is important to realize that UAVs can be used as a threat to human interests. This can for example happen if a UAV is hijacked during a mission or if illegal operations are conducted. Nevertheless, UAVs is a significant resource in numerous applications that are useful for the society, especially in emergency situations.

2.5 Sensor Calibration

A key part in mapping and surveillance is to connect a measurement in an image to an Earth-fixed coordinate system. This requires a transformation that depends on the sensor itself and time-synchronized data. This section describes thermal camera calibration and synchronization of different sensors. Calibration of mounting misalignment errors is described in Chapter 5, and is also of great importance in remote sensing operations.

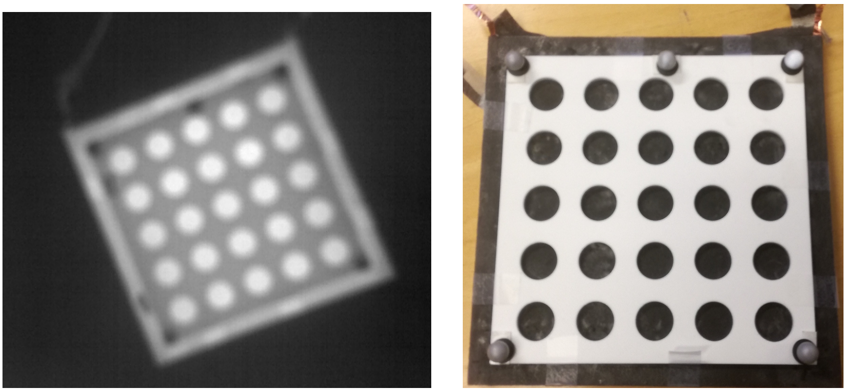
2.5.1 Camera calibration

Mapping pixels in the image frame to an Earth-fixed coordinate system requires precise knowledge about the camera intrinsic matrix, which can be extracted from the camera and lens specification. However, in practice, the true camera model often deviates from the theoretical model and calibration is necessary. Calibration of visual spectrum cameras is addressed extensively in the literature. Most methods are based on [130] and use a chess-board pattern for calibration. The principle is based on capturing several images of the calibration pattern where the camera changes its position and orientation with respect to the calibration surface. Several images are used together with the known properties of the calibration pattern to identify intrinsic and extrinsic camera parameters.

Calibration is of similar importance for thermal cameras. However, because the image sensor is sensitive to a different spectrum than the visual, the standard chess-board calibration surface is not clearly visible in thermal images. Therefore, another calibration surface is needed. Thermal camera calibration has been addressed before [40, 65, 123], but a common calibration surface has not been accepted for thermal cameras. A new surface was proposed in [46] and the rest of this section is based on this article.

Most calibration patterns for thermal cameras use some sort of heat source to generate a visible structure. The pattern proposed in [46] is based on heating a circle grid so that it is visible and can be detected in the LWIR infrared spectrum.

The circle grid is a 3D-printed square plastic plate. The plate is mounted on a wooden surface, which is covered with a conductive (resistive) coating that is heated when electric current flows through. By connecting a power source to the coating, the wooden surface is warmed up quickly. When the circle grid is attached to the wood, the heat escapes through the circles and they seem warmer than the rest of the plastic plate. The main advantage with this surface is that the circle pattern is visible rapidly. The heating time is just about a minute and the power source keeps the wood at a high temperature throughout the duration of the calibration. In addition, the temperature difference between the circles and the rest of the plastic plate is large because the plate conducts (or transfers) heat poorly, giving a high signal to noise ratio. Therefore, the calibration surface is advantageous compared to surfaces that are heated by lamps or cooled down passively because the temperature difference remains large for a longer period. An example of an image captured during calibration is displayed in Figure 2.5 together with a visual-spectrum image of the calibration plate.



(a) (b)

Figure 2.5: A thermal image used for camera calibration (a) and a visual-spectrum image of the calibration plate (b)

The goal of the calibration procedure is to obtain an accurate representation of the intrinsic camera matrix and distortion coefficients. The intrinsic camera matrix is defined as

$$\mathbf{A} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f \cdot m_x & 0 & c_x \\ 0 & f \cdot m_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where f is the focal length of the camera, c_x and c_y are the coordinates of the principal point and expected to be in the middle of the image. m_x and m_y are scale factors relating pixels to meters. Therefore, f_x and f_y are the focal lengths (in pixels) in the horizontal and vertical dimension, respectively. A camera with a pixel resolution of $u \times v$, focal length f and sensor size of $w \times b$ has the following theoretical intrinsic matrix:

$$\mathbf{A} = \begin{bmatrix} \frac{u}{w}f & 0 & \frac{u}{2} \\ 0 & \frac{v}{b}f & \frac{v}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

Due to manufacturing inaccuracies, the true intrinsic matrix normally has other values than the theoretical matrix.

Distortion coefficients are usually identified with camera calibration. Identification of distortion parameters and calculation of undistorted pixels coordinates are described in [65]. It is not discussed further in this thesis because distortion coefficients only affected the experimental results marginally. Therefore, they are considered to be negligible with respect to other error sources discussed later in this thesis.

Calibration results for the FLIR Tau 2 thermal camera

Camera calibration was conducted for the FLIR Tau 2 mounted in the payload described in Section 2.3. The technical specification for the camera is given in Appendix A. The spatial resolution in pixels is 640 × 512, and the focal length is 19 mm. A set of 30 independent images was used to conduct the calibration and the following intrinsic matrix was obtained:

$$\mathbf{A}_{\text{Tau2}} = \begin{bmatrix} 1159.2 & 0 & 313 \\ 0 & 1167.8 & 265 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

The values are close to the theoretical matrix for the camera, which is

$$\mathbf{A}_{\text{Theoretical}} = \begin{bmatrix} 1117.7 & 0 & 320 \\ 0 & 1117.7 & 256 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The principal point of the camera was estimated to be 7 and 9 pixels from the theoretical center in the horizontal and vertical direction, respectively. Moreover, the focal lengths are slightly longer than the length given by the manufacturer, but the difference is small. Calibration was also conducted on a different set of images and gave an intrinsic matrix close to the values here. Thus, the calibration result is assessed to be reliable.

2.5.2 Time synchronization

The UAV pose must be known accurately when an image is captured in mapping applications. Therefore, robust time stamping of the capture time and a navigation system with sufficiently high update rate are needed. Furthermore, accurate synchronization between the time frames of the camera and the rest of the sensors in the payload is required. Timing errors in the magnitude of fractions of a second are troublesome in georeferencing (see Chapter 5), and especially vital during fast maneuvers where the attitude is changing quickly. Synchronization is achieved with SenTiBoard [1]. The precision of the time stamping using SenTiBoard is tested to be 1 µs for navigation sensors, such as IMUs and GNSS receivers.

SenTiBoard has two main tasks in the payload. It is used to synchronize the navigation sensors, namely two IMUs and two RTK-GPS receivers. In addition, it is used to synchronize the images with the navigation sensors. A simple alternative is to use the on-board computer time for when a sensor measurement is received, but this solution is not accurate since there may be a significant bias between the on-board computer time and the GPS time. Moreover, the on-board computer clock may drift during the duration of the experiment, and can be busy with other tasks when measurements are received. Thus, a varying delay or latency is added if the on-board computer time is used in a naive fashion.

The internal frame rate of the FLIR Tau2 is 30 Hz, but the output rate of images is reduced to 7.5 Hz because of export regulations. External triggering of image capture is not possible. Therefore, to obtain the capture time of images, SenTiBoard synchronizes the internal camera clock with the GPS time every second and thereby the camera and the navigation sensors. A potential drift in the camera clock is removed with this approach. However, this solution neglects the delay between exposure of the pixels and when the image is ready on the camera output bus, but must be accepted when external triggering is unavailable. Nevertheless, the delay is only minor because the internal time between consecutive images is 33.3 ms in the camera core, so any potential camera latency is significantly smaller than that.

Chapter 3

Introduction to Target Tracking

Target tracking is a popular research field and important in many industries such as for autonomous cars and vessels, surveillance and monitoring of ships, aircraft and icebergs. The literature describing tracking is too large to survey in detail. Different strategies exist and the choice of tracking sensor and targets influence if a particular tracking strategy is suitable or not. This chapter seeks to introduce the reader to the field of target tracking. A few fundamental concepts are described in Section 3.1 together with a general literature review on the field of target tracking. Tracking of floating objects using UAVs and optical sensors is described in Chapter 6. Thus, this chapter has a broader perspective and works as a basis for the contents of Chapters 6 and 7. The following topics are covered in this chapter:

- Section 3.1 gives an introduction to the field of target tracking, recent trends and related literature.
- Section 3.2 gives a fundamental definition of the tracking problem in the Bayesian context and relates this to the Kalman filter.
- Section 3.3 concerns data association and explains how new measurements can be connected to existing tracks in the tracking filter.
- Section 3.4 describes track maintenance and topics such as track initialization and termination of existing tracks.
- Section 3.5 describes consistency of tracking filters, and is especially important for multi target tracking (MTT) and reliable covariance estimation.

3.1 Background

Locating an object in one image is most often not sufficient for situational awareness. It is necessary to monitor targets when they are within the surveillance region and predict their motion whenever they are not observed by the tracking sensor. This task falls within the field of target tracking where the goal is to maintain state estimates of one or multiple objects. Targets can be anything from people to cars, ships, airplanes, forest fires and icebergs. The type of target is not necessarily important for the choice of tracking strategy. However, the choice of tracking sensor obviously depends strongly on the target(s) of interest, and the sensor characteristics can affect the choice of tracking strategy.

Target tracking has been studied for a long time and the complexity of the algorithms has increased with the computational capacity of computers. A significant breakthrough was reached when the Kalman filter (KF) was introduced in the 1960s [59]. The Kalman filter provided the opportunity of estimating the states of different targets in a simple way, but did not solve the data association problem in MTT. Moreover, the optimality of the Kalman filter is restricted to certain assumptions. Therefore, later development led to other filtering strategies for nonlinear and non-Gaussian tracking problems. This includes variations of the Kalman filter, such as the extended Kalman filter (EKF) [117] and the unscented Kalman filter (UKF) [115]. Popular methods for target tracking today are the particle filter [33] and multi hypotheses tracking [94]. More recently, random finite set statistics have also been used in target tracking to model target appearance and disappearance from the surveillance region using finite set theory [71, 96].

How targets are modeled is important for the fundamental assumptions in tracking. The most common approach is to model targets as point masses. This means that it is assumed that a single target can only produce a single measurement in each sensor scan. Moreover, it means that a measurement originates from a single target and cannot stem from multiple targets. This is sensible in situations where the size of the target is small in the sensor frame (large distance between sensor and target compared to size of target). Large targets and high sensor resolution (such as for optical sensors) can violate this assumption. Extended target tracking covers these situations and means that a target is allowed to produce several measurements in a single sensor scan [34]. Extended target tracking opens the possibility of tracking the contour and size of a target. Only target-tracking in the form of point masses is considered in this thesis.

Perhaps the greatest challenge in target tracking is the measurement origin uncertainty. In MTT, several measurements can be present in the same sensor scan and it is necessary to relate these measurements to the existing set of tracks. Furthermore, some measurements may be originated from clutter or false reports meaning that they do not originate from any new or existing target. Solving this problem is called data association or measurement association, and is necessary in MTT or in situations where a single target is tracked in clutter [58]. This is obviously more challenging than tracking a single or multiple targets without measurement origin uncertainty. Erroneous data association leads to increased uncertainty in the state estimates, which eventually lead to track loss, track divergence or track swaps (two tracks are interchanged).

Several different alternatives for data association exist. It is common to distinguish between what is called firm and soft association. Firm data association means that each measurement is discarded or associated to a single track. The track can be a new track or an existing track. In certain situations, two different measurements might be equally likely to originate from a single target. Assigning only one measurement to a target can be challenging and lead to erroneous association. Soft data association handles this issue by allowing multiple measurements to be associated to one track and weighting each measurement with the likelihood of that measurement belonging to the track (or another probability measure).

The simplest form of firm association is to use the nearest neighbour approach where the closest measurement in the validation gate is assigned to a target. This must be solved globally in a situation with multiple targets to minimize the total distance [61]. Another possibility is to use the strongest neighbour where the measurement with the highest intensity in the validation gate is chosen [3]. This is only applicable to tracking sensors that provide some sort of intensity measure (radar for instance). Soft data association can be handled with the probabilistic data association (PDA) filter for a single target in clutter. In the PDA filter, a weighted average of all measurements in the validation region is created, based on the likelihood of a measurement originating from the target [3]. This has also been extended to cover multiple targets in clutter with the joint probabilistic data association (JPDA) filter [4]. The PDA filter has also been extended to handle maneuvering targets by combining it with the interacting multiple model (IMM) approach. The resulting algorithm is called IMMPDAF [60].

In several tracking systems, the concepts of target existence and visibility are used to simplify track maintenance. The traditional approaches augment the state vector with binary existence and visibility variables. Augmenting the PDA filter to include these variables led to the development of the integrated probabilistic data association (IPDA) filter in the single target case [83]. This was naturally extended to handle multiple targets and called joint IPDA (JIPDA) [82]. Target existence is also included in tracking strategies based on random finite set statistics where a track is modeled as a set that can be both empty and a single entity. Algorithms based on random finite sets are for example the multi-Bernoulli filter [71] and probability hypothesis density filter [72].

3.2 Bayesian Tracking of Single Target

This section discusses tracking of a single target in the Bayesian context and relate that to the Kalman filter. It is assumed that the goal is to estimate the states of a single target which is modeled as a point mass. Moreover, a tracking sensor is available and provides insight into the target states. The following assumptions are made:

- A single target exists in the surveillance region so data association is neglected.

- A target only produce a single measurement.
- A measurement only originates from one target or clutter (false alarm).

The goal in target tracking is to recursively use measurements to calculate the states of the target as accurately as possible using all available information. This includes utilizing the entire measurement history, that is the current measurement(s) and all of the previous measurements. Let the state and measurement at time k be denoted $\mathbf{x}_k$ and $\mathbf{z}_k$, respectively. The set of the current and previous measurements at time k is defined as $\mathrm{Z}^k := \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$. In Bayesian target tracking, the objective is to construct the posterior probability density function (PDF) of the state at each time step. The posterior PDF at time k is $p(\mathbf{x}_k|\mathrm{Z}^k)$. The prior probability density function at time k is denoted $p(\mathbf{x}_k|\mathrm{Z}^{k-1})$.

Lets assume that an initial PDF of the target states is available at time zero and denoted $p(\mathbf{x}_0|\mathrm{Z}^0)$. The posterior PDF for the next time step is obtained through a prediction and a correction. The prior PDF is based on a model describing the expected transition from the posterior PDF at the previous time step. This transition is denoted $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the prediction for the current state is defined as

$$p(\mathbf{x}_k|\mathrm{Z}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathrm{Z}^{k-1})d\mathbf{x}_{k-1} \tag{3.1}$$

When a new measurement is available, it is used to modify the prior PDF and create a resulting posterior PDF. The measurement model is denoted $p(\mathbf{z}_k|\mathbf{x}_k)$ and the use of Bayes rule gives the following posterior PDF

$$p(\mathbf{x}_k|\mathrm{Z}^k) = \frac{p(\mathrm{Z}^k|\mathbf{x}_k)p(\mathbf{x}_k|\mathrm{Z}^{k-1})}{p(\mathrm{Z}^k|\mathrm{Z}^{k-1})} \tag{3.2}$$

where the denominator is the normalization constant.

The posterior distribution in (3.2) is a general expression that is valid as long as the current measurement is assumed to depend only on the current state and not the previous measurements. It is generally assumed that the state $\mathbf{x}$ satisfies the Markov property at each time step. This means that the current state can be deduced from the previous state alone and that the necessary information is preserved in the previous state without needing knowledge regarding older states.

The goal in tracking is to find an estimate $\hat{\mathbf{x}}$ that represents the posterior distribution. This problem can be simplified under certain assumptions. Assume that the state transition model and the measurement model can be written as

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{w}_k \tag{3.3a}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{3.3b}$$

where the functions $\mathbf{f}$ and $\mathbf{h}$ can be nonlinear or linear, and $\mathbf{w}$ and $\mathbf{v}$ are additive process and measurement noise, respectively. Both noise processes are assumed to be Gaussian distributed with zero-mean. The covariance of the process and measurement noise processes are denoted $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. The state transition model is also referred to as the motion model of the target and is used to predict the next prior state. This leads to the Kalman filter and its variations presented in the next sections.

3.2.1 The Kalman filter (KF)

If (3.3a) and (3.3b) are linear, the optimal solution to the tracking problem is given by the KF [59]. The state transition model and the measurement model are expressed as

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{w}_k \tag{3.4a}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{3.4b}$$

The estimate of the state $\mathbf{x}$ is denoted $\hat{\mathbf{x}}$ and the estimate of the covariance of the state estimation error is denoted $\mathbf{P}$. The subscripts in $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ indicate that the estimates are calculated at time k given the measurement history until time $k-1$, and are the prior estimates. Moreover, $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ mean estimates at time k given the measurement history until time k, and are the posterior estimates. The equations for the Kalman filter are:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} \tag{3.5a}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^\top + \mathbf{Q}_k \tag{3.5b}$$

$$\hat{\mathbf{z}}_k = \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \tag{3.5c}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k \tag{3.5d}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1} \tag{3.5e}$$

$$\boldsymbol{\nu}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \tag{3.5f}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\boldsymbol{\nu}_k \tag{3.5g}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\top + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\top \tag{3.5h}$$

The prediction step is given by (3.5a)-(3.5b) and the measurement update step by (3.5c)-(3.5h). The Joseph form is used for the covariance measurement update in (3.5h) because it is valid for any gain. It can be simplified for a linear system when the Kalman gain (3.5e) is used.

One beneficial property of the KF from a remote sensing perspective is the ability to predict the states of the target in periods where measurements are unavailable. In these situations, the posterior estimates are simply equal to the prior estimates because the prior represents the best guess of the current state with $\mathbf{K}_k = \mathbf{0}$. Therefore, it is important to choose a motion model which reflects the expected behavior of the target. It is obviously impossible to find a general model that represents multiple scenarios so the target of interest influences this choice.

3.2.2 The extended Kalman filter (EKF)

The KF assumes that the motion and measurement models are linear. This is key since Gaussian random variables are propagated through the system equations. In practice, sensors used for tracking can seldom be modeled as a linear process. Moreover, target maneuvers are often represented by nonlinear models. So what happens if the measurement or motion models are nonlinear? One solution is the EKF, which uses a Taylor series expansion to linearize the nonlinear functions. Lets assume that both the motion model $\mathbf{f}(\mathbf{x}_{k-1})$ and the measurement model $\mathbf{h}(\mathbf{x}_k)$, in (3.3a) and (3.3b), are nonlinear. These functions can be linearized through the Jacobian and evaluated at the current best estimate in the following manner:

$$\mathbf{F}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}} \tag{3.6a}$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}} \tag{3.6b}$$

By adding linearization of these functions at each time step in the KF, the states can be estimated by the following equations that define the EKF:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}) \tag{3.7a}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k \tag{3.7b}$$

$$\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}) \tag{3.7c}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \tag{3.7d}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1} \tag{3.7e}$$

$$\boldsymbol{\nu}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \tag{3.7f}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \boldsymbol{\nu}_k \tag{3.7g}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top \tag{3.7h}$$

The linearization shown in (3.6) is the first-order linearization and the resulting filter is called the first-order EKF. Higher-order versions of the EKF are designed by adding more terms from the Taylor-series expansion when finding the Jacobians. The drawback with the EKF is that it is a suboptimal approach. The optimality of the KF is lost when nonlinear transformations are added to the tracking problem. Nevertheless, the EKF tends to work well in practical applications as long as the filter is properly initialized. Since the nonlinear transformations are linearized about the state estimates, it is vital that they are close to the true states to accurately represent the nonlinear transformations locally. Poor initial accuracy is a common reason for divergence in the EKF.

3.2.3 Other filtering options

More advanced state estimators exist as mentioned in Section 3.1. The unscented transform has been used in the UKF [115] as an alternative to EKF in situations where the state probability density is non-Gaussian. This can e.g. be when the distribution is skewed or multi-modal. The idea is to sample the probability distribution at deterministic sampling points instead of linearizing the nonlinear transformations. The computational complexity of the UKF is comparable to that of the KF and EKF.

A more computationally heavy approach is particle filters [33]. It is a general approach that can handle nonlinear and non-Gaussian distributions. A sampling approach is used to approximate the PDF of the state. The state is represented as a weighted average of all the particles, and particles that represent a more probable state are weighted more than unlikely particles.

Tracking maneuvering targets is challenging because it is impossible to predict their motion based on a single model. Therefore, a strategy that use multiple models in the estimation problem was developed and called IMM [9]. The idea is to estimate the states under several model hypotheses with the assumption that the correct model is in the set of all hypotheses. Moreover, transition probabilities for switching between two models must be defined for all models. The models are mixed and weighted based on the likelihood that each model produced the current measurement. The total state estimate is the weighted average of all individual model estimates.

3.3 Multiple Target Tracking & Data Association

Multiple target tracking handles the case when several targets are located in the surveillance region at the same time. A common assumption is that targets operate independently so that the motion of one target does not affect the motion of other targets. This is reasonable in most situations unless the distance between the targets is very small. The first thing to consider in MTT is how to estimate the states simultaneously. The easiest way, and a common choice, is to keep one tracking filter for each target. Thus, a bank of tracking filters is created when multiple targets are present.

The biggest challenge in MTT, or single target tracking in severe clutter, is data association as described in Section 3.1. Only firm data association is considered in this thesis. This means that each measurement is associated to a single or zero targets at each time step, and one target can only be associated with one measurement. The reasoning behind this choice will be clear later, but has to do with nature of how measurements are extracted and the amount of clutter and false reports.

It is assumed that measurements are point masses and not distributed over multiple cells (pixels). This is obviously violated in this research since the tracking sensor is a thermal camera. Nevertheless, only the center of each target is used as a measurement so it is still a valid simplification. If it is assumed that the true measurement conditioned on the entire measurement history is normally distributed so that [3]

$$p(\mathbf{z}_k|\mathrm{Z}^{k-1}) = \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}, \mathbf{S}_k) \tag{3.8}$$

, then the true measurement is located in the region given as:

$$\boldsymbol{\nu}_k(\gamma) = \{\mathbf{z}_k : (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^\top \mathbf{S}_k^{-1}(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \leq \gamma\} \tag{3.9}$$

with gate probability given by the threshold γ. This region is referred to as the validation region. Note that the measurement innovation $\boldsymbol{\nu}$ from (3.5f) is used in the validation region. The squared expression in (3.9) is also called the normalized innovation squared (NIS). It is chi-square distributed where the number of degrees of freedom is given directly by the dimension of the measurement vector. The validation region is typically used to decide if a measurement can originate from a specific target or not. The gate threshold γ is chosen so that all measurements belonging to a target fall into the validation region with a selected probability (e.g. with 95%-99% probability). In MTT, each target has its own validation region. Measurements that fall outside of all validation regions are normally interpreted as clutter or a new target.

3.3.1 Global nearest neighbour data association

Data association is not the main topic in this thesis, but one firm data association procedure is described as an example. That is the global nearest neighbour (GNN) data association method. In a scenario with n measurements and m existing tracks, a distance matrix $\mathbf{D}$ can be created [67]:

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,m} \\ d_{2,1} & d_{2,2} & \dots & d_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \dots & d_{n,m} \end{bmatrix} \tag{3.10}$$

The distances $d_{i,j}$ are calculated with (3.9). If the distance $d_{i,j}$ exceeds the threshold γ, it is an unlikely measurement association and the distance is adjusted to infinity. If a column in $\mathbf{D}$ only has elements equal to infinity, a measurement for the corresponding track is likely not present. In the same manner, if a row only has elements equal to infinity, then the corresponding measurement does not originate from existing tracks and must be clutter or a new target. These rows and columns are removed from the distance matrix before the rest of the association procedure is finished.

The GNN approach minimizes the global total distance based on the remaining elements in $\mathbf{D}$ under the requirement that each remaining measurement should be assigned to one target (and one target only). The traditional Kuhn-Munkres algorithm [10] can be used to solve this problem, but the Jonker-Volgenant-Castanon algorithm is much faster in a real-time implementation [74].

An advantage optical sensors have in MTT, is visual features that can be used to help the association procedure. Properties such as size, intensity and contour can be used to distinguish targets that are closely located in the validation region. Moreover, if the estimation accuracy is poor, visual features can be used to decide if a measurement belongs to a target or not if it is located outside of the validation region. This is especially advantageous when firm association is used because it decreases the chance of interchanging measurements belonging to different targets. One approach that utilizes visual features is shown in [67] where a weighted tunable sum of the normalized innovation squared is combined with a distance measure for visual features. Another possibility is to include visual features in the state vector of the target so that the validation region is based on both the physical distance and feature distance. This is not investigated further in this thesis, but is a natural and intuitive extension of the method in [67].

3.4 Track Maintenance

Track maintenance is a broad term in target tracking. It is used as a generic term for handling different parts of the life cycle for a track. A critical question is to answer when a new track should be initiated. Should every measurement not associated to an existing track be interpreted as a new target or is this a poor idea? These issues are handled in track initialization. Another important question is what to do when a target is not observed for a significant amount of time. In other words, when is it acceptable to terminate a track? This is handled in track termination.

3.4.1 Track initialization

Creating a new track whenever an unassociated measurement is received is a poor idea when severe clutter or false alarms are expected. This will lead to several false tracks that the tracking system must maintain. Therefore, it is not common to do this unless the tracking sensor has characteristics that show that false alarms are rare. A more common approach is to use a logic-based strategy that uses a gating approach, and require a certain number of detections in the gate before a track is created. One convenient approach is the one sketched in [3] with the following initialization procedure:

1. Unassociated detections are stored as tentative tracks with a validation region based on the expected measurement noise and the largest expected target motion.

2. If a measurement is present in the region at the next sensor scan, the track is stored as a preliminary track. If not, the tentative track is deleted and the previous measurement interpreted as clutter. If multiple measurements are present in the validation region, additional logic is required such as the GNN approach.
3. A Kalman filter is created for each preliminary track, and the validation region is calculated for each forthcoming time step.
4. If the validation region contains at least one detection in m of the following n sensor scans (n > m), the track is confirmed. Otherwise, the preliminary track is terminated.

This logic is normally referred to as m out of n logic and is a simple way to accept tracks. The magnitude of m and n is something that must be selected and depends on the sensor characteristics. A common choice is for example $m = 3$ and $n = 5$, which means that the validation region must contain at least one measurement for three of the forthcoming five sensor scans to be accepted after the initial two. The choice is a trade-off between minimizing the number of false tracks without rejecting tracks that are based on real targets.

3.4.2 Track termination

Track termination is something that must be considered whenever a target is not observed in several sensor scans. However, how to choose this logic is not straightforward and depends on the sensor characteristics and the application. For example, when tracking objects with UAVs, a significant amount of time between detections occur because targets are outside the field of view of the sensor for extended periods.

After a long period without measurements, the covariance of the state estimates grows more and more because the estimates are based on prediction. Consequently, the validation region increases and might grow to a level where many measurements fall into the region. Depending on the measurement association strategy, this can lead to track divergence. Therefore, it is necessary to decide if a track should be terminated or not. One possibility is to terminate a track whenever the number of measurements in the validation region is above a certain threshold. The reasoning behind this strategy is that several targets are seldom expected to be close to each other, and that many targets in the validation region indicate that the region is too large. Moreover, such a situation might prevent the track from converging towards the true track again.

Track termination, when target tracking is conducted from a UAV, leads to some important considerations. First of all, the surveillance region might be fixed, but the optical sensor is not going to cover the entire surveillance region with a single image. In fact, missions where a significant amount of time is needed to cover the entire region are common. Therefore, only using the size of the validation region or the time since the previous measurement might not be the best choice. One interesting possibility is to use image features to associate tracks with measurements in situations where the validation region is too large. Furthermore, if the filter is too optimistic (meaning that the validation region is too small), visual features can be used to investigate the origin of unknown measurements.

Another possibility is to terminate tracks, but store their last predicted state together with visual properties. If the number of tracks is limited, it is possible to compare the appearance of new measurements, both with active and terminated tracks. With this approach, a terminated track can be reinitialized instead of creating a new track for the same target. This is beneficial in applications where it is critical to uniquely identify each target and not have several track histories for the same target.

3.5 Consistency of Tracking Filters

As mentioned previously, the KF is the optimal estimator for linear systems where the process and measurement noise are zero-mean and white (sequentially uncorrelated) with a Gaussian distribution. Such ideal conditions cannot be expected in practice with the use of real sensors. Targets are not going to move exactly like expected, and typical measurement noise includes a bias and other systematic errors. Because of these issues, a tracking system cannot simply be evaluated just based on the root-mean-squared (RMS) errors for the state estimates. In data association and MTT, it is vital that the system is tuned properly so that the covariance estimates represent the true estimation error correctly. It is obviously also important for filter performance since the gain is based on the estimated covariance. These issues are investigated through consistency considerations. A state estimator is said to be consistent if the following criteria are fulfilled [4]:

1. The state errors should be acceptable as zero mean and have magnitude commensurate with the state covariance as calculated by the filter.
2. The innovations should also have the same property.
3. The innovations should be acceptable as white.

The innovation was defined in Section 3.2 as the difference between the true measurement and the predicted measurement in the Kalman filter. Normally, only the second and third criteria can be verified for experimental data because the true target state is unknown.

If the true state is known, the *first criterion* is verified with the normalized estimation error squared (NEES), which is defined as

$$\epsilon_k = \tilde{\mathbf{x}}_{k|k}^{\top} \mathbf{P}_{k|k}^{-1} \tilde{\mathbf{x}}_{k|k} \tag{3.11}$$

where $\tilde{\mathbf{x}}_{k|k} = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}$ is the a posteriori estimation error and $\mathbf{P}_{k|k}$ is the a posteriori covariance matrix. The NEES should be chi-square distributed with n_x degrees of freedom (DOF), where n_x is the dimension of the state vector, to comply with the first criterion. The *second criterion* is verified with the normalized innovation squared, which was defined in Section 3.3.

$$\epsilon_{\nu,k} = \boldsymbol{\nu}_k^\top \mathbf{S}_k^{-1} \boldsymbol{\nu}_k \tag{3.12}$$

where $\boldsymbol{\nu}$ is the innovation and $\mathbf{S}$ is the innovation covariance matrix calculated in the Kalman filter. ϵ_ν should be chi-square distributed with n_z DOFs, where n_z denotes the number of elements in the measurement vector, to comply with the second criterion. The *third criterion* is checked through autocorrelation to investigate if subsequent innovations are uncorrelated.

For experimental data, these tests have a high variability because only a single realization of the process is available. Lower variability tests are obtained in simulation environments because independent Monte-Carlo simulations can be conducted. Low variability tests for criteria two and three using experimental data from a single test are described in [4]. Care must be taken if the first criterion is investigated for experimental data because the true state cannot be known perfectly. Reliable velocity information is for example hard to acquire when real targets are tracked, but a position reference can in many cases be quite reliable through GNSS measurements.

Example 3.1 This example shows the details of how criteria 1 and 2 can be verified in a particular scenario. Assume that the north and east positions and velocities of a target are estimated. Moreover, assume that north and east positions are measured by the tracking sensor.

The dimension of the state vector is four meaning that $n_x = 4$ and the number of elements in the measurement vector is two: $n_z = 2$. Therefore, the NEES is chi-square distributed with 4 DOFs and the NIS is chi-square distributed with 2 DOFs. The first criterion is verified if the NEES is within the interval:

$$\boldsymbol{\nu}_k \in [r_1, r_2] \tag{3.13}$$

where r_1 and r_2 are the limits of the acceptance region. The two-sided 95% confidence interval for the chi-square distribution with 4 DOFs is $[0.48, 11.14]$. The same test is carried out for the NIS to verify the second criterion. However, since the innovation vector has two elements, the confidence interval is given by the limits for 2 DOFs, and the confidence interval becomes $[0.051, 7.38]$.

If several Monte-Carlo simulations are conducted, the limits of the confidence interval shrink and decreases the variability in the test. The confidence interval for the NIS becomes $[1.5, 2.6]$ if 50 Monte-Carlo simulations are conducted as stated in [4]. This is not described further because the rest of this thesis mostly deals with experimental data.

Chapter 4

Detection of Ground Objects using Optical Sensors

Detection of targets in the sensor frame is critical for measurement corrections in the tracking filters described in Chapter 3. Object detection in images captured by optical sensors is a research field that has evolved significantly in the last decade. Simple algorithms based on fundamental image-processing techniques, such as edge detection and thresholding, dominated for a long time. Today, state-of-the-art detectors use machine learning and neural networks for object detection. This chapter concerns detection of ground and floating objects in images captured with UAVs. Real-time performance on small embedded computers is key and the main focus. This chapter is based on the work described in [42, 43, 79], and the following topics are covered:

- Section 4.1 covers recent advancements in object detection and related work. It is an introduction to the vast amount of literature that exist for object detection in images.
- Section 4.2 covers feature detection in thermal images.
- Section 4.3 covers a specific method for detection of floating objects in thermal images using edge detection.
- Section 4.4 presents two individual case studies for object detection. The first case study concerns detection of floating objects in thermal images using the technique in Section 4.3. The second case study concerns detection of a stationary tarpaulin in visual spectrum images for precision drop [79].

4.1 Background & Related Work

Optical sensors are useful in a vast number of applications and industries, including the maritime sector, car industry, surveillance, monitoring and obviously also in many other applications. Images contain an enormous amount of information and that may be why machine and robotic vision are attractive research fields. Object detection is the task of identifying objects of interest within an image. One simple way to visualize object detection is to decide if each pixel in an image belongs to an object or a non-object region. This is also referred to as finding the foreground (areas with an object) and background in an image. This type of image segmentation results in a binary image where every pixel containing an object is white and the background is black.

Detecting objects in images captured from a moving platform is different than detecting objects in images captured at rest, even though the goal is equal. Moreover, detecting objects in a maritime environment is not similar to identify pedestrians at an intersection or cars on a road. Developing one algorithm for object detection that is able to detect many different objects is not realistic even though neural networks have pushed the field forward. Consequently, it is necessary to find viable algorithms for specific applications or scenarios, especially if a real-time solution on a small embedded computer is required. This also includes choosing the type of optical sensor based on the conditions that are expected. Visual spectrum cameras are advantageous in some situations while sensors sensitive to another spectrum might perform better in other cases. This work primarily uses a thermal camera. A comparison between thermal and visual spectrum cameras is given in Table 4.1.

When looking for floating objects at sea, small structures and vessels are expected to be the majority of the objects. It may be challenging to distinguish small objects from the sea surface and detect dark vessels with a visual spectrum camera, particularly during poor illumination. That is why sensors capturing images at another spectrum are attractive. Vessels at sea have heat sources and materials that are warmer or with higher emissivity than the sea surface. Therefore, using thermal cameras for object detection at sea is a viable alternative and is utilized later in this chapter.

Thermal cameras	**Visual spectrum cameras**
Low resolution	High resolution
Lower frame rate	High frame rate
Expensive	Cheap in comparison
Works at night	Needs light or artificial illumination
More robust in poor weather conditions	Not robust in poor weather conditions
Can observe temperature differences	Can observe color differences
Less texture making segmentation easier in some cases	More texture. Better detail level, but sometimes harder to do segmentation
Not as good for classification and recognition because color information are unavailable	Good for classification and recognition.

Table 4.1: Short comparison between thermal and visual spectrum cameras

Detection of floating structures in images captured from a UAV operating at high speed is challenging because the velocity of the camera typically exceeds the velocity of the objects. Therefore, traditional motion-based segmentation techniques are not appropriate unless one can compensate for the camera motion accurately. This requires knowledge of the navigation states of the UAV and time-synchronized data with high precision. Experience has shown that it is difficult to use motion-based detection techniques in images from an agile platform. Thus, it is necessary to identify algorithms that work well within this restriction.

4.1.1 Related literature

Object detection and tracking are studied thoroughly in the literature and a great number of surveys exist [53, 63, 125, 126]. Covering all of the relevant research in one chapter is impossible. Therefore, this section starts broadly and narrows the scope to detection of floating objects using UAVs.

Image segmentation and feature detection are classical methods in image processing and object detection. A study on segmentation techniques is given in [129] and also discusses edge detection. A survey on advancements in feature detection is described in [68]. [87] describes the problem of object tracking in video surveillance and discusses object representation, descriptors that can be used for feature matching and motion segmentation. Feature-based techniques are used in many computer-vision applications and include popular point detectors such as SIFT [70], SURF [6] and the KLT detector [103]. Point detectors are usually computationally effective and quite robust since they may be scale and rotation invariant. However, they need to be combined with another detection method because features may be located in places where objects are not present. Moreover, several features can be present on the same object. Another issue is that it is easier to associate features between two subsequent images than in a large sequence of images. Therefore, using single features obtained with e.g. SIFT, is not very robust for tracking purposes where you want to track objects over longer periods of time. This is obviously also a consequence of point detectors ability to find many features because association is harder to do on a large set of features. The large set is, on the other hand, very beneficial when calculating e.g. optical flow [44].

Deep-learning based object detection has dominated the literature in recent years, due to impressive detection accuracy and performance on public datasets. A survey on deep neural network architectures is presented in [69]. [38] focuses on object detection, and describes salient-object detection and category-specific detection. It also describes convolutional neural networks and models that have succeeded in detection competitions. The major bottlenecks with convolutional neural networks are the high computational complexity and the amount of training data that is needed. [109] discusses efficient processing of neural networks and how that can be achieved without restricting the accuracy. Effective processing of images through neural networks is key if deep learning is going to be applied on-board UAVs in real-time.

A survey on object detection in optical remote sensing images from an aerial perspective is presented in [14] and focuses on generic object classes such as roads, buildings, ships and several other classes. Technologies for automatic forest fire detection and monitoring using UAVs are presented in [128]. This article discusses fusion of visual spectrum and infrared images which is an interesting possibility. Object recognition in aerial images captured by UAVs using convolutional neural networks is discussed in [92]. [97] discusses classification of thermal images captured from a UAV using deep learning in a search and rescue context. Multi-scale detection in remote sensing imagery, which also includes marine vessels, is described in [21].

A template matching approach for visual detection and tracking with UAVs is presented in [91]. However, in order to be robust, a great number of templates need to be available because template matching is neither scale nor rotation invariant. Consequently, the objects of interest need to be known beforehand and limits the usefulness in many situations. Color segmentation has also been applied in object detection with UAVs [110], but requires tailored parameters towards a specific type of objects. Thus, it is not as robust since floating objects at sea can have many different appearances. A correlation method for ship detection with a visual sensor mounted in a UAV is described in [57]. [64] demonstrates a step-wise solution for detecting marine vessels in thermal images, which works well as long as the temperature difference between the sea surface and objects of interest is sufficient. The method described in Section 4.3 is based on this work.

4.2 Feature Detection and Optical Flow

Feature detectors use different techniques to find keypoints. That can be edges, corners, blobs, lines or other structures that generally appear in images. They are not designed to find specific objects, but parts of objects that later can be merged to a contour. Feature detectors are not computationally expensive compared to neural networks. In fact, feature detectors can often be used at a low-level stage of neural networks. High-performance feature detectors, such as SIFT [70] and SURF [6], are scale and rotation-invariant which is important for aerial images since the scale and rotation of objects change rapidly with the attitude and altitude of the UAV. The main challenge with feature detectors, in the context of floating object detection, is how to relate features to objects. One object usually consists of several features and they need to be associated together. An advantage with feature detectors is the fact that only the current image is used to find features. Thus, a change in background, which must be expected to occur in images captured from a UAV, will not necessarily affect the ability to find features. This is not the case for methods relying on some sort of background subtraction/modeling.

A familiar concept in detection of moving objects is optical flow [8]. Optical flow can be defined as a velocity field that transforms one image into the next image in a sequence of images. A single optical flow vector can be understood as the two-dimensional displacement (in the image plane) of a feature detected in two consecutive images. Feature detectors can actually be used to calculate the displacement of features from one image to the next when features from subsequent images are associated together. Feature descriptors are used to associate features in different images, e.g. through a FLANN nearest neighbor search [81]. Optical flow has been used in vision-based navigation systems for fixed-wing UAVs [30, 31, 51], and is used for velocity extraction in Chapter 5. An example where optical flow between two images is calculated with the SIFT feature detector is shown in Figure 4.1.

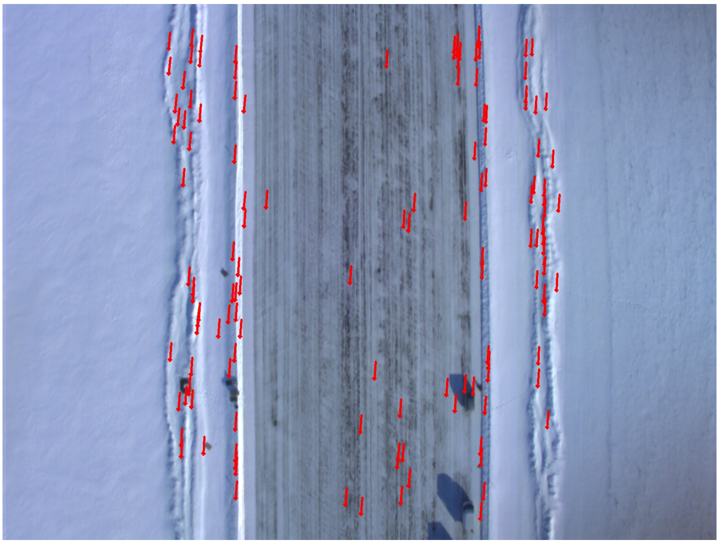


Figure 4.1: Optical flow between two aerial images calculated by SIFT

Thermal images at sea have a homogeneous background and features are mostly present on targets. If it is assumed that features are only located on objects, an estimate of the object position in the image can be calculated as the mean position of features clustered together. Moreover, a mean value for the optical flow of an object can be calculated in the same manner. This strategy can be used for object detection at sea in simplified scenarios, but is not robust in general. A thermal image with features and optical flow vectors for the corresponding features are displayed in Figure 4.2. How optical flow can be utilized in the context of target velocity reconstruction is described in Section 5.4.

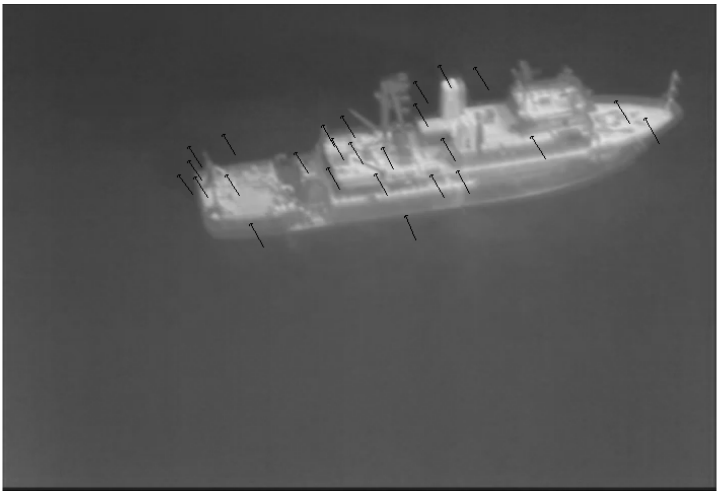


Figure 4.2: Optical flow (arrows) and features (arrow origin) in a thermal image at sea

4.3 Floating Object Detection

The previous section discussed feature detection and optical flow, but it is necessary to identify a method for target detection in thermal images that is more robust. This section describes a machine vision system that utilizes thermal images to find floating objects at sea. The object detection module extracts the pixel position of the center of each object, which fits into the framework of tracking of point masses described in Chapter 3. The method is also able to identify the contour of objects if extended object tracking is used instead. The method runs in real-time on a small embedded computer and can be used in applications that require on-line detection of objects. The method is tailored for thermal images with a homogeneous background (such as the sea surface) and uses classical image-processing techniques. Therefore, it is not equally effective in other environments. The method was originally presented in [64] and later extended in [43].

A raw thermal image is denoted $\mathbf{I}$. A typical thermal image at sea with three marine vessels is displayed in Figure 4.3(a). The raw image might be affected by noise. Therefore, the first processing step is image smoothing, which reduces the influence of noise. The drawback with smoothing is that the detail level is reduced (less contrast) because smoothing is a form of averaging. The smoothing operation is conducted by convolving (denoted by $*$) the image with a Gaussian kernel $\mathbf{g}$. The kernel $\mathbf{g}$ with size $n \times n$ approximates a Gaussian distribution with standard deviation σ_g. Thus, the operation can be formulated as

$$\mathbf{I_s}[x,y] = (\mathbf{I} * \mathbf{g})[x,y] = \sum_{k=-\frac{n-1}{2}}^{k=\frac{n-1}{2}} \sum_{m=-\frac{n-1}{2}}^{m=\frac{n-1}{2}} \mathbf{I}[x-m, y-k]\mathbf{g}[m,k] \tag{4.1}$$

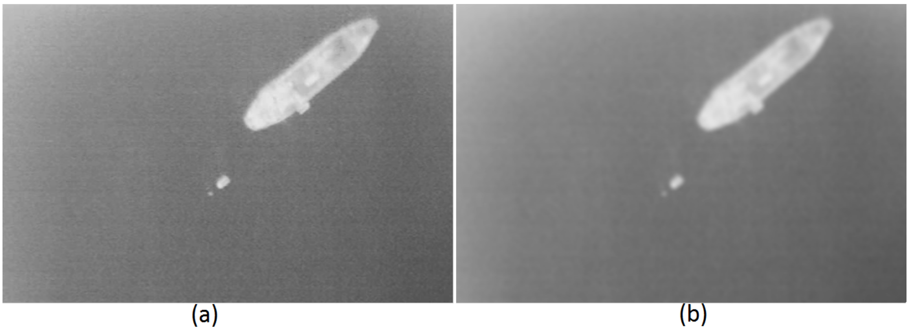


Figure 4.3: Before (a) and after (b) smoothing of the original image. The image is showing a large boat (length of 55m), a rigid-inflatable boat and a small buoy.

where $\mathbf{I}_s$ is the resulting image after smoothing. $\mathbf{I}$ and $\mathbf{I_s}$ are $w \times h$ matrices, where w and h are the width and height of the raw image $\mathbf{I}$. Moreover, $[x, y]$ are integers representing a pixel coordinate in the image, and $[m, k]$ are integers representing a coordinate in the kernel $\mathbf{g}$. The result of smoothing the image in Figure 4.3(a) is shown in Figure 4.3(b). A big vessel with length 55m, a rigid-hulled inflatable boat (RHIB) and a small buoy are visible in the images.

The contour around the objects is visible for the human eye in the smoothed image and there is a clear edge between the objects and the background. Consequently, the next step in the detection process is to use edge detection to sharpen these edges. A simple and computationally cheap way to find edges is through the gradient image, which is calculated from $\mathbf{I}_s$. The gradient image $\mathbf{G_I}$ is calculated as:

$$\begin{aligned}
\mathbf{G_I}_x[x,y] &= (\mathbf{I}_s * \mathbf{P})[x,y] = \sum_{k=-1}^{k=1}\sum_{m=-1}^{m=1} \mathbf{I}_s[x-m, y-k]\mathbf{P}[m,k], \\
\mathbf{G_I}_y[x,y] &= (\mathbf{I}_s * \mathbf{P}^T)[x,y] = \sum_{k=-1}^{k=1}\sum_{m=-1}^{m=1} \mathbf{I}_s[x-m, y-k]\mathbf{P}^T[m,k], \\
\mathbf{G_I}[x,y] &= \sqrt{\mathbf{G_I}_x^2[x,y] + \mathbf{G_I}_y^2[x,y]}
\end{aligned} \tag{4.2}$$

where $\mathbf{P}$, also referred to as the Prewitt operator [105], is defined as the 3×3 matrix

$$\mathbf{P} := \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.3}$$

The result of calculating the gradient image from the smoothed image in Figure 4.3(b) is shown in Figure 4.4(a). The intensity of the objects is much stronger than the intensity of the background in the gradient image. Hence, it is easy to realize that multiple objects are present. However, it is also a significant amount of noise in the gradient image, probably caused by waves and ripples that affect the emissivity. In addition, a part of the noise from the raw image is still visible.

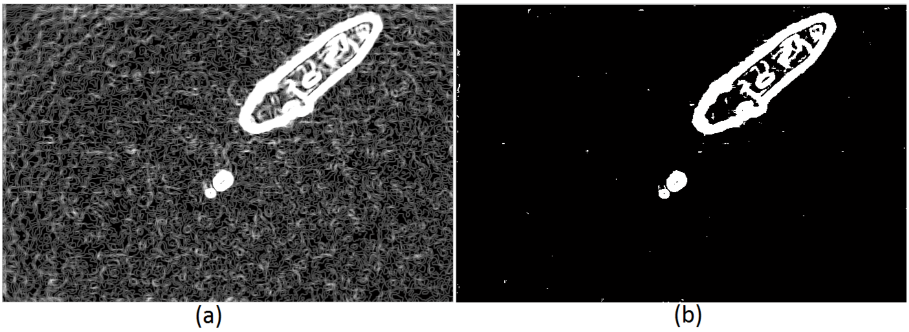


Figure 4.4: Gradient image before (a) and after (b) gradient magnitude thresholding

Nevertheless, the noise intensity is weaker in magnitude than the objects and can be removed by using a threshold value for the magnitude of the gradients. That is, all pixels in the gradient image that have a magnitude less than a certain threshold T_g are interpreted as background. This is achieved by the following thresholding operation

$$\mathbf{G_I}(x,y) = \begin{cases} \text{maxValue} & \text{if } \mathbf{G_I}(x,y) \geq T_g \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

where maxValue is the maximum brightness value a pixel in $\mathbf{G_I}$ can take and is based on the bit depth in the image. Figure 4.4(b) shows the resulting image after gradient magnitude thresholding. Mostly objects with a distinct heat signature are left in the image.

The resulting image after thresholding is repeated in Figure 4.5(a). Some of the bright pixels (blobs) in the image do not originate from objects (i.e, the small dots scattered across the image). These pixels must be filtered out and this is achieved through a connected component algorithm [108], which groups neighboring components together in blobs. Moreover, the area of each blob is calculated, and blobs with a smaller or larger area than what is expected from an object of interest are removed from the image. The result of this process is seen in Figure 4.5(b) and referred to as the binary image, $\mathbf{B}$, of the raw image $\mathbf{I}$.

The remaining blobs in the binary image are interpreted as objects. Bounding boxes for detected objects are drawn in the original image in Figure 4.6(a). Large objects with texture within the interior can trigger multiple bounding boxes within one object. Thus, the next step is to remove bounding boxes contained inside a larger box. The result of this process is seen in Figure 4.6(b). Three objects are remaining after the entire process and are interpreted as objects.

A consequence of using edge detection is that only the exterior edges of an object is detected. When performing for instance recognition based on characteristics such as size, average temperature and overall form, it is necessary that the whole object is evaluated. Thus, the final step is to expand the detections to also include the interior of objects. This is conducted with an algorithm that seeks to fill holes in

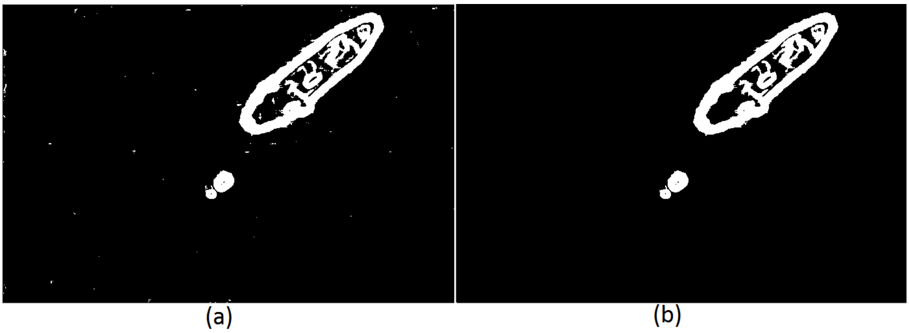


Figure 4.5: Before (a) and after (b) removing blobs which are either too small or too large to be an object of interest

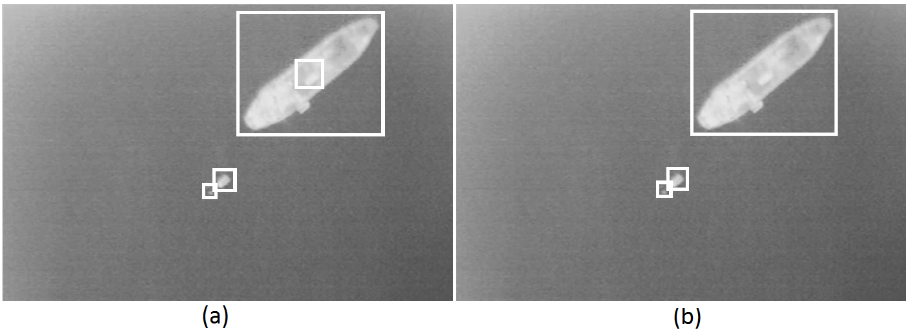


Figure 4.6: Before (a) and after (b) removing detections completely contained in the interior of other detections

the binary image [105]. The result of this operation is shown in Figure 4.7(b). The detection algorithm provides the position of the object center in the image. These positions are later processed in the tracking system and this is described further in Chapter 6.

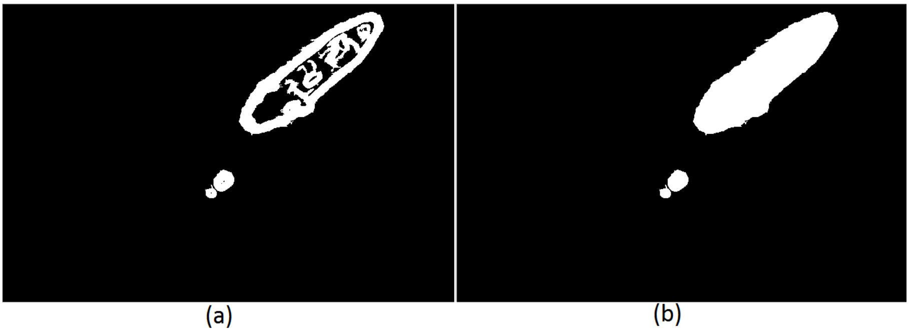


Figure 4.7: Before (a) and after (b) filling the interior holes in the detected objects

A special case arises when objects are located close to the boundary of the image, such as in Figure 4.8(a). The center of the detected object in this image is not going to correspond with the true center of the object. The calculated detection center is illustrated with a red cross in Figure 4.8(b), and since the boat is only partly visible in the image, the blob center does not coincide with the true object center. In tracking, this leads to measurements with a systematic bias that can degrade the accuracy of the target estimates. Therefore, it is desired to compensate for this issue. A simple solution is obviously to neglect detections that intersect the image boundary. However, it is possible to adjust the center of detection to a position that is closer to the true center if the object size is known or have been detected previously.

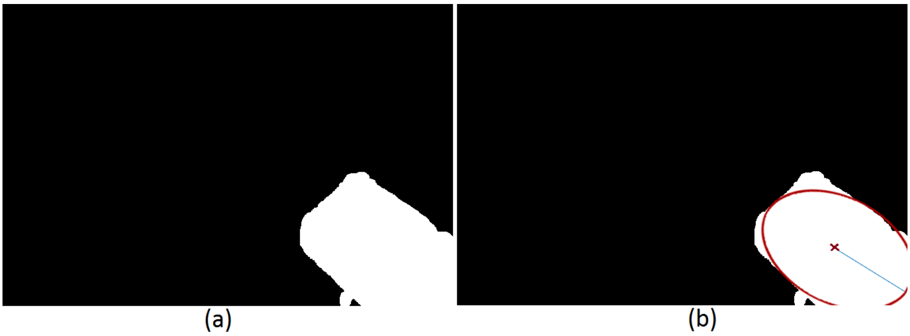


Figure 4.8: Partially visible object (a), and its corresponding ellipse approximating the object shape (red circle), object center (red cross), and the orientation of the major axis (blue line) illustrated in (b)

The orientation of the detected object is calculated by approximating the detected blob as an ellipse with a minor and a major axis. This can be done by using the second order central moments to construct a covariance matrix for pixels in the detected blob. The $(q+p)$th order central moment of a blob or an image region $\mathbf{O}$ is [108]

$$m_{pq} = \sum_{x,y \in \mathbf{O}} x^p y^q \mathbf{B}(x,y), \quad p,q = 0,1,... \tag{4.5}$$

and the second order covariance matrix of an image region $\mathbf{O}$ is calculated as [108]

$$cov[\mathbf{O}(x,y)] = \begin{bmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{bmatrix} \tag{4.6}$$

The eigenvectors of this matrix correspond to the major and minor axes of the ellipse approximating the interior of the object in the image region $\mathbf{O}$. As a result, the angle between the major axis and the image boundary can be used to approximate the orientation of the detected blob. Figure 4.8(b) shows the ellipse constructed using the two eigenvectors as major and minor axes.

The blue line shows the direction of the major axis of the blob, and can be used as a measurement of the direction. Note that in order for this process to yield information about the orientation of an object, the object has to be non-circular, and a sufficient part of the object has to be visible so that the length of the major axis of the blob is larger than the expected full length of the minor axis.

If it is assumed that the full length of the partially detected object is known prior to this image, the expected length of the boat in pixels can be calculated. Using this information combined with the results of the above mentioned calculations, a more accurate estimate of the object center in the image can be identified. This is achieved by moving the calculated blob center the following amount

$$\Delta\mathbf{c} = \frac{\mathbf{L}}{2} - (\mathbf{a} - \mathbf{i}) \tag{4.7}$$

where $\mathbf{L}$ is the expected object length decomposed in the image plane according to the approximated orientation, i.e $\|\mathbf{L}\| = L$, where L is the expected length of the object. $\mathbf{a}$ is the major axis and $\mathbf{i}$ is the vector from the object center to the closest intersection with the image boundary along the direction of the major axis, illustrated as a blue line in Figure 4.8(b). Adding $\Delta\mathbf{c}$ to the calculated blob center effectively moves the center along the calculated major axis of the blob towards the expected location of the true object center. This can sometimes lead to an object center outside of the image boundaries, but this is not an issue in georeferencing.

The steps in the floating object detection algorithm can be summarized as:

1. Smooth the raw thermal image with a Gaussian kernel to reduce the noise level.
2. Calculate the gradient image of the smoothed image using the Prewitt operator.
3. Use gradient magnitude thresholding to remove noise caused by waves, ripples and other disturbances in the gradient image. This creates a binary image where objects are white and the background is black.
4. Use a connected component algorithm to identify pixels that are linked together (blobs) and remove blobs that are smaller than a certain size. The intention is to remove white pixels not belonging to objects.
5. Calculate bounding boxes for contours that are remaining and remove all bounding boxes completely contained in a larger bounding box.
6. Fill the interior of the objects to identify characteristics such as size, shape and object center.
7. (*Optional*) For objects intersecting the boundary, move the detected center towards the true center if the size of the object is known.

The detection algorithm described here is able to run in real-time on-board UAVs using a small embedded computer. In [67], the detection algorithm was tested together with a tracking system based on a Kalman filter, and the total computation time was 0.07 seconds for detection and update of the tracking system on a small embedded computer. Consequently, more than ten images can be processed each second. The major limitation with the algorithm is that it requires a homogeneous background to work efficiently. Nevertheless, when used at sea, the detection rate is near perfect as will be demonstrated later. Moreover, the number of false alarms is limited. Detection of humans, vessels and buoys have successfully been conducted with this algorithm.

The algorithm requires two parameters that must be designed by the user. The first parameter is the detection threshold T_g in (4.4). This parameter depends on the intensity of the object gradients. A threshold that is too low will lead to more noise being processed further in the binary image. A threshold that is large might risk removing parts of objects. Thus, this is a trade-off between removing the noise and risk removing parts of objects. The second parameter is the smallest expected area. The value depends on the scenario. A human covers fewer pixels than a small vessel. Nevertheless, the intention with this step is to remove the remaining part of the noise. Most of the remaining noise, that is connected together in blobs, only cover a few pixels. Therefore, it is most often possible to remove the noise without removing small objects. Moreover, a few false alarms are not critical if the appropriate tracking filter is chosen.

The method presented here is able to extract the interior of an object. As a result, a possibility is to combine the method with a feature detector to calculate optical flow for features located in the interior of the object. This approach is not discussed further in this thesis, but solves the challenges discussed at the end of Section 4.2.

4.4 Case Studies

This section presents two independent case studies used to illustrate how different objects can be detected. The case study in Section 4.4.1 investigates the effectiveness of the algorithm in Section 4.3 on several data sets containing thermal images captured at sea. The case study in Section 4.4.2 investigates how object detection can be utilized in precision drop, which is an interesting application for UAVs. A modified detection algorithm is used in the second case study. This is because a visual spectrum camera is utilized and the objective is to find an object with a distinct color on the ground.

4.4.1 Case study 1 - Detection of marine vessels

The intention with this case study is to illustrate the effectiveness of the detection algorithm described in Section 4.3 in detection of marine vessels and to show a couple of examples. Moreover, the case study ends with a short discussion regarding the adjustable parameters in the detection algorithm.

This case study summarizes the results presented in two different articles [43, 67]. The detection algorithm has also been used in several case studies later in this thesis. In [43], the detection algorithm was used to detect the marine vessel shown in Figure 4.2. 441 images included the vessel and the vessel was successfully detected in 438 images, which corresponds to a detection probability of 0.993. The high success rate is explained by the high signal to noise ratio between the sea surface and the vessel in most images. The temperature and emissivity differences between marine vessels and the sea surface are often large and enables the detection algorithm to work well in this type of scenario. The vessel was not detected in three images because only a small part of the vessel was visible in these images. As mentioned in Chapter 3, clutter and false detections are challenging to handle in target tracking. The 441 images that were processed in [43] did not produce a single false positive, which means that the algorithm is robust without missing many true positives.

A much larger set of images was analyzed in [67]. 27866 images were processed in this work, and a subset of 4903 images contained one or two marine vessels where the entire contour was inside the field of view. The vessels were detected in all images. These results support the detection rate experienced in [43]. A total of 15 false positives were experienced when all images were processed. These false detections were examined manually and belonged to unidentifiable thermal signatures.

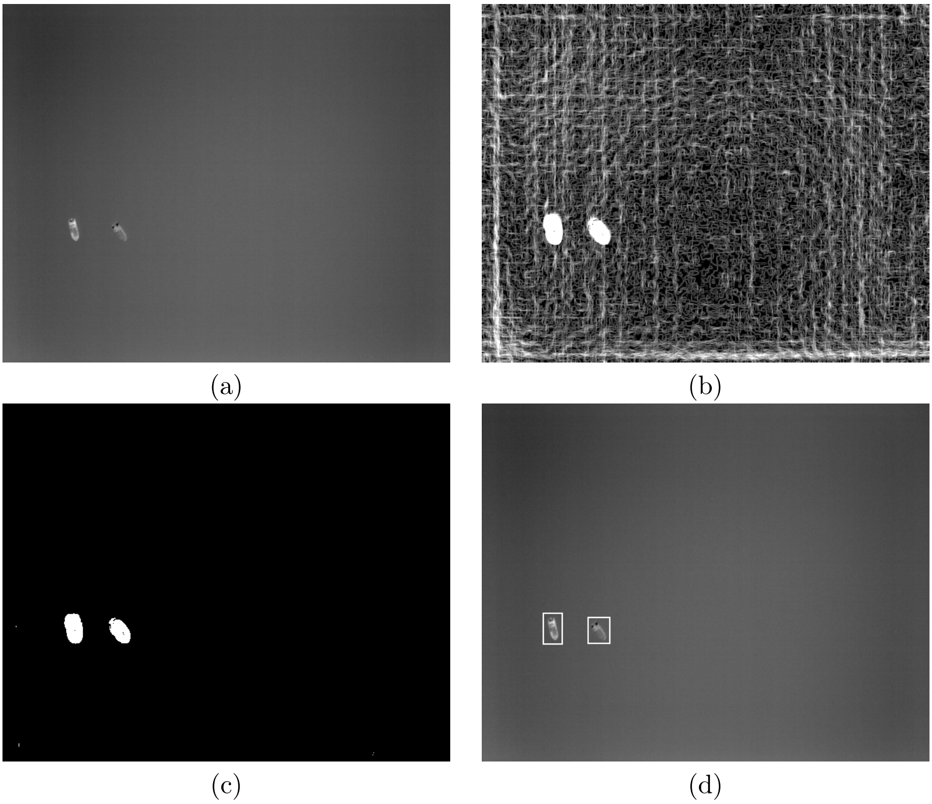
Examples

Figure 4.9 shows an example where two marine vessels are located in the same image. Figure 4.9(a) is the raw thermal image. Two vessels are visible in this image. The gradient image is shown in 4.9(b) where the contours of the objects are amplified. The binary image after gradient magnitude thresholding is shown in Figure 4.9(c) and only the vessels and a few pixels with noise (bottom left corner) are remaining. This image is filtered based on the minimum expected area of the contours and the noise is removed. Figure 4.9(d) shows the raw image where bounding boxes are drawn around detected objects.

Figure 4.10(a) shows an example of a raw thermal image without any objects present. The gradient image of this image is shown in Figure 4.10(b). Thermal noise caused by waves and other factors are present since the gradient image has some texture. The noise is greatest near the image boundaries, which may indicate that the thermal image sensor is most prone to noise near the boundaries. Nevertheless, the intensity is much lower than the one in Figure 4.9(b) and no objects are detected in this image after gradient magnitude thresholding and removal of small contours.

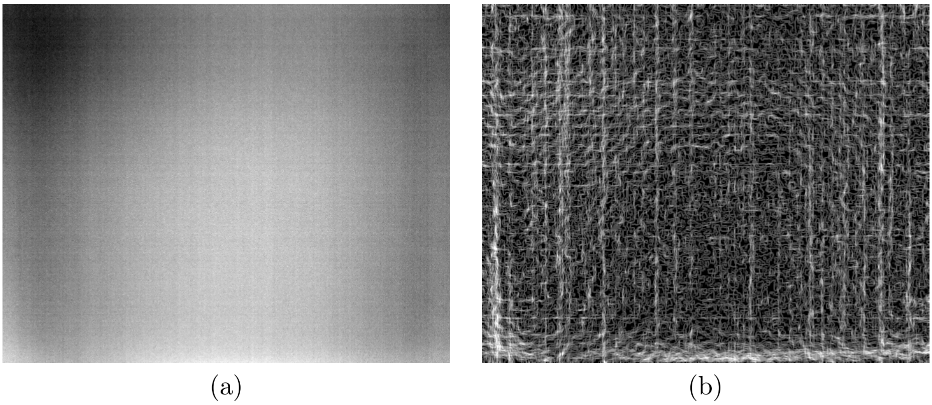
Adjustable parameters in the detection algorithm

Two parameters are adjustable in the detection algorithm and the values for these parameters have not been discussed. Both of these values have been constant in the data discussed in this case study. The first adjustable parameter is the gradient



(a) (b) (c) (d)

Figure 4.9: Example of multiple object detection



(a) (b)

Figure 4.10: Example of image without detection

magnitude thresholding value which was chosen to be 230. The pixel depth of the gradient image was typically 8 bits, which means that each pixel could take a value between 0 and 255. The second adjustable parameter is the smallest accepted area of an object. It was chosen to be 50 pixels and most marine vessels are significantly larger. This parameter must obviously be adjusted if objects of interest include missing people or smaller structures. A possibility is to scale this parameter with the altitude of the UAV since the altitude directly affects the perceived size of an object in the image.

An advantage with this algorithm is that noise is suppressed effectively. The gradient image is typically affected by noise so that a few pixels have intensity larger than 230. This also happens in images without objects. However, noise is seldom distributed together in larger blobs. Thus, the second parameter ensures that only larger blobs are kept and that single pixels cannot be interpreted as an object.

4.4.2 Case study 2 - Detection for precision drop

Precision drop is delivery of small packages to specific locations [32]. Fixed-wing UAVs can be used to deliver items autonomously in missions where machine vision is used to locate the target. Another system use that location to calculate a drop point based on the altitude, speed and wind conditions. This sort of operation is useful in rescue missions where items are delivered to people in need or in surveillance of icebergs where GPS trackers are dropped on icebergs [80].

This case study is based on the work in [79]. It is meant to illustrate that a similar detection technique, as the one presented in Section 4.3, can be used in other applications. The goal in [79] was to drop an item on a blue tarpaulin autonomously with the Skywalker X8 fixed-wing UAV displayed in Figure 4.11. An important part of this operation was to find the target (tarpaulin) using machine vision. A visual spectrum camera was used to collect images of the area, and the UAV had no knowledge about the position of the tarpaulin beforehand. The UAV looked for visual features and used machine vision with real-time on-board data processing to identify the target on the ground. Image processing based on color segmentation was used to detect the target and find the pixel position in the image.

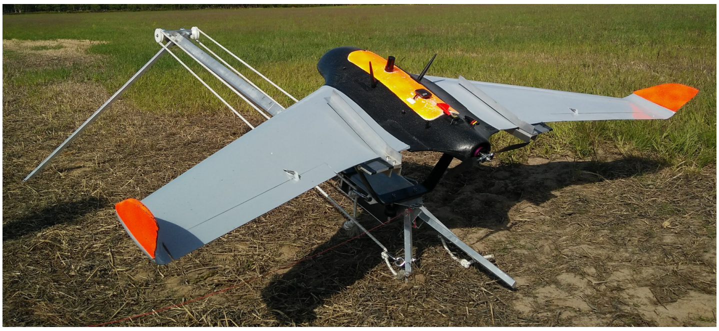


Figure 4.11: The Skywalker X8 used for precision drop

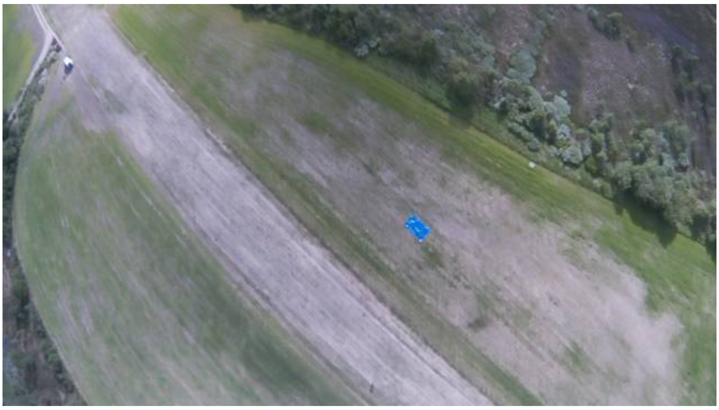


Figure 4.12: Raw image with the blue tarpaulin visible in the middle of the image

The target had a distinct blue color which is easy to separate from the background in the visual spectrum as long as the horizon is avoided. This is illustrated in Figure 4.12 where the tarp is visible in the middle of the image. Therefore, the target detection algorithm developed for this purpose utilizes color segmentation, which is a simple and effective strategy in this particular scenario.

A pixel in a visual spectrum image is normally represented as a 3D-vector with three color values, namely R (red), G (green) and B (blue). However, color segmentation is usually easier to conduct in the HSV (hue, saturation, value) color space. The HSV color space separates color information from lighting and reduces the impact from differences in lighting conditions. Consequently, color thresholding techniques are more reliable and effective when using the HSV representation.

The main goal with color segmentation is to create a binary image, which is separated into foreground (object) and background (no object). This is similar to gradient magnitude thresholding in Section 4.3. The detection algorithm developed to search for the blue tarpaulin consists of the following steps, which were executed in real-time on the on-board computer:

1. Downscale the images to a lower resolution to decrease the computation time. Images downscaled to a resolution of 640 × 360 pixels were used in the experiments.
2. Transform the color space of the image from RGB to HSV.
3. Use thresholding to filter out blue regions in the image. This gives you a binary image where blue regions are light and other regions are black. Figure 4.13 shows the resulting image after this operation.
4. Search for contours in the binary image to find pixels that are connected and creates a region.
5. Calculate the area of all detected regions and keep the detection with the largest area. Discard the detection if the area is smaller than an adjustable

value. The area limit was chosen to be 50 pixels in the experiments. This step is used to filter out noise and regions that only consist of a few pixels. Moreover, regions inside larger regions are also removed. Only the largest region is kept since a single detection is desired and the tarpaulin is the only blue object expected in the mission area. An extension of this method can be used if several targets are of interest.

6. Calculate the pixel center of the detection. Accept the detection if the center is at least 30 and 60 pixels from the nearest image boundary in the horizontal and vertical direction, respectively. Detections too close to the border are discarded because they can be caused by the horizon.

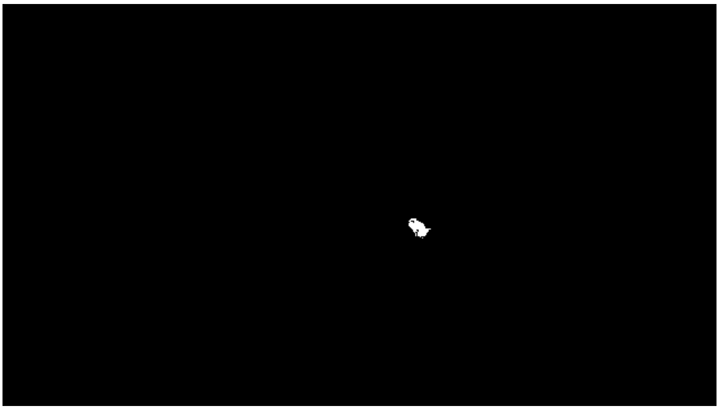


Figure 4.13: The binary image after thresholding. The tarpaulin is easy to distinguish from the background.

Figure 4.14 shows the resulting image when the image in Figure 4.12 is processed through the steps above. The white circle is the detected center of the tarpaulin and the red regions show the contours detected in the image. It is critical to avoid false detections in this particular application because an item is supposed to be dropped on the target. Thus, it is necessary to use ranges for the thresholding operation that are strict and avoid contours with a color that is somewhat similar. That is also the reason for why the entire tarpaulin is not marked in Figure 4.14 since the blue color is weaker near the boundary of the tarpaulin. Nevertheless, this is not restricting the accuracy of the detected position in the image since the detected region is centered around the true center. Moreover, missing the true center with a few pixels may not be critical as it corresponds to a small distance in the Earth-fixed coordinate frame. During the experiments, the following thresholding values were used:

- Hue (has a range from 0 to 180) - Values in the range from 100 to 110 kept as a possible object. This is where the blue color is located in Hue.
- Saturation (has a range from 0 to 1) - Values from 0.75 to 1 kept as a possible object. The saturation value models various shades of color and a large value indicates that the object has a bright color with less gray shades.

- Value (has a range from 0 to 1) - Values from 0.75 to 1 kept as a possible object. The value component describes the amount of white and black a pixel contains and a large value represents a larger amount of white.

These parameters were tailored for detecting the tarpaulin in a robust manner. The large saturation and value components were used because the tarpaulin has a bright blue color without gray shades.

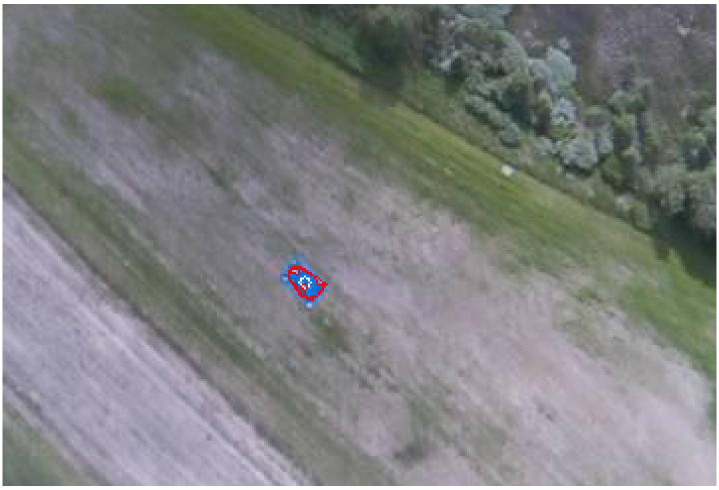


Figure 4.14: The resulting processed image with the detection marked. The red areas mark the detected contours and the white circle marks the center of detection that is used for georeferencing.

When a valid and accepted detection was identified during the experiment, georeferencing was used to find Earth-fixed coordinates of the object in real time. Real-time georeferencing is described in the next chapter. The georeferenced position from a single detection is somewhat inaccurate in general and false detections are problematic if only a few detections are available. It was desirable to find the target in several images before the object was dropped. Therefore, approximately 75 detections were gathered before each drop and the mean georeferenced position was used as the target position. Multiple successful drops within a few meters of the target were conducted. The details can be read in [79].

Chapter 5

Georeferencing

Optical sensors are suitable for mapping and surveillance of unknown territories and useful for situational awareness. Images can be used to detect undiscovered objects or gather information about the proximity of the sensor. A vital part of mapping and surveillance is the ability to relate findings in the optical sensor frame to Earth-fixed coordinates. This is called georeferencing and is a fundamental part of any mapping and surveillance system using a monocular optical sensor [49, 65]. This chapter concerns real-time georeferencing for passive optical sensors mounted in fixed-wing UAVs. Conceptually, there is no difference in georeferencing of thermal and visual spectrum images. The work presented in this chapter is mainly based on the work published in [44, 46] and the following topics are covered:

- Section 5.1 gives an introduction to georeferencing and related literature with a specific focus on optical sensors.
- Section 5.2 defines the mathematical notation and coordinate frames used in this chapter.
- Section 5.3 derives mathematical formulas for georeferencing using optical sensors mounted in fixed-wing UAVs. It also covers compensation of sensor misalignment errors.
- Section 5.4 explains how optical flow can be used to reconstruct the velocity of objects detected in two consecutive images. This is a concept closely related to georeferencing.
- Section 5.5 describes a navigation system based on a nonlinear observer that is used to estimate the UAV pose. Knowing the pose of the UAV accurately is necessary in georeferencing.
- Section 5.6 presents three independent case studies investigating the topics in this chapter experimentally. Case studies 1 and 2 concern georeferencing of a static object and of a moving vessel, respectively. Case study 3 concerns velocity reconstruction using optical flow.

5.1 Background & Related Work

Georeferencing is an important task in target tracking and remote sensing applications. It is particularly useful for sensors on UAVs because the sensor-specific coordinates depend on the position and attitude of the UAV. In addition, the motion of the camera (UAV) usually affects the target pixel position more than the target motion itself in many applications. Converting these measurements to an Earth-fixed frame is useful for the end user or other vehicles in the area. Georeferencing using a monocular camera is challenging since range information is missing. Consequently, it is hard to accurately reconstruct three unknown Earth-fixed coordinates from a single image which only has pixel coordinates. However, it is possible to remove the depth dependency if it is assumed that all pixels lie in the same physical plane. This is for example valid at the sea surface where objects are located on approximately the same altitude as the mean sea level. It is also a reasonable assumption if the terrain variations are small compared to the altitude of the camera or when a digital elevation map is available. In these situations, georeferencing can be used to find two-dimensional horizontal coordinates from a single image without any reference points on the ground.

The foremost challenge in real-time georeferencing is to obtain accurate coordinates from a single image without using ground reference points. Ground reference points are static markers with known position in the Earth-fixed frame. When these markers are visible in the images, it is possible to calculate the position of other objects much more accurately through the relative position. A solution with reference points is not wanted because it severely restricts the operational range and the ability to map new environments. It is also desirable to conduct georeferencing in real-time on a small embedded computer without needing to process data on a ground station. This is beneficial if the UAV operates in areas far away from the ground station and without a reliable and fast communication link.

Georeferencing relies on the camera pose (position and attitude) at the exact time when an image is captured. Small deviations from the true pose give large errors in the georeferenced position. Hence, conducting real-time georeferencing of thermal images on-board a UAV with significant roll and pitch motions is more difficult than georeferencing from a stationary or slowly-moving platform. Accurate time synchronization among the sensors is critical. In addition, thermal images have lower resolution than visual spectrum images and are consequently more fragile for pixel and sub-pixel errors that occur in object detection.

The most accurate existing solutions for real-time georeferencing use reference points on the ground at known locations [40, 41, 121, 132], but that is not feasible in unknown territories or when no markers are within the field of view. The solution presented in [15] requires common features in consecutive images, which is hard to obtain in thermal imagery of marine areas because features are rare on the sea surface, particularly in situations without any objects present. This is therefore, not a robust solution at sea.

A few systems for direct real-time georeferencing without the use of reference points exist. However, these solutions lack the desired accuracy, are not suitable for fixed-wing UAVs or are not validated experimentally with a significant amount of data. A georeferencing system for fixed-wing UAVs with a thermal camera is presented in [65], but the variance in the results is large when considering the altitude of the UAV, which is below 100 meters. Capable accuracy is shown in georeferencing of the length and width of a runway [49] at an altitude exceeding 300 meters, but the results are based on a single image so the amount of data is limited. Moreover, reconstructing Earth-fixed coordinates is a different task because calibration errors in the principal point do not affect the size of an object. Challenges related to synchronization between the sensors and mounting misalignments are not addressed thoroughly in the aforementioned works. Time synchronization and camera calibration are addressed in [19, 93], but these solutions are tailored towards multi-copters so it is arguably hard to verify how they apply to the fast dynamics and operating envelope of fixed-wing UAVs.

The rest of this chapter addresses real-time georeferencing for optical sensors mounted in UAVs. The focus is to do this efficiently and accurately on a small embedded computer without using reference points on the ground. Moreover, identification of mounting misalignments is addressed thoroughly because it is key for the accuracy. Time synchronization was discussed in Section 2.5.2.

5.2 Preliminaries

This section gives a short introduction to the mathematical notation used in this chapter, and to the pinhole camera model which is key in georeferencing and target tracking. It also presents the different coordinate frames that are relevant in georeferencing.

5.2.1 Mathematical notation

Vectors and matrices are represented by lowercase and uppercase bold letters, respectively. $\mathbf{X}^{-1}$ denotes the inverse of a matrix and $\mathbf{X}^\top$ the transpose of a matrix or vector. $\mathbf{0}_{m\times n}$ is a matrix of zeros with dimension $m \times n$. $\mathbf{I}_n$ is the square identity matrix of dimension n. The operator $\mathbf{S}(\mathbf{x})$ transforms the three-dimensional vector $\mathbf{x}$ into the skew-symmetric matrix

$$\mathbf{S}(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{5.1}$$

The skew-symmetric operator can be used for the vector cross-product between two vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b} \tag{5.2}$$

A rotation matrix between two coordinate frames a and b is denoted $\mathbf{R}_b^a$ and is an element of SO(3) - the special orthogonal group of matrices of dimension 3×3 [29]. The matrix $\mathbf{R}_b^a$ transforms a vector $\mathbf{v}$ decomposed in frame b to frame a by the following transformation:

$$\mathbf{v}^a = \mathbf{R}_b^a \mathbf{v}^b \tag{5.3}$$

The Greek letters ϕ, θ, and ψ represent the roll, pitch, and yaw angles of the UAV respectively, defined according to the zyx convention for principal rotations [29]. Optical sensors are often mounted in a gimbal in UAVs. It is assumed that a gimbal with two degrees of freedom is used. Gimbal pan and tilt angles are denoted ψ_{gb} and θ_{gb}, respectively. They are simply set to zero in situations where the camera is strapped directly to the airframe. A two-dimensional camera image has coordinates (u, v) in the image plane. The derivatives ($[\dot{u}, \dot{v}]^\top$) of the image plane coordinates are called optical flow and was defined in Section 4.2. $s\theta$ and $c\theta$ denote the sine and cosine functions with angle θ as input.

5.2.2 Coordinate frames

Several coordinate frames are considered in this thesis and some of them are illustrated in Figure 5.1. The following reference frames are considered in this chapter:

- The body-fixed frame {b} is fixed to the UAV and usually originated in the center of gravity [7]. The x axis of {b} points forward out the nose of the UAV in the longitudinal direction, the y axis points out the lateral direction (to the right) and the z axis points down out of the belly of the UAV. This frame is used both for navigation and remote sensing since sensors often are mounted relative to {b} with a lever arm and a specific orientation.
- The Earth-centered-Earth-fixed (ECEF) coordinate frame denoted {e} has its origin in the center of the earth and the axes rotate with the Earth. Therefore, it rotates with respect to an inertial frame with angular velocity given by the rotation rate of the Earth. It is used for guidance and navigation on a global scale.
- The north-east-down frame (NED) {n} is a tangent plane on the surface of the Earth. The x axis points towards the true north, the y axis towards east and the z axis down. The origin of {n} is usually defined at a specific longitude and latitude, but can also move with the UAV in many applications. In this thesis, NED is fixed and not moving with the UAV. It is assumed to be inertial locally (for locations close to the origin and for shorter periods). NED is used for navigation, georeferencing and target tracking.
- The camera-fixed frame {c} is placed in the principal point of the camera. The x axis points along the horizontal image dimension and the y axis points in the vertical direction in the image. The z axis points straight out of the camera lens. The lack of range information in monocular optical sensors arise from the fact that the distance to the pixels along the z axis is unknown. The

camera-fixed frame is used in georeferencing and target tracking to relate pixels in the image plane to other coordinate frames.

- The frame {m} is called the mounted frame and is used in situations when a gimbal is available or when the camera is mounted in a specific orientation relative to {b}.
- The image plane is a two-dimensional coordinate frame. The horizontal component is denoted u and the vertical component v (see Figure 5.1).

The rotation from {n} to {b} is represented by the matrix $\mathbf{R}_b^n \in SO(3)$. Similar transformations exist between the other reference frames. A vector decomposed in {b}, {n}, {c}, {e} and {m} has superscript ^{b}, ^{n}, ^{c}, ^{e} and ^{m}, respectively. A vector $\mathbf{p}_{nb}^n$ is interpreted as the position of {b} relative to the origin of {n} decomposed in {n}.

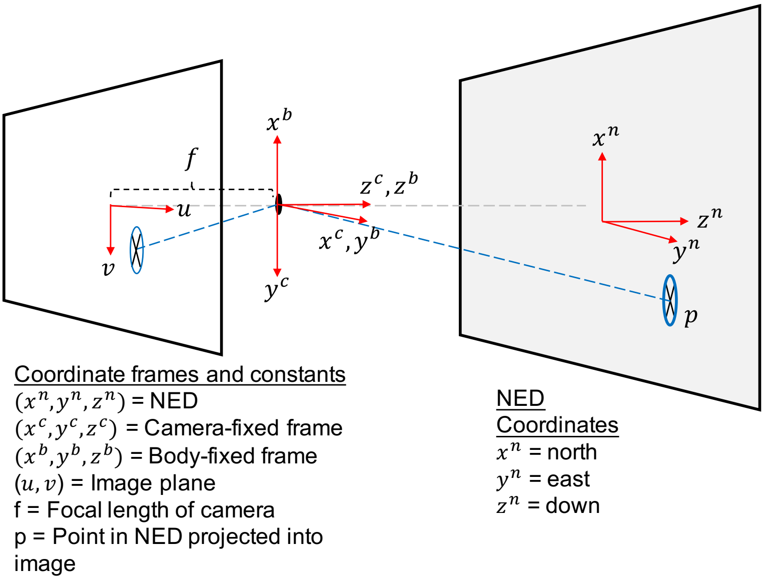

f
x^b
u
v
z^c, z^b
x^c, y^b
y^c
x^n
z^n
y^n
p
Coordinate frames and constants
(x^n, y^n, z^n) = NED
(x^c, y^c, z^c) = Camera-fixed frame
(x^b, y^b, z^b) = Body-fixed frame
(u, v) = Image plane
f = Focal length of camera
p = Point in NED projected into image
NED Coordinates
x^n = north
y^n = east
z^n = down

Figure 5.1: Illustration of coordinate frames

5.2.3 The pinhole camera model

A camera model is defined as

Definition 5.1 A camera model is a mathematical model describing the relationship between pixel coordinates in the image plane and the corresponding position of the pixel in the camera-fixed frame {c}.

There are several camera models, but one of the simplest with high validity is the pinhole camera model [55]. It is a familiar model which defines the mapping between the image plane and the camera-fixed frame in a simple manner. The axes of {c} can be defined in different ways. The axes in Figure 5.1 are used in this thesis, which means that the x axis of {c} coincides with horizontal direction in the image plane (u). Consequently, the relationship between the image plane coordinates (u, v) and the camera-fixed coordinates (x^c, y^c, z^c) is given as

$$\frac{u}{f} = \frac{x^c}{z^c} \tag{5.4a}$$

$$\frac{v}{f} = \frac{y^c}{z^c} \tag{5.4b}$$

where f is the focal length of the camera given by the lens specification. (5.4a) and (5.4b) are often combined and expressed as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix}, z^c \neq 0 \tag{5.5}$$

The range z^c can obviously never be zero, but this is not a problem for UAVs since z^c is proportional to the altitude of the UAV as long as the camera is pointing towards the ground.

The pinhole camera model is used later in georeferencing and when the velocity of objects is reconstructed with optical flow. It is a good approximation, but it neglects lens distortion and assumes that the image projection is continuous. All cameras are prone to lens distortion to some extent because of manufacturing errors. Lens distortion could e.g. be identified as straight lines appearing as bended lines in an image. The geometric distortion is most present in the peripheral of an image so objects close to the image boundary are more sensitive to distortion than objects in the center of the image plane. As mentioned previously, distortion is described more closely in [65].

5.3 Georeferencing using Optical Sensors

This section derives specific formulas for georeferencing of images captured by monocular optical sensors. Georeferencing is used as a term for transforming image plane coordinates to NED coordinates in the rest of this chapter and covered in e.g. [46, 49, 65]. Direct georeferencing is georeferencing without using reference points or other known features in the image.

Pixel coordinates have two degrees of freedom and NED coordinates have three. Hence, finding NED coordinates of a single pixel is an ill-posed problem. A common way to avoid this issue (in direct georeferencing) is to assume that all pixels within an image are located in the same plane. This is known as the flat-earth assumption and necessary for obtaining NED coordinates of a single pixel unless an elevation map exists. It is reasonable for an image showing the sea surface, but not in an area with mountains.

The pinhole camera model is the starting point for georeferencing. The matrix form in (5.5) can also be formulated as

$$z_t^c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_t^c \\ y_t^c \\ z_t^c \end{bmatrix}}_{\mathbf{p}_{ct}^c} = \mathbf{A}\mathbf{p}_{ct}^c \tag{5.6}$$

where $\mathbf{A}$ is the camera intrinsic matrix. f_x and f_y are the focal lengths (expressed in pixels) in the horizontal and vertical direction, respectively. c_x and c_y are the principle point (center of lens) and should theoretically be in the image center. Note that the origin of the pixel coordinates (u, v) is assumed to be in the top-left corner of the image in (5.6), and explains why the principle point is added to the model. $\mathbf{p}_{ct}^c$ is the camera-fixed coordinates of the object located at pixel (u, v). The goal in georeferencing is to decompose $\mathbf{p}_{ct}^c$ in NED since the origin of NED is stationary. This is achieved by utilizing a homogeneous transformation $\mathbf{G}_n^c$ between {c} and {n}:

$$z_t^c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{G}_n^c \underline{\mathbf{p}}_{nt}^n \tag{5.7}$$

where $\underline{\mathbf{p}}_{nt}^n$ is the homogeneous coordinate vector of the pixel relative to {n} decomposed in {n}. $\mathbf{G}_n^c$ is the homogeneous transformation

$$\mathbf{G}_n^c := \begin{bmatrix} \mathbf{R}_n^c, & -\mathbf{R}_n^c \mathbf{r}_{nc}^n \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1, & \mathbf{r}_2, & \mathbf{r}_3, & -\mathbf{R}_n^c \mathbf{r}_{nc}^n \end{bmatrix}$$

where $\mathbf{R}_n^c$ is the rotation matrix that transforms a vector from {n} to {c}. The column vectors in $\mathbf{R}_n^c$ are denoted $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$. $\mathbf{r}_{nc}^n$ is the position of the origin of {c} relative to {n} decomposed in {n}, and can be calculated based on the UAV position in NED:

$$\mathbf{r}_{nc}^n = \mathbf{p}_{nb}^n + \mathbf{R}_b^n \mathbf{r}_{bc}^b \tag{5.8}$$

where $\mathbf{p}_{nb}^n$ is the NED positions of the UAV and $\mathbf{r}_{bc}^b$ is the lever arm between the origin of {c} and {b}. The lever arm can normally be neglected for UAVs because the distance is small compared to UAV position in NED, but might be necessary on large vehicles.

Without loss of generality, $\mathbf{R}_n^c$ is divided into three consecutive rotations and expressed as

$$\mathbf{R}_n^c = (\mathbf{R}_b^n \mathbf{R}_c^b)^{-1} = (\mathbf{R}_b^n (\mathbf{R}_m^c \mathbf{R}_b^m)^{-1})^{-1} \tag{5.9}$$

where $\mathbf{R}_b^n$ is the rotation matrix between {n} and {b}, defined according to the zyx convention and specified in terms of the Euler angles (roll (ϕ), pitch (θ), yaw (ψ)) [29]:

$$\mathbf{R}_b^n = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{5.10}$$

The mounted frame {m} is also referred to as the gimbal frame, and is used if the camera is mounted in a gimbal and not strapped on the airframe. $\mathbf{R}_b^m$ relates the body frame to the gimbal frame and is parametrized by the gimbal angles. If the camera is strapped directly to the airframe, $\mathbf{R}_b^m$ is simply chosen as the identity matrix or removed altogether. The second rotation, $\mathbf{R}_m^c$, relates the camera-fixed frame to the gimbal frame and is used to align the image plane with the gimbal frame in a specific way.

$\mathbf{R}_b^m$ depends on the gimbal. In this research, it is assumed that the gimbal has two degrees of freedom, namely pan and tilt angles. It is further assumed that the gimbal is mounted so that {b} is aligned with {m} when the gimbal has zero pan (ψ_{gb}) and tilt (θ_{gb}). In {b}, pan and tilt movements correspond to a rotation along the body z and y axes, respectively. Hence, the rotation is defined as

$$\begin{aligned}\mathbf{R}_b^m &= (\mathbf{R}_z(\psi_{gb})\mathbf{R}_y(\theta_{gb}))^\top = \mathbf{R}_y^\top(\theta_{gb})\mathbf{R}_z^\top(\psi_{gb}) \\ &= \begin{bmatrix} \cos\psi_{gb}\cos\theta_{gb} & \sin\psi_{gb}\cos\theta_{gb} & -\sin\theta_{gb} \\ -\sin\psi_{gb} & \cos\psi_{gb} & 0 \\ \cos\psi_{gb}\sin\theta_{gb} & \sin\psi_{gb}\sin\theta_{gb} & \cos\theta_{gb} \end{bmatrix}\end{aligned} \tag{5.11}$$

where $\mathbf{R}_z(\cdot)$ and $\mathbf{R}_y(\cdot)$ are principle rotations about the z and y axes, respectively [29]. The pinhole camera model (5.5) is defined so that the x axis of {c} is aligned with the horizontal direction in the image plane (u) and not the body-fixed x axis. Therefore, $\mathbf{R}_m^c$ is a rotation of -90 degrees about the camera z axis:

$$\mathbf{R}_m^c = \mathbf{R}_z(-90^o) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.12}$$

In general, $\mathbf{R}_m^c$ depends on how the camera and gimbal are mounted with respect to the body of the UAV, and how the pinhole camera model is defined. Therefore, it is platform dependent.

Only two coordinates in NED can be recovered by a single pixel (u, v) as explained in the beginning of this section. However, since objects at the sea surface are of interest, the down position of pixels is close to zero as long as the origin of {n} is placed at the mean sea level. Consequently, one can identify the north-east (NE) coordinates using a single pixel and set the down position to zero. The NE coordinates ($\mathbf{p}_{nt}^n$) of the pixel (u, v) relative to {n} are given by (5.7) as

$$\frac{1}{z_t^c}\begin{bmatrix} x_t^n \\ y_t^n \\ 1 \end{bmatrix} = \mathbf{G}_{NE}^{-1}\mathbf{A}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5.13}$$

where x_t^n and y_t^n are the north and east coordinates, respectively. $\mathbf{G}_{NE}$ is defined as

$$\mathbf{G}_{NE} := \begin{bmatrix} \mathbf{r}_1, & \mathbf{r}_2, & -\mathbf{R}_n^c\mathbf{r}_{nc}^n \end{bmatrix} \tag{5.14}$$

The depth z_t^c is left out of the calculation by normalizing the coordinates on the left side so that the down-component gets a value of 1. This concludes the direct georeferencing algorithm, which is summarized by the following steps that are applied whenever a new image is received:

1. Detect object(s) and find the pixel position (u, v), e.g. through image processing as described in Chapter 4.
2. Get UAV pose and gimbal angles from the navigation system.
3. Calculate $\mathbf{R}_n^c$ with equation (5.9)-(5.12) and find $\mathbf{G}_{NE}$ with (5.14).
4. Use (5.13) to calculate NE coordinates.

The georeferencing equations can be executed in real-time on a small embedded computer (as in Section 4.4.2), but the overall computation time is obviously limited by the complexity of the object detection algorithm in step 1.

5.3.1 Sensitivity analysis of georeferencing

The biggest pitfall in georeferencing is deviations between the true camera pose and the pose given by the navigation system. These errors are hard to detect during a mission without reference points. Consequently, it is necessary to be aware of the severity of this issue. This section seeks to illustrate how sensitive direct georeferencing is to errors in the camera pose, which is assumed to be caused by errors in the navigation estimates of the UAV.

The sensitivity analysis is carried out with three simplified simulation studies that are meant to illustrate level flight (zero roll, pitch and yaw), turn (roll angle of 25 degrees) and ascent (pitch angle of 20 degrees). All cases are simulated at an altitude of 100 and 400 meters. Moreover, the UAV has north and east positions of zero and the camera parameters are equal to the ones used in the case studies later in this chapter. They are based on the camera calibration from Section 2.5, and includes calibration of the focal lengths and the principal point. The object is assumed to be in the middle of the image. The displacement of the object, when navigation errors are added, is a measure of how sensitive the georeferencing algorithm is. Note that the magnitude of the georeferencing error also depends on the object position in the image so larger errors must be expected closer to the image boundaries. Therefore, the values shown in this section can be interpreted as the best-case scenario for different navigation errors since the georeferencing error increases further away from the image center.

Table 5.1 shows the error in the georeferenced position if a small bias is added to the UAV navigation states in level flight. Table 5.2 and Table 5.3 show the error during turn and ascent with the same errors, respectively. The systematic factors worth noticing are:

- An error in the down position (altitude) of the UAV leads to the same georeferencing error at both 100 and 400 meters. The georeferencing error due

Table 5.1: Error in georeferencing when there exist a bias in the UAV attitude and position for level flight

Offset	**Error with altitude 100m**	**Error with altitude 400m**
Roll offset 3°	5.3 m	21 m
Pitch offset 3°	5.3 m	21.2 m
Yaw offset 3°	0.5 m	1.7 m
North offset 10 m	10 m	10 m
East offset 10 m	10 m	10 m
Down offset 10 m	0.8 m	0.8 m
Roll and pitch offset 3°	7.5 m	30 m

Table 5.2: Error in georeferencing when there exist a bias in the UAV attitude and position during a turn (roll angle of 25 degrees)

Offset	**Error with altitude 100m**	**Error with altitude 400m**
Roll offset 3°	6.8 m	27 m
Pitch offset 3°	5.3 m	21.2 m
Yaw offset 3°	2.7 m	10.7 m
North offset 10 m	10 m	10 m
East offset 10 m	10 m	10 m
Down offset 10 m	5.1 m	5.1 m
Roll and pitch offset 3°	9 m	36 m

Table 5.3: Error in georeferencing when there exist a bias in the UAV attitude and position during ascent (pitch angle of 20 degrees)

Offset	**Error with altitude 100m**	**Error with altitude 400m**
Roll offset 3°	5.8 m	23 m
Pitch offset 3°	6.5 m	25.9 m
Yaw offset 3°	2.4 m	9.5 m
North offset 10 m	10 m	10 m
East offset 10 m	10 m	10 m
Down offset 10 m	4.5 m	4.5 m
Roll and pitch offset 3°	8.9 m	35.4 m

to altitude errors in level flight is nonzero because the principal point is calibrated to be a few pixels away from the image center.

- The georeferencing error caused by errors in the down position of the UAV is larger during a turn and ascent compared to level flight. This is expected as the ray from the camera center points almost directly towards the ground in level flight, but a longer distance during a turn or ascent.

- An error in the UAV north or east positions gives the exact same error in georeferencing.
- The georeferencing error is proportional to the altitude of the UAV when navigation errors in the attitude are considered. This is the most important factor to notice.
- Notice the magnitude of the georeferencing error at an altitude of 400 meters with a small bias in both roll and pitch (30 m to 36 m). This shows the necessity of having accurate knowledge about the camera orientation.

These factors illustrate the necessity of synchronizing the sensors and conduct a proper calibration since even small biases give large errors in georeferencing. It also reveals that attitude errors are the most significant error in georeferencing and should be minimized as much as possible.

5.3.2 Calibration of mounting misalignments

Camera calibration and time synchronization (described in Section 2.5) are key in georeferencing. Another error source is mounting misalignment errors, which can be even more critical. The previous section assumed that the gimbal frame {m} coincides with {b} when the gimbal has zero pan and tilt angles. In practice, it is hard to align these frames perfectly, which results in significant georeferencing errors as shown in the sensitivity analysis. This section derives a method for estimating the mounting misalignment. More specifically, the goal is to identify the rotation matrix between the body-fixed frame and the camera when the pan and tilt angles are zero. The procedure utilizes the georeferencing algorithm and is conducted after the gimbal and camera are mounted in the airframe. It is based on the method in [46].

The main requirement is to use a motion-capture system that provides information about the attitude and position of an object, relative to a fixed frame with high precision. For example OptiTrack [88] is a motion-capture system based on several cameras, and is able to give the attitude and position for a set of markers that form a rigid-body. Four markers are needed to define the body-fixed frame of the UAV (one on each wing, one in the front and one in the back). Moreover, a set of four markers is also mounted on a second object with a thermal signature that is recognized in the images (e.g. the thermal camera calibration surface).

The intention behind the procedure is to use the thermal camera to capture images of the second object. When the position and attitude of the camera are known (given by motion-capture system), it is possible to calculate where the object theoretically should be located by the georeferencing equations. This is compared with where the object is located according to the motion-capture system. Moreover, by using several images, an optimization problem is solved so that the rotation matrix between {b} and {c} is calculated based on where the object is and should be located. Note that the body frame the gimbal is aligned with through this method is given directly by the markers. The precision of the calibration is, therefore, limited by how accurate the markers are aligned with the true geometric body of the UAV.

The gimbal has two degrees of freedom (pan and tilt). In addition, it is beneficial to add a roll rotation because there can be mounting misalignment errors in all degrees of freedom. The theoretical NED positions for the second object (based on the pixel coordinates that are selected manually from the images) are calculated as

$$\frac{1}{z_t^c}\begin{bmatrix} x_t^n \\ y_t^n \\ 1 \end{bmatrix} = \mathbf{G}_{NE}^{-1}\mathbf{A}^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5.15}$$

The theoretical values $[x_t^n, y_t^n]^\top$ are compared with the values measured by the motion-capture system, and is formulated as an optimization problem with several known points in multiple images where the position and attitude of the camera and/or the objects vary. The optimization problem is stated as

$$J = \underset{\psi_{gb},\theta_{gb},\phi_{gb}}{\text{minimize}} \sum_{i=1}^{n_{\text{images}}} \sum_{k=1}^{n_{\text{points}}} f(i,k) \tag{5.16}$$

where ψ_{gb}, θ_{gb} and ϕ_{gb} are misalignments in pan, tilt and roll, respectively. The objective function is defined as

$$f(i,k) = ||\mathbf{p}_t^n(i,k) - \mathbf{p}_m^n(i,k)|| \tag{5.17}$$

and the subscript m and t are the measured (based on the motion-capture system) and theoretical (based on the position in the image) coordinates, respectively. Moreover, the subscripts i and k correspond to point k in image i. Three points in every image are needed because the optimization problem is formulated with respect to three unknown parameters. Optimization is conducted for several images to reduce the uncertainty a single image provides and assumed to mitigate the influence of noise. The problem is solved with the nonlinear least squares algorithm. Since misalignment errors are generally small, initialization is straightforward.

5.4 Target Velocity Reconstruction using Optical Flow

A challenge in remote sensing is to measure the velocity of different objects. Tracking filters can be used to estimate the velocity, but requires a minimum amount of position measurements before the velocity converges. In situations where only a few measurements are available, it is much harder to estimate the velocity and it would be desirable to measure it directly. Measuring the velocity of objects directly is not possible in a single image. Optical flow, which was defined in Section 4.2, can be used to detect moving objects and estimate their velocity, but is mostly used when the camera is stationary. This section looks into the relationship between target velocity and optical flow when the camera is mounted in a fixed-wing UAV. The definition of optical flow is repeated in the box below:

Definition 5.2 Optical flow is the displacement in pixels for a target detected in two consecutive images.

This section is motivated by the field of navigation where optical flow has been used to reconstruct the linear and angular velocities of fixed-wing UAVs [31, 51, 76], and is based on [44]. These works have used a monocular camera to reconstruct the body-fixed velocity of a UAV under the assumption that all detected features are located at rest on ground level. An interesting research question is to investigate if optical flow can be used to say something about target velocities if the velocity of the UAV is known. In other words, the opposite problem is considered where the UAV velocity is assumed known and features on the ground have unknown velocity.

To investigate this problem, it is necessary to derive the relationship between optical flow and velocity. Assume that a target at pixel position (u, v) is of interest and that the optical flow of the pixel is known. Differentiation of the pinhole camera model (5.5) yields

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{z_t^c} \begin{bmatrix} f & 0 & -f\frac{x_t^c}{z_t^c} \\ 0 & f & -f\frac{y_t}{z_t^c} \end{bmatrix} \begin{bmatrix} \dot{x}_t^c \\ \dot{y}_t^c \\ \dot{z}_t^c \end{bmatrix} \tag{5.18}$$

where $[\dot{u}, \dot{v}]^\top$ is the optical flow vector of the target. The vector $\dot{\mathbf{p}}_{ct}^c = [\dot{x}_t^c, \dot{y}_t^c, \dot{z}_t^c]^\top$ is the derivative of the camera-fixed coordinates of the target relative to {c}. It can be derived under the assumption that NED is inertial. Moreover, it is assumed that the target has zero angular velocity with respect to NED between two consecutive images.

$$\begin{aligned} \dot{\mathbf{p}}_{ct}^c &= \frac{\mathrm{d}}{\mathrm{dt}}\mathbf{p}_{ct}^c \\ &= \frac{\mathrm{d}}{\mathrm{dt}}(\mathbf{R}_n^c \mathbf{p}_{ct}^n) \\ &= \dot{\mathbf{R}}_n^c \mathbf{p}_{ct}^n + \mathbf{R}_n^c \dot{\mathbf{p}}_{ct}^n \\ &= \mathbf{R}_n^c \mathbf{S}(\boldsymbol{\omega}_{cn}^n)\mathbf{p}_{ct}^n + \mathbf{R}_n^c \mathbf{v}_{ct}^n \\ &= \mathbf{R}_n^c \mathbf{S}(\boldsymbol{\omega}_{cn}^n)\mathbf{R}_c^n \mathbf{p}_{ct}^c + \mathbf{v}_{ct}^c \\ &= \mathbf{S}(\boldsymbol{\omega}_{cn}^c)\mathbf{p}_{ct}^c + \mathbf{v}_{ct}^c \\ &= \boldsymbol{\omega}_{cn}^c \times \mathbf{p}_{ct}^c + \mathbf{v}_{ct}^c \\ &\approx \boldsymbol{\omega}_{ct}^c \times \mathbf{p}_{ct}^c + \mathbf{v}_{ct}^c \end{aligned} \tag{5.19}$$

where $\mathbf{v}_{ct}^c$ and $\boldsymbol{\omega}_{ct}^c$ are the linear and angular velocities of the target with respect to {c} decomposed in {c}, respectively. $\mathbf{p}_{ct}^c = [x_t^c, y_t^c, z_t^c]^\top$ is the position of the target decomposed in {c}. That is the lever arm in (5.19) when it is assumed that {b} and {c} coincides. Zero angular velocity for the target between consecutive images is a reasonable assumption as long as images are captured closely in time. Moreover, the target should not change orientation quickly (compared to the UAV) and that is rarely the case for boats and other floating objects.

Equation (5.18) might be rewritten by inserting the final expression from (5.19):

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \frac{1}{z_t^c}\left[\mathbf{B} \;\mid\; \mathbf{B}\right]\begin{bmatrix}\mathbf{v}_{ct}^c\\ \boldsymbol{\omega}_{ct}^c \times \mathbf{p}_{ct}^c\end{bmatrix} \tag{5.20}$$

$$\mathbf{B} = \begin{bmatrix} f & 0 & -f\frac{x_t^c}{z_t^c}\\ 0 & f & -f\frac{y_t^c}{z_t^c}\end{bmatrix}$$

By the properties of the crossproduct, the relationship between optical flow and the linear and angular velocities of the target with respect to {c} is

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \underbrace{\frac{1}{z_t^c}\left[\mathbf{B} \;\mid\; -\mathbf{B}\cdot\mathbf{S}(\mathbf{p}_{ct}^c)\right]}_{\mathbf{M}(\mathbf{p}_{ct}^c)}\begin{bmatrix}\mathbf{v}_{ct}^c\\ \boldsymbol{\omega}_{ct}^c\end{bmatrix} \tag{5.21}$$

where

$$\mathbf{M}(\mathbf{p}_{ct}^c) = \frac{1}{z_t^c}\cdot\begin{bmatrix} f, & 0, & -f\frac{x_t^c}{z_t^c}, & -f\frac{x_t^c}{z_t^c}y_t^c, & fz_t^c + f\frac{x_t^c}{z_t^c}x_t^c, & -fy_t^c\\ 0, & f, & -f\frac{y_t^c}{z_t^c}, & -fz_t^c - f\frac{y_t^c}{z_t^c}y_t^c, & f\frac{y_t^c}{z_t^c}x_t^c, & fx_t^c\end{bmatrix} \tag{5.22}$$

The velocities in (5.21) depend on the camera and UAV motions, and the target motion. The target motion with respect to {c} is not informative when the camera moves. Thus, it is more desirable to use the target velocity with respect to NED. Equation (5.21) can be rewritten as

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \mathbf{M}(\mathbf{p}_{ct}^c)\begin{bmatrix}\mathbf{v}_{cn}^c + \mathbf{v}_{nt}^c\\ \boldsymbol{\omega}_{cn}^c + \boldsymbol{\omega}_{nt}^c\end{bmatrix} = \mathbf{M}(\mathbf{p}_{ct}^c)\begin{bmatrix}\mathbf{v}_{cn}^c + \mathbf{v}_{nt}^c\\ \boldsymbol{\omega}_{cn}^c\end{bmatrix} \tag{5.23}$$

where it is utilized that the target has zero angular velocity with respect to NED.

If the velocities of the UAV decomposed in {b} are known (from a navigation system), it is possible to approximate the optical flow caused by $\mathbf{v}_{cn}^c$ and $\boldsymbol{\omega}_{cn}^c$. It will from now on be referred to as the optical flow caused by camera motion and denoted $[\dot{u}_c, \dot{v}_c]^\top$. It is defined as

$$\begin{bmatrix}\dot{u}_c\\ \dot{v}_c\end{bmatrix} := \mathbf{M}(\mathbf{p}_{ct}^c)\begin{bmatrix}\mathbf{v}_{cn}^c\\ \boldsymbol{\omega}_{cn}^c\end{bmatrix} \tag{5.24}$$

where $\mathbf{v}_{cn}^c$ and $\boldsymbol{\omega}_{cn}^c$ are the linear and angular velocities of NED with respect to {c} decomposed in {c}. Since the origin of {c} coincides with {b}, both {b} and {c} have the same linear velocity with respect to {n}. Therefore, the linear velocity is

$$\mathbf{v}_{cn}^c = \mathbf{R}_b^c\mathbf{v}_{cn}^b = \mathbf{R}_b^c\mathbf{v}_{bn}^b = -\mathbf{R}_b^c\mathbf{v}_{nb}^b \tag{5.25}$$

where $\mathbf{v}_{nb}^b$ is the known body-fixed linear velocity of the UAV with respect to {n} decomposed in {b}. The angular velocity is

$$\begin{aligned}\boldsymbol{\omega}_{cn}^c &= \boldsymbol{\omega}_{bn}^c + \boldsymbol{\omega}_{cb}^c\\ &= \mathbf{R}_b^c(\boldsymbol{\omega}_{bn}^b + \boldsymbol{\omega}_{cb}^b)\\ &= -\mathbf{R}_b^c(\boldsymbol{\omega}_{nb}^b + \boldsymbol{\omega}_{bc}^b)\end{aligned} \tag{5.26}$$

where $\boldsymbol{\omega}_{nb}^{b}$ is the known angular velocity of the UAV with respect to {n} decomposed in {b}. $\boldsymbol{\omega}_{bc}^{b}$ is the angular velocity of {c} with respect to {b} decomposed in {b}. It is given by the gimbal motion. A pan/tilt gimbal can only rotate about the body z and y axes. Thus, $\boldsymbol{\omega}_{bc}^{b}$ can be approximated as [24]

$$\begin{aligned} \boldsymbol{\omega}_{bc}^{b} &\approx \boldsymbol{\omega}_{z}(\dot{\psi}_{gb}) + \mathbf{R}_{z}(\psi_{gb})\boldsymbol{\omega}_{y}(\dot{\theta}_{gb}) \\ &= \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{gb} \end{bmatrix} + \mathbf{R}_{z}(\psi_{gb}) \begin{bmatrix} 0 \\ \dot{\theta}_{gb} \\ 0 \end{bmatrix} \end{aligned} \tag{5.27}$$

where $\dot{\psi}_{gb}$ and $\dot{\theta}_{gb}$ are the derivatives of the pan and tilt angles, respectively. They need to be measured or approximated by e.g. a Taylor-series approximation. If the gimbal has a fixed orientation, $\boldsymbol{\omega}_{bc}^{b}$ is zero.

Note that (5.25) and (5.26) are approximations from a practical point of view if the gimbal moves between two consecutive images. This is because $\mathbf{R}_{b}^{c}$ varies between two images in such a case. The relationship holds in continuous time, but is an approximation when the sampling rate is limited. Thus, gimbal motion should be avoided or small with respect to the frame rate of the camera. In practice, $\mathbf{R}_{b}^{c}$ is approximated with the mean orientation in the time period between consecutive images. Increased accuracy is achieved when the gimbal has a static orientation with respect to the UAV.

The optical flow caused by camera motion, (5.24), is calculated with (5.25) and (5.26). The optical flow caused by the camera motion is inserted into (5.23) where $\dot{u}_m$ and $\dot{v}_m$ now are defined as the measured optical flow, obtained through image processing (e.g. the SIFT feature detector discussed in Section 4.2). Consequently,

$$\begin{bmatrix} \dot{u}_m \\ \dot{v}_m \end{bmatrix} = \mathbf{M}(\mathbf{p}_{ct}^{c}) \begin{bmatrix} \mathbf{v}_{nt}^{c} \\ \mathbf{0} \end{bmatrix} + \mathbf{M}(\mathbf{p}_{ct}^{c}) \begin{bmatrix} \mathbf{v}_{cn}^{c} \\ \boldsymbol{\omega}_{cn}^{c} \end{bmatrix} \tag{5.28}$$

where the second term on the right hand side is recognized as the optical flow caused by camera motion. Thus, it is possible to rewrite (5.28) as

$$\begin{aligned} \begin{bmatrix} \dot{u}_m - \dot{u}_c \\ \dot{v}_m - \dot{v}_c \end{bmatrix} &= \mathbf{M}(\mathbf{p}_{ct}^{c}) \begin{bmatrix} \mathbf{R}_{n}^{c}\mathbf{v}_{nt}^{n} \\ \mathbf{0} \end{bmatrix} \\ &= \frac{f}{z_t^c} \begin{bmatrix} 1 & 0 & -\frac{1}{z_t^c}x_t^c \\ 0 & 1 & -\frac{1}{z_t^c}y_t^c \end{bmatrix} \mathbf{R}_{n}^{c}\mathbf{v}_{nt}^{n} \end{aligned} \tag{5.29}$$

where the gimbal motion should be small compared to the frame rate of the camera in a practical implementation as discussed above.

Equation (5.29) only has two terms on the left side and three unknown target velocity parameters on the right side. Nevertheless, since the main motivation is to locate surface objects at sea, the down velocity of objects on the sea surface is expected to be zero. Hence, the third column of $\mathbf{R}_{n}^{c}$ can be discarded in (5.29) and the NE velocities (v_t^N and v_t^E) of the target is calculated as

$$\begin{bmatrix} v_t^N \\ v_t^E \end{bmatrix} = \left(\frac{f}{z_t^c} \begin{bmatrix} 1 & 0 & -\frac{1}{z_t^c} x_t^c \\ 0 & 1 & -\frac{1}{z_t^c} y_t^c \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \dot{u}_m - \dot{u}_c \\ \dot{v}_m - \dot{v}_c \end{bmatrix} \tag{5.30}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first and second column vectors in the rotation matrix $\mathbf{R}_n^c$.

Reconstructing the target velocity is very challenging in practice. Ideally, the gimbal should maintain a fixed orientation with respect to the UAV between images. This increases the accuracy of (5.24). The UAV motion typically exceeds the target motion significantly so inaccuracies in (5.24) can potentially dominate the optical flow caused by target motion in (5.29). Moreover, the assumption of zero angular velocity for the target with respect to NED is an approximation. In addition, the camera-fixed coordinates of the target must be calculated with georeferencing and inserted in the matrix $\mathbf{M}(\mathbf{p}_{ct}^c)$. In other words, there are numerous error sources in a practical implementation. Nevertheless, if the gimbal moves slowly between images and the frame rate of the camera is large, it is possible to approximate the target velocity using two images. This is investigated experimentally in Section 5.6.3.

5.5 Nonlinear Observers for UAV Navigation

Real-time georeferencing relies on accurate information about the UAV pose as stressed several times in this chapter. The camera pose depends on the navigation states of the UAV and this makes UAV navigation an important topic in this chapter. This section describes a nonlinear observer that utilizes GNSS and inertial sensors to estimate the position, attitude and velocity of a UAV. The nonlinear observer is used in georeferencing and is a vital part of the system. Navigation is also discussed in Section 7.3.

The attitude dynamics are in general nonlinear and all degrees of freedom need to be considered. Navigation filters can be probabilistic, such as the extended Kalman filter (EKF) or the Multiplicative EKF (MEKF) [106], or deterministic such as nonlinear observers [26, 31, 36, 51, 73]. The main advantages with nonlinear observers are reduced computational complexity, easy implementation, and the possibility of proving global stability for the error dynamics through Lyapunov analysis since linearization is avoided in contrast to EKF-based techniques. The main advantage with EKF-based designs is that the state uncertainties are available directly through the covariance matrix. A nonlinear observer is used for navigation in this section and an EKF based technique is used in Chapter 7.

A state of the art nonlinear observer is presented this section. The structure of the observer was originally presented in [36]. The translational motion observer is based on measurements from three accelerometers and two RTK-GNSS receivers. The attitude observer utilizes angular rate measurements and reference vectors that are extracted from three accelerometers, and pitch and yaw measurements from two RTK-GNSS receivers with separate antennas [107]. The observer design is based on a kinematic model that is assumed to represent the true relationship between the states.

5.5.1 Strapdown equations

The states of interest in the navigation system are the position ($\mathbf{p}$), linear velocity ($\mathbf{v}$) and attitude of the body frame relative to the Earth-fixed-Earth-Centered coordinate frame. The two former are decomposed in {e}. A review of methods for attitude estimation is presented in [18]. In this section, attitude is parametrized as a unit quaternion ($\boldsymbol{q}_b^e$) and a rotation matrix between ECEF and the body-fixed frame of the UAV, denoted $\mathbf{R}_b^e$.

The kinematics (strapdown equations) are used to design the observer equations, and described mathematically as

$$\dot{\boldsymbol{p}}_{eb}^e = \boldsymbol{v}_{eb}^e \tag{5.31}$$

$$\dot{\boldsymbol{v}}_{eb}^e = -2\boldsymbol{S}(\boldsymbol{\omega}_{ie}^e)\boldsymbol{v}_{eb}^e + \boldsymbol{R}_b^e\boldsymbol{f}_{ib}^b + \boldsymbol{g}_b^e \tag{5.32}$$

$$\dot{\boldsymbol{q}}_b^e = \frac{1}{2}\boldsymbol{q}_b^e \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ib}^b \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ie}^e \end{bmatrix} \otimes \boldsymbol{q}_b^e \tag{5.33}$$

with

$$\boldsymbol{\omega}_{ie}^e = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_{ie}, \tag{5.34}$$

where ω_{ie} is the Earth's rotation rate and $\boldsymbol{g}_b^e$ is the gravity vector. The nonlinear observer is structurally the same as in [36].

5.5.2 Attitude observer

The nonlinear observer for estimating the attitude between $\{b\}$ and $\{e\}$ is given similar to [36],

$$\Sigma_1 : \begin{cases} \dot{\hat{\boldsymbol{q}}}_b^e = \frac{1}{2}\hat{\boldsymbol{q}}_b^e \otimes \begin{bmatrix} 0 \\ \hat{\boldsymbol{\omega}}_{ib}^b \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ie}^e \end{bmatrix} \otimes \hat{\boldsymbol{q}}_b^e, & (5.35a) \\ \hat{\boldsymbol{\omega}}_{ib}^b = \boldsymbol{\omega}_{\mathrm{IMU}}^b - \hat{\boldsymbol{b}}_{\mathrm{ars}}^b + \hat{\boldsymbol{\sigma}}_{ib}^b, & (5.35b) \\ \dot{\hat{\boldsymbol{b}}}_{\mathrm{ars}}^b = \mathrm{Proj}\left(\hat{\boldsymbol{b}}_{\mathrm{ars}}^b, -k_I\hat{\boldsymbol{\sigma}}_{ib}^b\right), & (5.35c) \end{cases}$$

where $\hat{\boldsymbol{b}}_{\mathrm{ars}}^b$ is the estimate of the angular rate sensor (ARS) bias. Proj($\star,\star$) denotes the ARS bias projection algorithm ensuring that $\|\hat{\boldsymbol{b}}_{\mathrm{ars}}^b\|_2 \leq M_{\hat{b}_{\mathrm{ars}}}$ for $M_{\hat{b}_{\mathrm{ars}}} > M_{b_{\mathrm{ars}}}$

[35], and k_I is the gain associated with the ARS bias estimation. Moreover, the nonlinear injection term, $\hat{\boldsymbol{\sigma}}^b_{ib}$, is given as

$$\hat{\boldsymbol{\sigma}}^b_{ib} = k_1 \underline{\boldsymbol{v}}^b_1 \times \boldsymbol{R}^\intercal(\hat{\boldsymbol{q}}^e_b)\underline{\boldsymbol{v}}^e_1 + k_2 \underline{\boldsymbol{v}}^b_2 \times \boldsymbol{R}^\intercal(\hat{\boldsymbol{q}}^e_b)\underline{\boldsymbol{v}}^e_2, \tag{5.36}$$

where the measurement vectors $\underline{\boldsymbol{v}}^b_{1,2}$ and reference vectors $\underline{\boldsymbol{v}}^e_{1,2}$ are calculated using

$$\underline{\boldsymbol{v}}^b_1 = \underline{\boldsymbol{f}}^b, \qquad \underline{\boldsymbol{v}}^e_1 = \underline{\boldsymbol{f}}^e, \tag{5.37}$$

$$\underline{\boldsymbol{v}}^b_2 = \underline{\boldsymbol{f}}^b \times \underline{\boldsymbol{p}}^b, \quad \underline{\boldsymbol{v}}^e_2 = \underline{\boldsymbol{f}}^e \times \underline{\boldsymbol{p}}^e. \tag{5.38}$$

Furthermore, the measurement and corresponding reference vector pairs in (5.37)–(5.38) are constructed as

$$\underline{\boldsymbol{f}}^b = \frac{\boldsymbol{f}^b_{\text{IMU}}}{\|\boldsymbol{f}^b_{\text{IMU}}\|_2}, \qquad \underline{\boldsymbol{f}}^e = \frac{\text{sat}_{M_f}(\hat{\boldsymbol{f}}^e_{ib})}{\|\text{sat}_{M_f}(\hat{\boldsymbol{f}}^e_{ib})\|_2}, \tag{5.39}$$

$$\underline{\boldsymbol{p}}^b = \frac{\boldsymbol{p}^b_{ba_1,1} - \boldsymbol{p}^b_{ba_1,0}}{\|\boldsymbol{p}^b_{ba_1,1} - \boldsymbol{p}^b_{ba_1,0}\|_2} \quad \underline{\boldsymbol{p}}^e = \frac{\boldsymbol{p}^e_{\text{RTK},1} - \boldsymbol{p}^b_{\text{RTK},0}}{\|\boldsymbol{p}^e_{\text{RTK},1} - \boldsymbol{p}^b_{\text{RTK},0}\|_2}, \tag{5.40}$$

where $\boldsymbol{p}^b_{ba_k,k}$ is the relative vector between the two antenna positions given in body. $\hat{\boldsymbol{f}}^e_{ib}$ is the estimated specific force, provided by the TMO, presented next. $\mathbf{p}^b_{RTK,0}$ and $\mathbf{p}^b_{RTK,1}$ are the position of the GNSS antennas given in {b}.

5.5.3 Translational motion observer

The translational motion observer (TMO) is similar to that of [36], and given as follows:

$$\Sigma_2 : \begin{cases} \dot{\hat{\boldsymbol{p}}}^e_{eb} = \hat{\boldsymbol{v}}^e_{eb} + \vartheta \boldsymbol{K}^0_{pp}\tilde{\boldsymbol{y}}^e_{eb} & \text{(5.41a)} \\ \dot{\hat{\boldsymbol{v}}}^e_{eb} = -2\boldsymbol{S}(\boldsymbol{\omega}^e_{ie})\hat{\boldsymbol{v}}^e_{eb} + \hat{\boldsymbol{f}}^e_{ib} + \quad \boldsymbol{g}^e_b(\hat{\boldsymbol{p}}^e_{eb}) + \vartheta^2 \boldsymbol{K}^0_{vp}\tilde{\boldsymbol{y}}^e_{eb} & \text{(5.41b)} \\ \dot{\hat{\boldsymbol{\xi}}}^e_{ib} = -\boldsymbol{R}(\hat{\boldsymbol{q}}^e_b)\boldsymbol{S}(\hat{\boldsymbol{\sigma}}^b_{ib})\boldsymbol{f}^b_{\text{IMU}} + \vartheta^3 \boldsymbol{K}^0_{\xi p}\tilde{\boldsymbol{y}}^e_{eb} & \text{(5.41c)} \\ \hat{\boldsymbol{f}}^e_{ib} = \boldsymbol{R}(\hat{\boldsymbol{q}}^e_b)\boldsymbol{f}^b_{\text{IMU}} + \hat{\boldsymbol{\xi}}^e_{ib}, & \text{(5.41d)} \end{cases}$$

where

$$\tilde{\boldsymbol{y}}^e_{eb} = \boldsymbol{p}^e_{RTK,k} - \hat{\boldsymbol{p}}^e_{eb} \tag{5.42}$$

, where $k \in [0, 1]$, and $\boldsymbol{K}_\star$ are gains associated with the RTK position measurement. $\boldsymbol{\xi}^e_{ib}$ is an auxiliary state used to estimate $\boldsymbol{f}^e_{ib}$. ϑ is a high-gain like parameter used to guarantee stability. Furthermore, by noting the linear time-varying (LTV) structure of (5.41) and defining

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{p}^e_{eb} & \boldsymbol{v}^e_{eb} & \boldsymbol{\xi}^e_{ib} \end{bmatrix}^\top, \tag{5.43}$$

the TMO can be written on LTV form as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}(t)\boldsymbol{u} + \boldsymbol{D}(t, \hat{\boldsymbol{x}}) + \boldsymbol{K}(t)(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}), \tag{5.44}$$

with the system matrices,

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \boldsymbol{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \boldsymbol{B}(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \boldsymbol{R}(\hat{\boldsymbol{q}}_b^e) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{R}(\hat{\boldsymbol{q}}_b^e) \end{bmatrix}, \tag{5.45}$$

the measurement matrix,

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}, \tag{5.46}$$

the vector,

$$\boldsymbol{D}(t, \hat{\boldsymbol{x}}) = \begin{bmatrix} \mathbf{0}_{3\times1} & -2\boldsymbol{S}(\boldsymbol{\omega}_{ie}^e)\hat{\boldsymbol{v}}_{eb}^e + \boldsymbol{g}_b^e(\hat{\boldsymbol{p}}_{eb}^e) & \mathbf{0}_{3\times1} \end{bmatrix}^\top, \tag{5.47}$$

and the gain matrix,

$$\boldsymbol{K}(t) = \begin{bmatrix} \boldsymbol{K}_{pp} \\ \boldsymbol{K}_{vp} \\ \boldsymbol{K}_{\xi p} \end{bmatrix} = \begin{bmatrix} \vartheta \boldsymbol{K}_{pp}^0 \\ \vartheta^2 \boldsymbol{K}_{vp}^0 \\ \vartheta^3 \boldsymbol{K}_{\xi p}^0 \end{bmatrix}, \tag{5.48}$$

where

$$\boldsymbol{K}^0(t) = \begin{bmatrix} (\boldsymbol{K}_{pp}^0)^\intercal & (\boldsymbol{K}_{vp}^0)^\intercal & (\boldsymbol{K}_{\xi p}^0)^\intercal \end{bmatrix}^\intercal \tag{5.49}$$

is given as $\boldsymbol{K}^0(t) = \boldsymbol{P}(t)\boldsymbol{C}^\intercal \boldsymbol{R}^{-1}(t)$, with $\boldsymbol{P}(t) = \boldsymbol{P}^\intercal(t) > 0$ being the solution of the time-scaled Riccati equation

$$\begin{aligned} \frac{1}{\vartheta}\dot{\boldsymbol{P}}(t) = \boldsymbol{A}\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{A}^\intercal - \boldsymbol{P}\boldsymbol{C}^\intercal \boldsymbol{R}^{-1}(t)\boldsymbol{C}^\intercal \boldsymbol{P}(t) \\ + \boldsymbol{B}(\hat{\boldsymbol{q}}_b^e)\boldsymbol{Q}(t)\boldsymbol{B}^\intercal(\hat{\boldsymbol{q}}_b^e). \end{aligned} \tag{5.50}$$

Finally, the input is given as

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{f}_{\text{IMU}}^b & -\boldsymbol{S}(\hat{\boldsymbol{\sigma}}_{ib}^b)\boldsymbol{f}_{\text{IMU}}^b \end{bmatrix}^\top \tag{5.51}$$

This results in the origin of the error dynamics being semi-globally exponentially stable. See [36] for details on the stability analysis. In addition, a given lever arm from the IMU to any of the GPS antennas may be incorporated into the TMO by replacing (5.42) with

$$\tilde{\boldsymbol{y}}_{eb,k}^e = \boldsymbol{p}_{RTK,k}^e - \hat{\boldsymbol{p}}_{eb}^e - \mathbf{R}(\boldsymbol{q}_b^e)\mathbf{r}_b^b, \tag{5.52}$$

where $\mathbf{r}_b^b$ represents the lever arm and $k \in [0, 1]$ is the index of the RTK position solution.

The nonlinear observer represents position in the Earth-Centered-Earth-Fixed (ECEF) coordinate frame since GPS measurements are given in ECEF. The georeferencing equations are represented in {n} so a transformation between ECEF and {n} is needed. ECEF coordinates are transformed into longitude (l), latitude(μ) and height (h) [114] by using the WGS-84 reference ellipsoid. Local NED coordinates are acquired from l, μ and h [27, 29]. The origin of {n} is placed at a user-defined (l_0, μ_0, h_0). Note that a local flat earth assumption is present when using NED coordinates and is only valid in proximity of (l_0, μ_0, h_0) because a tangent plane is used when defining {n}.

5.6 Case Studies

This section presents three independent case studies covering the different topics discussed in this chapter. Case study one in Section 5.6.1 investigates georeferencing of a static point in the environment where the focus is to show the effect of time synchronization and misalignment calibration. Section 5.6.2 presents the second case study where georeferencing is used to determine the position of a slowly moving marine vessel. The third case study is presented in Section 5.6.3 and covers target velocity reconstruction using optical flow. All of the results are based on experimental data gathered in three different flight experiments. Case studies one and two are based on the work in [46] and the third case study is based on [42].

5.6.1 Case study 1 - Georeferencing of static point

The first case study is based on data gathered at a flight experiment carried out near Agdenes, in the outer Trondheim fjord in Norway. The goal was to collect data for georeferencing of a stationary base station GPS antenna located two meters above sea level. The GPS receiver connected to the antenna had RTK capability, which means that the true position of the antenna is known with centimeter accuracy. The antenna was chosen as target to have a static object with known ground truth position. Nearly 8000 images were captured, and the flight experiment lasted for approximately 25 min. The GPS antenna was in the field of view and marked manually in 552 images. An uncertainty of a few pixels was potentially added during this process, but the mean error should be close to zero. The data have been processed after the flight, but georeferencing has also been conducted in real-time on the on-board computer more recently [79]. The sensor suite described in Section 2.3 and Appendix A was used to collect data:

The marked pixel positions were used together with the UAV pose estimated by the nonlinear observer presented in Section 5.5 to acquire NED positions of the antenna. The autopilot navigation data were also used in georeferencing for comparison. Both the autopilot and the nonlinear observer were calibrated for misalignment errors as described in Section 5.3.2. Consequently, the difference experienced in georeferencing is mainly caused by the difference in sampling rate (250 Hz for the observer and 10 Hz for the autopilot) and the quality of the sensors. Moreover, the autopilot and the nonlinear observer were synchronized with the camera through SenTiBoard as described in Section 2.5.2.

Navigation results

The path of the UAV is displayed in Figure 5.2, and the local NED frame is placed so that the target position is in the origin. The antenna was only in the field of view in a fixed part of the loiter motion near the origin so it was observed

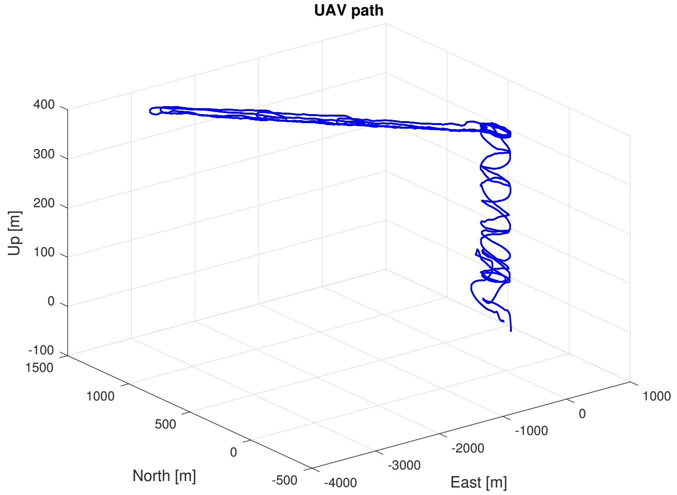
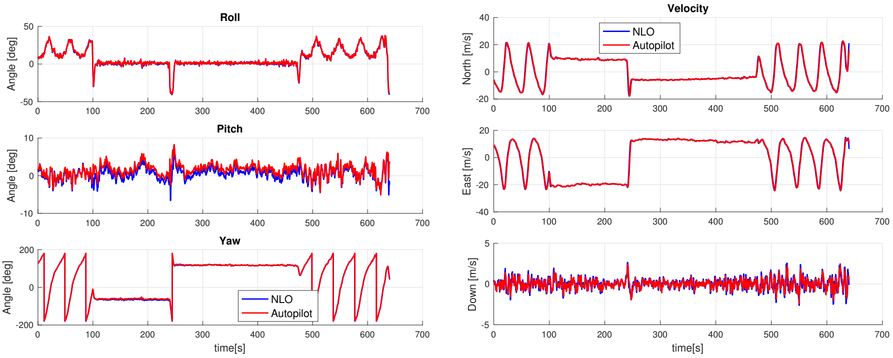


Figure 5.2: The UAV path estimated by the observer

from approximately the same attitude and altitude for each set of measurements. The UAV operated at an altitude of 350 m above sea level. Operating at greater altitudes increases the ground coverage of a single pixel and reduces the precision of georeferencing. A single pixel covers a square with sides equal to approximately $(8.95 \times 10^{-4} \times$ altitude$)$ m for the lens used in the experiment. Moreover, because the georeferencing error, as a consequence of errors in the camera orientation, is proportional to the altitude of the UAV, larger errors must be expected when the altitude is increased.



(a) (b)

Figure 5.3: The UAV attitude and velocity estimated by the nonlinear observer (blue) and autopilot (red)

Figure 5.3(a) shows the UAV attitude estimated by the nonlinear observer compared with the estimates from the autopilot. Only the time span from the first to the last antenna observation is shown. There are only minor differences between the autopilot and the observer, which indicate that both provide accurate estimates. Figure 5.3(b) shows the NED velocities, which are also similar. A true reference is not available for the velocity and attitude. Nevertheless, since the observer and autopilot used different GPS-receivers and IMUs, the estimates from both the autopilot and the observer are more credible when they are comparable.

Mounting misalignments results

The mounting misalignment of the gimbal was calculated to be (based on the method in Section 5.3.2)

$$\phi_{gb} = -1.7^o, \quad \theta_{gb} = 3.9^o, \quad \psi_{gb} = 1.9^o$$

Challenges related to sensor synchronization were avoided in the calibration process since the camera was stationary during image capture. The calibration process was verified by projecting the calibration points from the image plane back into NED and comparing that with the true position. The mean reprojection error was 0.0229 m and the calibration was conducted at a range of approximately 1.2 m. The remaining error is a combination of errors in the marked pixel position, the camera intrinsic matrix and the camera pose (the motion-capture system is not perfect).

If it is assumed that the remaining reprojection error is caused by errors in the orientation of the camera, the error grows to 6.6 m when the range is increased to 350 m, which is the operating altitude in this case study. This is only a rough indication for the error one can expect in the experimental data as the error is caused by other sources as well, which are not proportional to the altitude.

Although the gimbal misalignment has been identified, a misalignment between the body frame as defined by the IMU, and the body frame defined by the geometry of the UAV (which is the body frame used in the gimbal calibration) can be present. IMU misalignment was detected by comparing the Euler angles computed by the navigation system with zero when the UAV was leveled by a leveler. IMU calibration was also conducted, giving an offset in roll, pitch and yaw of 1.1^o, -5.7^o and 0.25^o, respectively. These angles are obviously equally important for the accuracy of georeferencing.

Georeferencing results

Figure 5.4(a) shows the georeferenced positions of the antenna using the observer, while Figure 5.4(b) shows the georeferenced positions using the autopilot. The measurements are centered around the true position, and distributed in a way which is reminiscent of a Gaussian distribution without a bias. This is useful in target tracking where several measurements are used to estimate the target states.

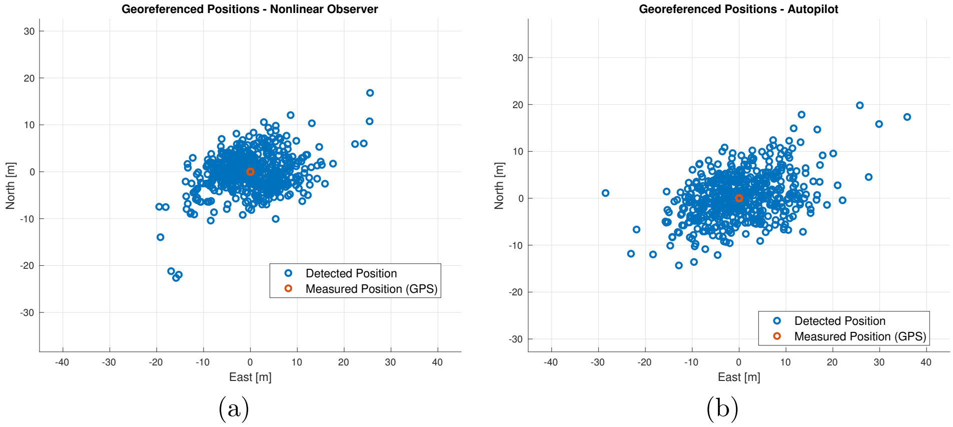


Figure 5.4: a) The georeferenced antenna position together with the true position. The UAV pose is extracted from the nonlinear observer (a) and the Pixhawk/Ardu-Plane autopilot (b).

Table 5.4: Main results of experiment 1. SD = standard deviation, MAE = mean absolute error

	Observer	**Pixhawk Autopilot**	**Reference**
Mean north position	−0.18 m	0.29 m	0 m
Mean east position	−0.12 m	0.09 m	0 m
SD north position	4.10 m	4.83 m	-
SD east position	6.47 m	7.84 m	-
MAE position	6.40 m	7.70 m	-

Table 5.4 summarizes the main results using the observer and autopilot navigation data. The performance is comparable, but the standard deviation (SD) of the measurements is 4.10 m and 6.47 m for the observer, and 4.83 m and 7.84 m for the autopilot. The mean georeferenced position for all images using the observer is −0.18 m and −0.12 m in north and east, respectively. Since the true position is in the origin, the exact location (with centimeter accuracy) of the GPS antenna is known when all measurements are used. The mean north and east positions using the autopilot are 0.29 m and 0.09 m, respectively. The small difference in mean position indicates that both the observer and autopilot are competitive. Moreover, it shows that the quality of the sensors is less important than proper misalignment calibration and time synchronization.

The mean absolute error (MAE) for the georeferenced position using a single image is 6.40 m with the observer (7.70 m for the autopilot) at an operating altitude of 350 m. This is a significant result as the accuracy is comparable to previous work achieved at lower altitudes [65]. In fact, the error is also slightly lower than what was expected from the remaining reprojection error after the calibration process (6.6 m). Nevertheless, a part of the error is related to factors such as errors in the

UAV pose and in the detection process. Moreover, the calibration process identified the misalignment error for the gimbal, but the attitude of the UAV was assumed known through the motion-capture system. In a field experiment, the UAV pose is also uncertain and that can both increase and decrease the error.

Figures 5.5(a) and 5.5(b) show the corresponding georeferencing results without time synchronization of the camera (the image timestamps are assigned directly by the on-board computer) and without IMU misalignment calibration, respectively. Moreover, these results are summarized in Table 5.5. The MAE in a single image is 11.10 m without time synchronization compared to 6.40 m with time synchronization. The results without IMU misalignment calibration are dreadful in comparison. The MAE in a single image is 40.89 m, which is explained by the mean position being −38.16 m and 9.28 m from the true position in north and east, respectively. Moreover, the distribution does not resemble a normally distributed variable. The banana shape of the georeferenced points indicates that an error in the attitude is present since this is the expected shape during a turn if there is a bias in roll or pitch. This is clearly a significant reduction in performance even though the misalignment of the IMU was only 1.1° in roll, −5.7° in pitch and 0.25° in yaw. These results support the need for a proper and reliable misalignment calibration and time synchronization.

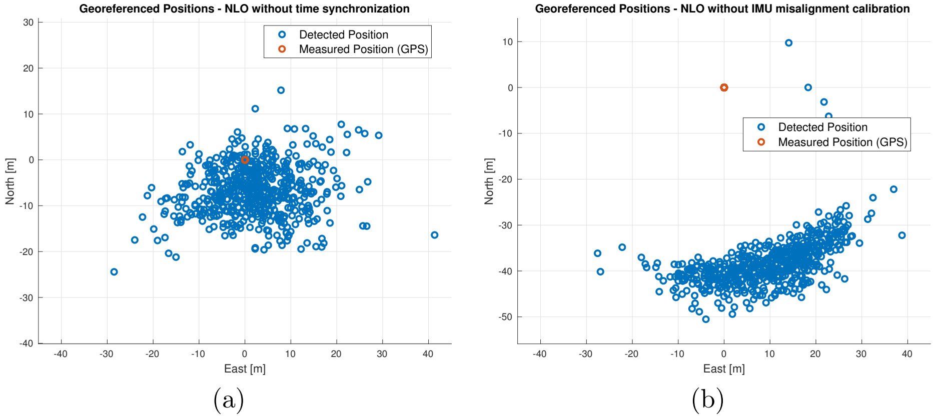


Figure 5.5: The georeferenced antenna position together with the true position without time synchronization (a) and misalignment calibration (b)

Figure 5.6 shows the measurement and estimation error when the georeferenced measurements (using the observer) are filtered in a Kalman filter with a motion model corresponding to an object at rest (zero velocity). The estimation error is less than a meter after the first batch of measurements. Moreover, the estimation error in the end is less than 0.5 m. In practice, this confirms that it is possible to track slowly moving objects accurately with georeferencing. The largest measurement error occurs in the end where a single measurement has an error just surpassing 30 m. The overall best measurement has an absolute error below 0.3 m.

Table 5.5: Main results of experiment 1 with and without time synchronization and IMU misalignment calibration. SD = standard deviation, MAE = mean absolute error

	Observer	**Without time synchronization**	**Without misalignment calibration**
Mean north position	−0.18 m	−6.67 m	−38.16 m
Mean east position	−0.12 m	2.04 m	9.28 m
SD north position	4.10 m	5.64 m	5.40 m
SD east position	6.47 m	8.96 m	10.63 m
MAE position	6.40 m	11.10 m	40.89 m

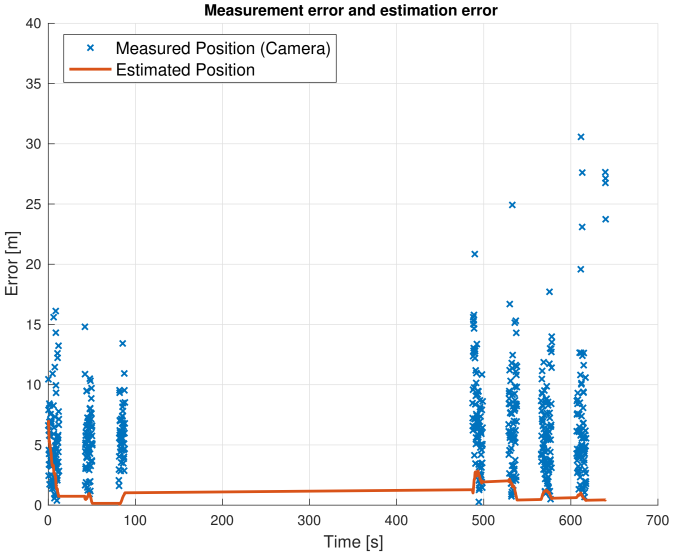


Figure 5.6: Measurement and estimation error in position when a Kalman filter is used to filter the georeferenced measurements of the antenna

Figure 5.7 shows the estimated autocorrelation for the innovation process (defined in Section 3.2) in the north and east positions. The autocorrelation shows that the innovation process is correlated in both north and east, which means that the innovation in the previous measurement is correlated with the innovation in the current measurement. From a practical point of view, this is expected because consecutive images are captured from almost the same UAV pose. Therefore, a systematic error in the camera orientation in one image is most likely present at the time when the next image is captured, and thus the georeferencing error is correlated for consecutive images.

In field experiments, it is not realistic to expect that the innovations are white.

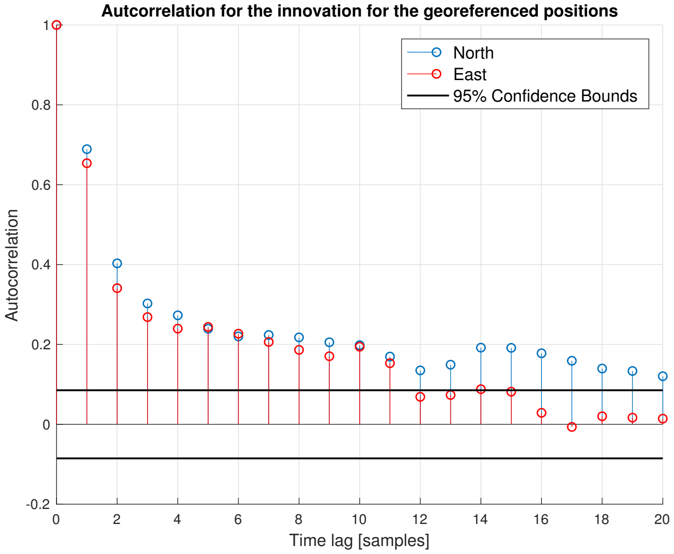


Figure 5.7: Autocorrelation for the innovations in the north and east georeferenced positions

Moreover, since navigation errors are the most significant challenge, the errors in georeferencing depend on the output of a navigation filter. The estimation error in the navigation filter is often correlated for two consecutive estimates and propagates colored noise into the georeferencing algorithm. Nevertheless, the results show that it is beneficial to observe the target for a longer period so that errors in the navigation states are averaged as the target is observed from new poses and over a longer time period. Furthermore, the innovations illustrate that a Markov (or other colored noise model) augmentation could be beneficial when georeferenced measurements captured from a moving platform are used in a tracking system. This is investigated further in Section 7.2. Additionally, the spread of the points in Figure 5.4 indicates that the measurements can be approximated as a normally distributed variable for a large batch of measurements, even though consecutive innovations are correlated. It should also be emphasized that from an application point of view, the resulting georeferencing error in terms of root-mean-square error (RMSE) is often more important than consistency.

5.6.2 Case study 2 - Georeferencing of slowly moving vessel

A second independent experiment was carried out to study the accuracy of the georeferencing system in a relevant remote sensing application. The motivation behind this experiment was to measure the position of the marine vessel displayed in Figure 5.8. The motion of the vessel was mostly caused by drift in the sea water. However, a few short maneuvers were conducted during the experiment.

The position of the vessel was measured with a single-frequency GPS receiver with a rate of 0.5 Hz for reference (ground truth). Consequently, the reference is more uncertain than for the base station because it does not have RTK capability or a high update rate. Nevertheless, the GPS position is used as a measure for the true position, but the error in the reference is potentially a few meters. The same UAV platform and payload as in the previous case study were used in this experiment.

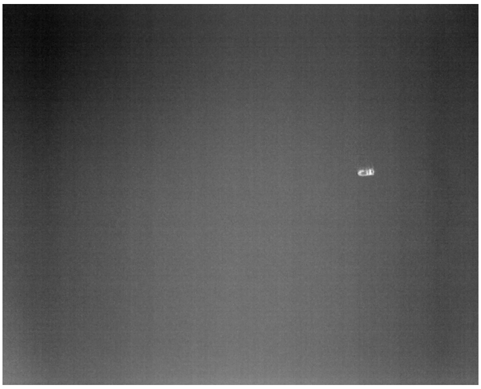


Figure 5.8: The small marine vessel used in the second case study

The vessel position in the image was detected automatically with the method presented in Section 4.3. The experiment lasted for almost 50 min and more than 22000 images were captured in total (only a part of them included the vessel). The vessel was detected in 3635 images and georeferencing was conducted for each detection. The exact same system parameters (with respect to IMU and gimbal misalignments) as in the first case study were used in this experiment. Thus, this case study is independent, but uses the same calibration and can, therefore, verify the results in the previous case study. In addition, the amount of data is larger and increases the credibility of the system.

Figure 5.9 shows the path of the UAV. The UAV operated at an altitude of approximately 200 m during the periods where the vessel was recorded. The vessel was in the field of view of the camera in many different periods and the variation in UAV poses during detection is much greater in this case study.

Figure 5.10(a) shows the distribution of georeferenced positions for a subset of 1250 images using the nonlinear observer. Figure 5.10(b) shows the distribution using the autopilot. A subset of 1250 images is shown in Figure 5.10 to enlarge the level of details. Figure 5.11 shows the distribution of georeferenced points in the entire flight using the nonlinear observer and verifies that the system works well on all images. The segment of 1250 images can be recognized in the upper left corner of Figure 5.11. The distributions in Figure 5.10(a) and 5.10(b) fit well with the GPS reference. The distribution from the nonlinear observer is perhaps following the trajectory of the reference better than the distribution from the autopilot for the subset of 1250 images, but the difference is minor.

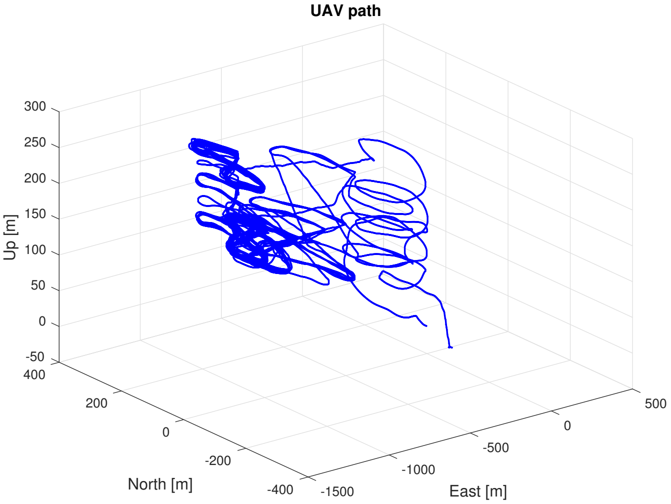
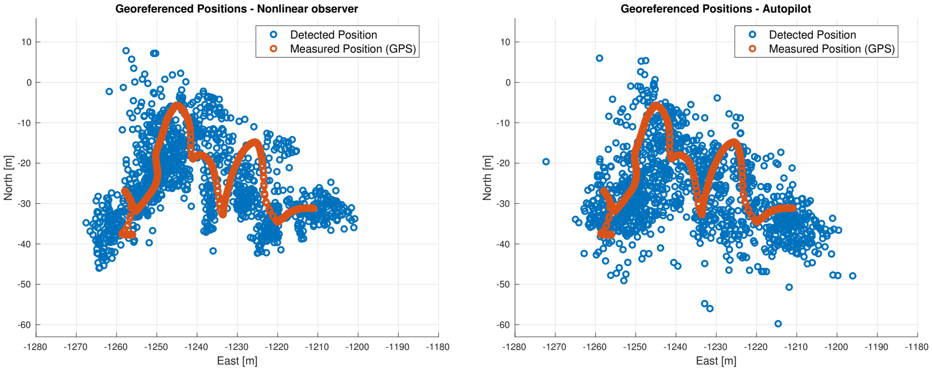


Figure 5.9: The UAV path estimated by the nonlinear observer



(a) (b)

Figure 5.10: The georeferenced vessel position for a subset of 1250 images using the nonlinear observer (a) and autopilot (b)

Table 5.6 summarizes the accuracy of the georeferenced positions for both the nonlinear observer and the autopilot for the entire flight. Table 5.7 shows the same metrics using the subset of 1250 images. The mean error for the entire flight using the observer is 1.98 m and −0.76 m in north and east, respectively. The standard deviation of the error is comparable for the observer and autopilot. The MAE of a single image is 10.25 m and 9.37 m for the observer and autopilot, respectively. Thus, the performance is comparable and in line with what was observed in the previous case study. The observer has the best performance for the reduced set of images, but the difference is minor.

The MAE in a single image is larger than in the first case study (10.25 m compared

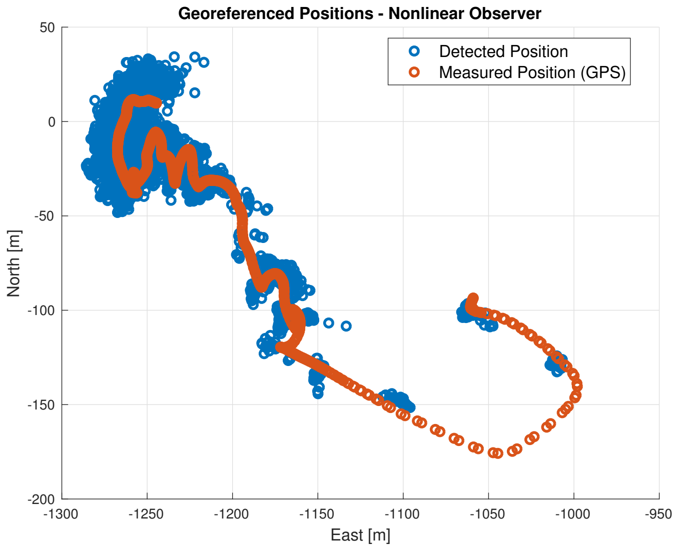


Figure 5.11: The distribution of georeferenced positions using the nonlinear observer for all images

Table 5.6: Main results of experiment 2. SD = standard deviation, MAE = mean absolute error

	Observer	**Pixhawk Autopilot**
Mean north error	1.98 m	0.37 m
Mean east error	−0.76 m	3.38 m
SD north error	7.77 m	7.83 m
SD east error	7.96 m	7.50 m
MAE position	10.25 m	9.37 m

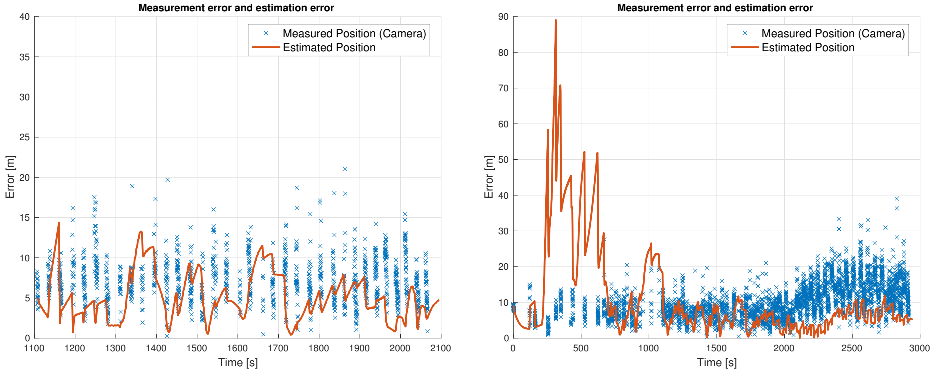
to 6.40 m), even though the altitude of the UAV was lower. This is explained by the reference, which is more uncertain. In addition, the vessel obviously moved more than the stationary GPS base station antenna, and the GPS receiver used as reference was not located exactly in the center of the vessel as assumed in the object detection algorithm. Since the images were captured from a larger set of different UAV poses and maneuvers than in the first experiment, the accuracy in this case study is reasonable. The mean position has an overall accuracy just surpassing 2 m, which means that the target position is located in a trustworthy manner and that the true position can be estimated from a small set of images.

Figure 5.12(a) shows the measurement and estimation errors when a Kalman filter, with a constant velocity motion model, is used to track the vessel for the subset of 1250 images. Figure 5.12(b) shows the tracking results for the entire flight. The initial error is larger for the entire flight since the vessel had a larger velocity in the beginning, and thus the Kalman filter is struggling initially (the filter is

Table 5.7: Main results of experiment 2 for the subset of 1250 images used in Figure 5.10. SD = standard deviation, MAE = mean absolute error

	Observer	**Pixhawk Autopilot**
Mean north error	−0.68 m	−3.53 m
Mean east error	−0.37 m	2.89 m
SD north error	4.88 m	6.37 m
SD east error	6.09 m	4.99 m
MAE position	7.29 m	7.40 m

initialized with zero speed). It is possible to tune the Kalman filter in a better way to reduce the initial error, but is not necessary since the intention of these figures is to show the achievable estimation error and the magnitude of the error in the measurements. The largest measurement error for the entire flight is almost 40 m, but the majority of the measurements have an error below 20 m. The measurement error grows somewhat at the end as displayed in Figure 5.12(b), but the estimation error is still small and within a few meters. The better part of the measurements are between five and ten meters from the reference.



(a) (b)

Figure 5.12: The georeferencing error and the estimation error when a Kalman filter is used to track the target for the subset of 1250 images (a) and the entire flight (b). The time frame in (a) corresponds to the time frame in (b).

5.6.3 Case study 3 - Velocity reconstruction using optical flow

The final case study in this chapter is based on data gathered in the Azores with the Skywalker X-8 fixed-wing UAV. Thermal images of the large marine vessel displayed in Figure 5.13 were captured. The SIFT feature detector and the FLANN nearest neighbour search were used to find common features in consecutive images. Moreover, optical flow was calculated as the displacement of these features. The sensor suite is described in [65], but the system did not include active time synchronization or misalignment calibration. Moreover, the autopilot navigation data were used so the UAV pose is less trustworthy.

The vessel was located in the camera field of view for a short period on two separate occasions. The main motivation behind this case study is to use optical flow to reconstruct the vessel velocity as described in Section 5.4. Figure 5.14(a) shows the UAV path (estimated by the autopilot) and the path of the vessel (measured by a single-frequency GPS receiver) for a period of 80 seconds. Figure 5.14(b) shows the gimbal orientation in the same period. The vessel was almost stationary, but the system had no prior knowledge about the motion of the vessel. Note that the change in the gimbal tilt angle is significant and violates some of the assumptions in Section 5.4. The camera frame rate was 7.5 Hz and not large enough to minimize the effect of the gimbal motion.

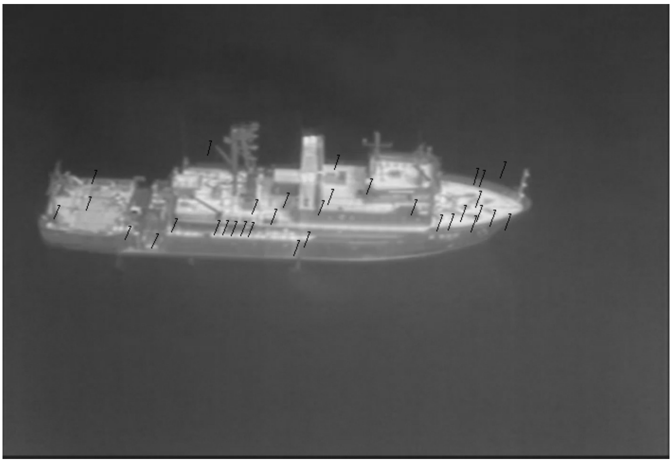
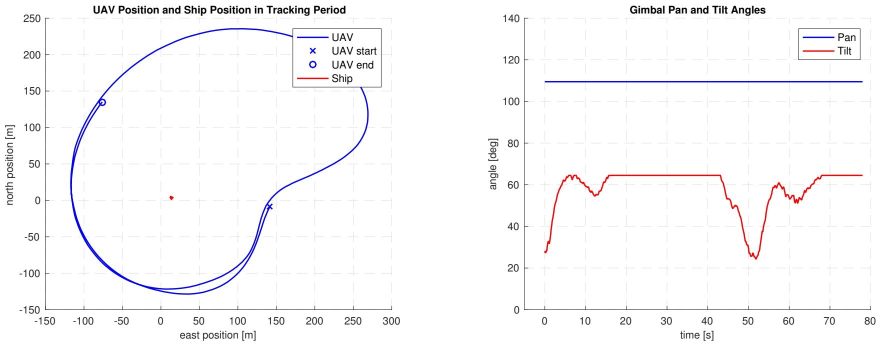


Figure 5.13: The large vessel used in the third case study with optical flow vectors

The vessel was not in the camera field of view in the time intervals [20, 50]s and [72, 80]s. Furthermore, the feature detector was not able to find features on the ship in some images. 600 images were captured in the time period and features were detected on the vessel in 250 images. A part of the vessel was visible in approximately 400 images. However, 100 of these images only contained a very small part of the vessel. The whole vessel was visible in 200 images.

Figure 5.15 shows the theoretical flow (optical flow calculated based on camera motion) and optical flow measured by SIFT in the horizontal (a) and vertical (b) direction in the image plane. Since the vessel was at rest, the theoretical flow is expected to be equal to the measured optical flow. The noise level is large, but the theoretical flow and measured optical flow are correlated. When considering the uncertainty related to synchronization of data, the gimbal motion and taking the accuracy of the sensors [65] into account, the results are reasonable.



(a) (b)

Figure 5.14: (a) Position in the NE plane for the UAV and the vessel. (b) Gimbal orientation in the same time period.

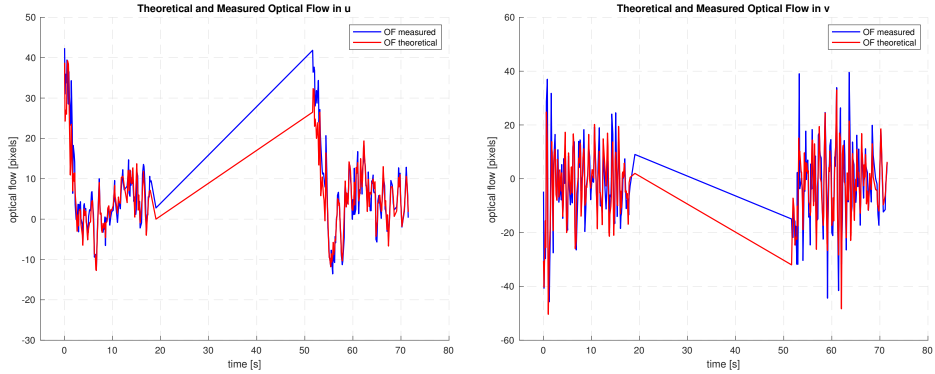


Figure 5.15: Comparison of theoretical and measured optical flow in the horizontal direction (a) and vertical direction (b)

Figure 5.16 shows the velocity of the ship reconstructed with optical flow. The variance is very large when considering that the vessel was stationary, but the mean error is within 1 m/s in both north and east. The reconstructed velocity is particularly vulnerable for synchronization errors since it is important to know the exact attitude, velocity and position of the UAV when two consecutive images are captured. This is somewhat problematic since the sampling rate of the UAV navigation data was 10 Hz, and just slightly exceeded the frame rate of the camera. Moreover, the gimbal motion was significant and far from ideal. Nevertheless, in a situation with accurate synchronization among the sensors and less gimbal motion, it is likely that the variance in the reconstructed velocity would decrease. Note that every pair of subsequent images with optical flow vectors have been used in Figure 5.16. In practice, one should filter out unlikely velocities with a threshold for maximum speed or avoid them altogether by neglecting pair of images with substantial gimbal motion between the images. This was not implemented since the goal was to analyze the accuracy without additional logic.

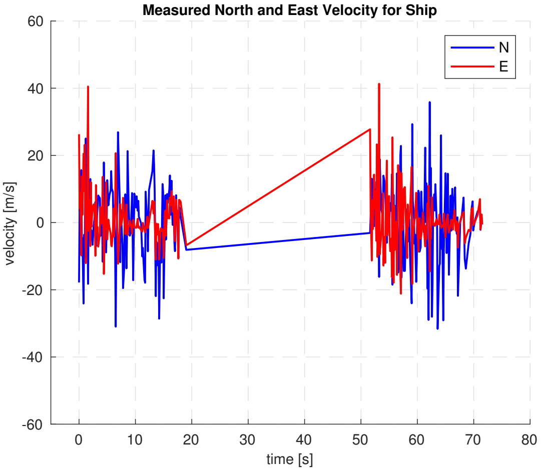


Figure 5.16: Reconstructed velocity of ship obtained with optical flow

Chapter 6

Tracking of Floating Objects Assuming Known Sensor Pose

Chapter 5 concerned georeferencing and how pixel coordinates can be transformed to more useful coordinates in an Earth-fixed coordinate frame. This chapter looks into tracking of floating objects using a monocular thermal camera. The objective is to find a suitable strategy for tracking of floating objects that utilizes detections in thermal imagery. Several tracking architectures are suitable in such a scenario, and both linear and nonlinear solutions are sensible. A linear tracking filter based on georeferencing is presented together with the more conventional solution based on the EKF and a nonlinear measurement model. Common for both solutions is the dependency on the camera pose, which is assumed to be known perfectly in this chapter (without considering potential navigation uncertainty). Tracking in the presence of uncertainty in the camera pose is covered in Chapter 7. This chapter is based on the work in [43, 44, 67] and the following topics are covered:

- Section 6.1 gives an introduction to tracking of floating objects. It introduces the problem and discusses related work and existing solutions.

- Section 6.2 presents a linear solution for the tracking problem based on a Kalman filter where georeferencing is used to acquire target observations.

- Section 6.3 presents a nonlinear solution for the tracking problem based on the extended Kalman filter. Advantages and disadvantages compared to the linear formulation are also discussed in this section.

- Section 6.4 presents four independent case studies that are used to investigate the topics covered in this chapter. They are mostly based on experimental data with a single target. The final case study covers a multi-target scenario.

6.1 Background & Related Work

An introduction to target tracking was given in Chapter 3. This chapter will apply the theory presented there in a specific application, namely tracking of floating objects using a UAV equipped with a monocular thermal camera. Since target tracking has been introduced already, related work is not covered with great depth here. Target tracking from UAVs has been covered in several articles [43, 64]. A vision-based tracking strategy for UAVs is presented in [78]. Vision-based target geo-localization is covered in [5] and multi-target tracking using recursive Ransac is described in [84]. Common for most strategies is that computationally effective solutions are needed if detection and tracking are conducted in real-time on-board UAVs.

Tracking of floating objects is an interesting research challenge that is relevant in different applications. It can be useful for search and rescue operations where missing people or vessels must be located accurately. It is also relevant for precision drop, autonomous landing on ships, and in collision avoidance for autonomous ships [56]. An advantage in tracking of floating objects, compared to the more general tracking problem, is that it is known that objects of interest are located on the sea surface. Consequently, it is not necessary to estimate the altitude of the target and this simplifies the bearing-only tracking problem that generally is present when a monocular camera is used. The problem becomes observable as long as the camera altitude is known.

Due to computational limitations on-board UAVs, only state estimators with a small computational footprint are utilized experimentally in thesis. Particle filters are for example assumed to be too computationally expensive. The Kalman filter and the extended Kalman filter are used as the main tracking filters in this chapter. Motion and measurement models are described for tracking of floating objects, but the more general equations were presented in Chapter 3. The focus here is on how floating objects can be tracked as accurately as possible within the framework of the Kalman filter and the extended Kalman filter. This involves filter design and tuning of filter parameters.

The states of interest for floating objects are the position and velocity in the horizontal plane (north and east coordinates). Therefore, the state vector consists of these states and has dimension four.

6.2 Linear Tracking Filter using Georeferencing

Georeferencing was covered thoroughly in Chapter 5 and is used to extract measurements of the north and east positions of detected objects. Since the north and east positions are acquired through georeferencing, the position is measured directly and it is possible to design a linear measurement model. As long as a linear motion model is chosen, estimation of the target states can be achieved with a regular Kalman filter as described in Section 3.2. This is beneficial because differentiation is avoided in contrast to designs based on the EKF. This section describes tracking of floating objects within a linear framework where the Kalman filter is used for estimation of the target states.

6.2.1 Constant velocity motion model

When predicting the next state in the Kalman filter, a motion model is used as a basis for prediction. The motion model describes how the target is expected to move and measurements are used to correct for deviations from the motion model. A survey of motion models for maneuvering targets is presented in [98].

Floating objects typically drift with the velocity of the sea surface or move because of actuation from propellers. The motion model of the target should represent the expected motion. It is hard to generalize the motion, but the (near) constant velocity (CV) model usually represents the motion of floating objects in a fitting manner. Marine vessels typically maintain a constant course and speed for most of the time. Extensive maneuvering is not expected and higher order models are not more accurate in general. Therefore, it is common to represent the expected motion of a target with a decoupled CV model (also called white noise acceleration model). This model is defined in discrete-time as

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{\Gamma}\mathbf{v}_k \\ \mathbf{F} &= \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ 0 & \frac{1}{2}T^2 \\ T & 0 \\ 0 & T \end{bmatrix}\end{aligned} \tag{6.1}$$

where T is the sampling interval and the state vector of the target is defined as $\mathbf{x} = [\mathbf{p}_t^n, \mathbf{v}_t^n]^\top = [p_t^N, p_t^E, v_t^N, v_t^E]^\top$, and includes position and velocity in north and east. The motion is assumed to be decoupled between the coordinate dimensions as shown in $\mathbf{F}$. The covariance of the process noise $\mathbf{v}$ is calculated as

$$\begin{aligned}\mathbf{Q} &= E[\mathbf{\Gamma}\mathbf{v}_k\mathbf{v}_k\mathbf{\Gamma}^\top] \\ &= \mathbf{\Gamma}\sigma_v^2\mathbf{\Gamma}^\top \\ &= \sigma_v^2 \begin{bmatrix} \frac{1}{4}T^4 & 0 & \frac{1}{2}T^3 & 0 \\ 0 & \frac{1}{4}T^4 & 0 & \frac{1}{2}T^3 \\ \frac{1}{2}T^3 & 0 & T^2 & 0 \\ 0 & \frac{1}{2}T^3 & 0 & T^2 \end{bmatrix}\end{aligned} \tag{6.2}$$

where σ_v is a design parameter and should be in the order of the maximum expected longitudinal acceleration [4]. A larger value indicates that the constant velocity assumption is violated more significantly than what a smaller value indicates. It should be adjusted based on the target of interest. The target control input is unknown in this model and assumed to be white noise. This assumption is obviously violated during maneuvers. Moreover, decoupling between the coordinates is also a simplification, but works well in general. The advantage with this model is the simplicity and the main drawback is that maneuvers rarely can be modeled as white noise since they are correlated in time. An analysis of how the covariance of the process noise can be chosen in the CV model for different vessels is presented in [119].

6.2.2 Measurement model

The measurement model is simple when georeferencing is used to acquire measurements. Detections of targets in the image plane are transformed to position measurements (north and east) outside of the tracking filter. Therefore, the following simple measurement model is used:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x}_k + \mathbf{v}_k \tag{6.3}$$

A fundamental challenge with the measurement model is that it is a converted measurement approach. The true measurement, which is the pixel location of targets in the image plane, is converted to north and east coordinates with georeferencing. Thus, the true measurement goes through a nonlinear transformation and this leads to issues related to optimality and tuning of the filter parameters. That is discussed in Section 6.2.4.

6.2.3 Target gating

Designing the validation region (see Section 3.3 for specific formulas for the validation region) is not particularly challenging in the linear tracking filter based on georeferencing. The innovation is simply the difference between the georeferenced position received by the Kalman filter and the predicted position as shown in (3.5f). The innovation covariance is given by (3.5d) where the measurement matrix **H** is given in (6.3). The validation gate is then calculated to investigate if new measurements fall within the gate or not.

6.2.4 Tuning of filter parameters and initialization

Perhaps the biggest challenge when using a linear tracking filter based on georeferencing is tuning of the covariance for the measurement noise. As mentioned in Section 6.2.2, this type of filter formulation is a converted measurement approach.

When the true measurement goes through a nonlinear transformation, the covariance of the measurement noise should in theory go through the same transformation to represent the new and transformed uncertainty. This is not straightforward because of the complexity of the georeferencing algorithm. Moreover, it is not beneficial to go through this process because the noise in the detection process is less influential than navigation errors in georeferencing. Consequently, tuning of the measurement noise depends more on the application, sensor suite and type of sensor calibration than the noise induced by the detection algorithm. Nevertheless, the estimated covariance in the Kalman filter is not optimal when georeferencing is used so the validation region is not necessarily representing the true uncertainty precisely.

A convenient way to model the measurement noise could be to use the pixel uncertainty of the detection algorithm and transform that into an uncertainty in meters. However, this approach is also struggling with the fact that navigation errors are dominating the pixel uncertainty. Thus, the most appropriate approach is to use results and statistics from georeferencing experiments to tune the measurement noise covariance. [67] presents a strategy where the measurement noise covariance is scaled with the altitude of the UAV. The reasoning behind this strategy is that georeferencing errors, caused by errors in the camera orientation, are proportional with the altitude. This is a reasonable way to design the covariance of the measurement noise in a dynamic manner.

The process noise covariance must also be tuned for the CV model described in Section 6.2.1. Small vessels have the ability to maneuver quickly and usually need a larger value for σ_v than tankers that maneuver slowly with large time constant. It is difficult to find generic values that work well in several scenarios, and it is shown in [119] that different vessels need different values. A possibility is to use different modes based on the type of vessel if image recognition is used for classification. However, it is possible to find values that work sufficiently in several scenarios as shown in Sections 7.5.3 and 7.5.4.

Initialization of position for the linear formulation is straightforward. The first georeferenced position measurement is used to initialize the estimated position. The initial velocity is unknown. A sensible approach is to initialize the velocity at zero with large covariance that reflects the uncertainty. It is also possible to use differencing between two consecutive measurements to approximate the velocity, but it can be quite dangerous if the noise level in georeferencing is large (especially if the target velocity is small). [75] also argues that it is better to initialize the filter with a single position measurement than with a two-point strategy. The initial covariance in position is typically chosen somewhat larger than the corresponding measurement covariance. Large initial covariance means that the first few measurements are weighted strongly and vice versa. Thus, it is not beneficial to initialize the covariance larger than necessary because poor measurements in the beginning of the tracking period can reduce the performance for a long time unless the measurement noise is modeled correctly. This is especially an issue if it is a significant amount of time between batches of measurement, which normally occur because targets move outside of the camera field of view. Specific values for tuning are discussed more closely in the case studies in Section 6.4.

Data association and track initialization are described in the case studies when these topics are relevant for the purpose. Since, clutter is unexpected, tracks can be initiated whenever a target is detected in two consecutive images. The *m out of n* strategy discussed in Section 3.4 can also be used. If multiple targets occur in the validation region, a firm data association is conducted to find the most likely measurement for each target. Track termination is not considered because it is not the focus of the case studies later in this chapter. The target moves outside of the field of view of the camera for extended periods and could potentially lead to track termination and re-initialization numerous times.

6.2.5 Advantages and disadvantages

There are several advantages and disadvantages with a linear framework for target tracking. The obvious advantage is that it is simple and that the Kalman filter is computationally effective compared to other solutions, especially since the number of states is limited. Moreover, the measurements can easily be related to the target position and are thus intuitive compared to the EKF that requires differentiation.

The obvious drawback is that the optimality of the Kalman filter is violated when measurements are acquired through georeferencing. The measurement noise cannot be interpreted as zero-mean Gaussian noise because the measurement error is dominated by the accuracy of the camera pose. Since images are captured closely in time, errors in georeferencing are both correlated in time and have a non-zero mean unless a long time period is considered. As a result, tuning of filter parameters must be based on statistics from real experiments and cannot be connected to the detection algorithm. Moreover, it is hard to find values that work in multiple scenarios, even though an interesting approach for generalizing these parameters is presented in [67].

6.3 Nonlinear Tracking Filter

The conventional approach when designing a tracking filter based on optical sensors is to use an EKF to estimate the target states. This is because of the nonlinear nature of camera measurements. This section describes a nonlinear approach for tracking of floating objects where the biggest difference compared to the linear approach lies in how the measurement model is defined. The target motion model is linear (the CV model) and equal to the one presented in Section 6.2.1. The state vector is obviously also equal so the difference solely is in the measurement model, but results in a different choice for target gating. Note that the knowledge about the target being located on the sea surface is utilized in this approach as well.

Therefore, this is not a pure bearing-only tracking problem, which usually appears in tracking with monocular optical sensors.

6.3.1 Measurement model

The measurement model for the nonlinear tracking approach is based on the pinhole camera model presented in Section 5.2.3. The target detection algorithm provides the pixel position of the object center in the image plane. The detected pixel position (u, v) is related to the camera frame through the pinhole camera, which is repeated here for completeness:

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z_t^c} \begin{bmatrix} x_t^c \\ y_t^c \end{bmatrix} \tag{6.4}$$

where $\mathbf{z}$ is the measurement vector, f is the focal length and (x_t^c, y_t^c, z_t^c) are the coordinates of the pixel decomposed in the camera-fixed coordinate frame. The camera-fixed position of a target detected in a thermal image is related to the camera pose and target NED positions so that

$$\mathbf{p}_{ct}^c = \begin{bmatrix} x_t^c \\ y_t^c \\ z_t^c \end{bmatrix} = \mathbf{R}_n^c(\mathbf{p}_{nt}^n - \mathbf{p}_{nb}^n) \tag{6.5}$$

where $\mathbf{R}_n^c$ is the rotation matrix between {c} and {n} and assumed known. $\mathbf{p}_{nt}^n$ is the target position in NED and $\mathbf{p}_{nb}^n$ is the NED positions of the UAV. Note that (6.5) assumes that the camera is placed in the origin of the body frame so that {c} and {b} coincides. Equation (6.5) is inserted in (6.4) so that the pinhole camera model is expressed in terms of the target coordinates decomposed in {n} and the navigation states of the UAV. Therefore, the measurement model can be expressed as a nonlinear function of the target coordinates and the navigation states of the UAV:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}_t, \mathbf{p}_{nb}^n, \boldsymbol{\Theta}_{nc}) \tag{6.6}$$

where $\boldsymbol{\Theta}_{nc}$ is used to represent the attitude of the camera relative to NED and $\mathbf{x}_t$ is the target state vector. At this stage, the type of attitude parametrization is insignificant because the main requirement is to know the rotation matrix between {c} and {n}. The choice of attitude representation is more important in Chapter 7 and discussed further there. Note that (6.6) only depends on the target position and not the target velocity as seen in (6.5).

The EKF (see Section 3.2 for information about the EKF) requires linearization of the measurement model (6.6) with respect to the target states. The Jacobian takes the form

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}_t} = \begin{bmatrix} \frac{\partial z_1}{\partial p_t^N}\Big|_{\hat{\mathbf{x}}_t, \mathbf{p}_{nb}^n, \boldsymbol{\Theta}_{nc}}, & \frac{\partial z_1}{\partial p_t^E}\Big|_{\hat{\mathbf{x}}_t, \mathbf{p}_{nb}^n, \boldsymbol{\Theta}_{nc}}, & 0, & 0 \\ \frac{\partial z_2}{\partial p_t^N}\Big|_{\hat{\mathbf{x}}_t, \mathbf{p}_{nb}^n, \boldsymbol{\Theta}_{nc}}, & \frac{\partial z_2}{\partial p_t^E}\Big|_{\hat{\mathbf{x}}_t, \mathbf{p}_{nb}^n, \boldsymbol{\Theta}_{nc}}, & 0, & 0 \end{bmatrix} \tag{6.7}$$

where $\hat{\mathbf{x}}_t$ is the most recent estimate of the target states. The last two columns of the Jacobian are zero because the measurement model does not include the target velocity. Finding the Jacobian theoretically is obviously possible with a symbolic derivation, but the resulting expression is too long to write down in this thesis. Moreover, it might be more effective to find the Jacobian numerically in a computer implementation.

6.3.2 Target gating

Creating the validation region for target gating is somewhat different for the non-linear approach. The first possibility is to use the measurement model in (6.4) directly. The innovation based on this model can be expressed as

$$\boldsymbol{\nu}_1 = \begin{bmatrix} u \\ v \end{bmatrix} - \underbrace{\frac{f}{z_t^c} \begin{bmatrix} x_t^c \\ y_t^c \end{bmatrix}}_{\mathbf{h}} \tag{6.8}$$

where (6.5) replaces the camera-fixed coordinates and the vector $[u, v]^\top$ is the detected pixel coordinates (received measurement). The validation region is computed with the formula in Section 3.3 where the innovation covariance is extracted from the EKF, and the Jacobian is used when finding the measurement matrix.

Another possibility is to represent the measurement model in terms of azimuth (φ) and elevation (ϑ) angles (instead of pixel coordinates), as illustrated in Figure 6.1. This representation takes the form

$$\begin{aligned} \mathbf{z} &= \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathbf{h}_2(\mathbf{x}_t, \mathbf{p}_{nb}^n, \boldsymbol{\Theta}_{nc}) \\ \mathbf{z} &= \begin{bmatrix} \varphi \\ \vartheta \end{bmatrix} = \underbrace{\begin{bmatrix} \arctan\left(\frac{y_t^c}{x_t^c}\right) \\ \arctan\left(\frac{z_t^c}{\sqrt{(x_t^c)^2+(y_t^c)^2}}\right) \end{bmatrix}}_{\mathbf{h}_2} \end{aligned} \tag{6.9}$$

where (6.5) is inserted for the camera-fixed coordinates to get a model depending on the target states and the known camera pose. Azimuth and elevation angles are calculated from detected pixel coordinates in the following manner

$$\begin{bmatrix} \varphi \\ \vartheta \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{\frac{1}{f_y} v}{\frac{1}{f_x} u}\right) \\ \arctan\left(\frac{f_x}{u} \cos(\varphi)\right) \end{bmatrix} \tag{6.10}$$

where f_x and f_y were defined in Section 5.3 as focal lengths in the horizontal and vertical part of the image plane. The innovation is defined from (6.9) and (6.10), where (6.10) is the measurement (u and v are detected) and (6.9) is the predicted measurement based on the target states and the known camera pose. Therefore, the innovation is expressed as

$$\boldsymbol{\nu}_2 = \underbrace{\begin{bmatrix} \arctan\left(\frac{\frac{1}{f_y}v}{\frac{1}{f_x}u}\right) \\ \arctan\left(\frac{f_x}{u}\cos(\varphi)\right) \end{bmatrix}}_{\text{received measurement}} - \underbrace{\begin{bmatrix} \arctan\left(\frac{y_t^c}{x_t^c}\right) \\ \arctan\left(\frac{z_t^c}{\sqrt{(x_t^c)^2+(y_t^c)^2}}\right) \end{bmatrix}}_{\text{predicted measurement}} \tag{6.11}$$

The validation region is calculated as described above.

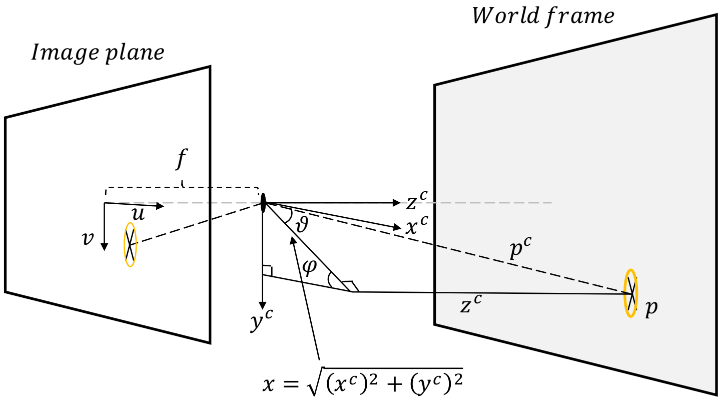


Figure 6.1: Illustration of the azimuth (φ) and elevation (ϑ) angles. The camera points straight down from the image plane to the world-frame in this illustration where z^c is the range from the camera to the plane the pixel is located in. This is simply the altitude of the UAV with zero roll, pitch and gimbal angles.

The second representation, $\mathbf{h}_2(\cdot)$, is described because it is argued that it is more accurate near the image boundaries for data association [12]. Therefore, $\mathbf{h}(\mathbf{x}_t, \mathbf{p}^n_{nb}, \boldsymbol{\Theta}_{nc})$ is used in the measurement update step and $\mathbf{h}_2(\mathbf{x}_t, \mathbf{p}^n_{nb}, \boldsymbol{\Theta}_{nc})$ can be used for gating.

6.3.3 Tuning of filter parameters and initialization

Using a nonlinear tracking filter eliminates one of the biggest issues related to the linear tracking filter based on georeferencing, at least in theory. The true measurement, which is the pixel location of targets in the image plane, is used directly in the tracking filter and thus the expected uncertainty in the detection algorithm can be used to tune the measurement noise covariance. However, the Jacobian still depends on the camera pose as shown in (6.5). Consequently, the Jacobian is affected by the accuracy of the UAV position and attitude. If only the uncertainty in the detection algorithm is considered, the filter will be optimistic and estimate a covariance that is much smaller than the real estimation error. Moreover, the measurements are trusted more than they should, which can result in poor accuracy. Thus, this type of formulation requires some prior knowledge or experimental experience for tuning of the measurement noise. It will also be platform dependent as discussed in Section 6.2.4.

With the discussion above in mind, a clear advantage with the nonlinear representation of the tracking system is perhaps not intuitive. Nevertheless, it is necessary to use this representation in Chapter 7. Moreover, this representation is based on the true measurement model and not the converted measurement approach in the linear formulation. It can be used in situations where it is necessary to remove the flat-earth assumption and covers a more general formulation of the tracking problem. Removing the flat-earth assumption (and known target altitude) leads to a bearing-only tracking problem. Note that observability of the tracking estimates is questionable in bearing-only tracking and that camera motion is needed for observability [12]. Moreover, initialization requires several images and camera or target motion between the images.

Initialization of the nonlinear filter is not straightforward because the measurement model cannot be used to extract a specific set of position coordinates as in the linear formulation. One way to do it is to combine the first few measurements and find the intersection between these measurements since the measurement model can be interpreted as a vector with known direction and unknown length. However, since the targets are located on the sea surface, a much simpler approach is to use georeferencing of the first measurement to find the initial position. The discussion regarding initialization of the velocity and the covariance matrix in Section 6.2.4 is equally relevant in the nonlinear formulation and is not repeated. This also applies to the discussion regarding track initialization, track termination and data association.

6.3.4 Advantages and disadvantages

The main advantage with the nonlinear approach is that the detected pixel position of the target is used directly in the measurement model. Thus, it is much simpler to tune the measurement noise in situations where the camera pose is known accurately. However, the severe camera motion is problematic in practice and the camera pose is a much greater error source than the detection algorithm. Thus, the benefit compared to the linear formulation is to some degree canceled by this issue. Nevertheless, a clear advantage is the fact that the nonlinear formulation is valid in the general sense, and that it can be used for target tracking in other environments as long as observability is maintained.

The clear drawback with this formulation is the use of a nonlinear state estimator. The filter is more prone to initialization errors and can diverge with poor initialization. Thus, accurate tuning of the initial covariance is vital for the EKF to maintain stability of the estimates. This is particularly challenging because the EKF is weaker for errors in the navigation pose due to the linearization. Another factor is the increase in computationally complexity due to the calculation of the Jacobian at each time step.

6.4 Case Studies

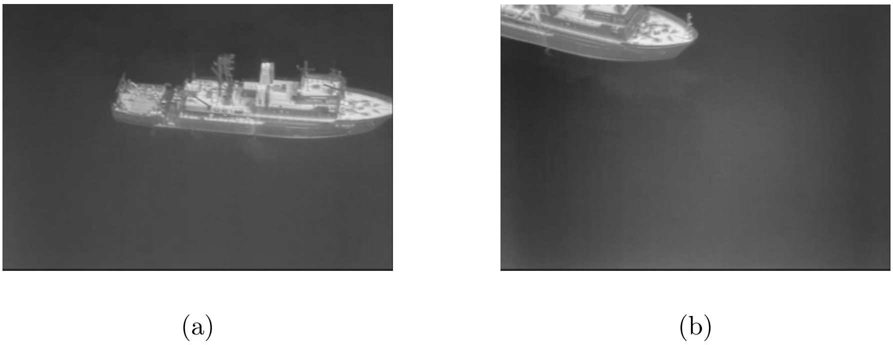
This section presents four independent case studies. The main objective with the case studies is to investigate aspects of the tracking formulations presented in this chapter. Case study one, in Section 6.4.1, presents tracking of a single target within the nonlinear formulation and is based on [43]. Case study two, in Section 6.4.2, compares the linear and nonlinear formulation on a large set of data. These results have not been published previously. The third case study is presented in Section 6.4.3 and covers data association for the different tracking architectures. It was originally published in [43]. The final case study is presented in Section 6.4.4 and shows a multi-target scenario using data from [67] with the linear tracking formulation.

6.4.1 Case study 1 - Tracking of large ship with the nonlinear formulation

This case study evaluates the performance of the nonlinear tracking formulation described in Section 6.3 with the (near) CV motion model. This case study is based on navigation data and images gathered at a flight experiment with the X8 Skywalker fixed-wing UAV. The payload is described in [65]. Note that time synchronization and camera misalignment calibration were not conducted for this payload, and the lack of calibration is a weakness that degrade the accuracy. Navigation data were stored at a rate of 10 Hz. The vessel displayed in Figure 6.2 was used as target and has a length of approximately 70 m. Data association is neglected because all measurements are known to originate from the target. Thus, evaluation of the object detection algorithm and the nonlinear filtering part of the tracking system is the focus in this case study.

The altitude of the UAV was approximately 100 m during the flight. The target was visible in 441 images, which correspond to a period of 45 s. Note that these images were based on five different segments in time that have been merged into one continuous image sequence. This ensured that the ship was visible in all images. The position of the target was measured with a single-frequency GPS receiver and used as ground truth. Estimates with and without the centroid adjustment described in Section 4.3 are shown in this case study to illustrate why it can be beneficial to calculate the center of the ship in situations where only a part of the vessel is visible. Only a part of the vessel (as illustrated in Figure 6.2(b)) was visible in the majority of the images. The filter parameters are described in [43], but is not important for the purpose of this case study.

The position was initialized with georeferencing of the first available detection, and the initial velocity was set to zero. Moreover, an approximation of the size of the ship (in pixels) was assumed to be known. In practice, the size can be determined the first time the whole vessel is visible in an image. The ship was almost at rest



(a) (b)

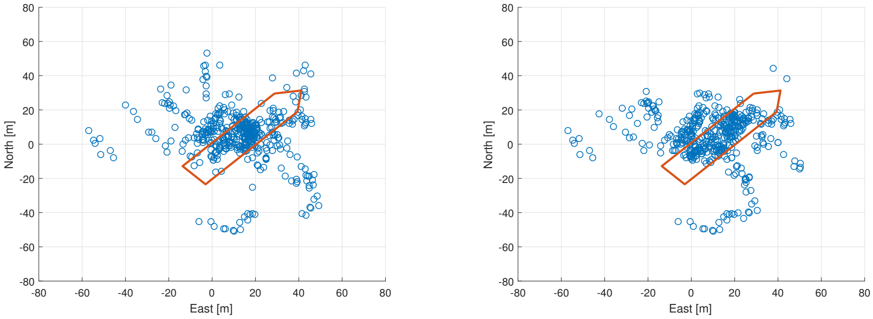
Figure 6.2: Thermal images of the ship that is tracked in the first case study. Many images only contain a small part of the ship (as illustrated in (b)).

during the tracking period, but the tracking filter had no knowledge about the behavior beforehand.

441 images have been processed in this case study. The object detection algorithm was able to detect the vessel in 438 of the images, which corresponds to a success rate of 99.3 %. The high success rate is explained by the high signal-to-noise ratio in the images (see Figure 6.2) as discussed in Section 4.4.1.

Figure 6.3 shows the NE positions obtained with georeferencing with and without the centroid adjustment. A larger part of the measurements are distributed near the center of the vessel with the centroid adjustment, even though the difference is quite small. Some important measures are summarized in Table 6.1. The mean georeferenced position is somewhat closer to the GPS reference with the centroid adjustment. Therefore, the benefit of finding the center is illustrated. The variance increases a bit with the centroid adjustment. That is perhaps because the adjustment is only performed in situations where the major axis is significantly larger than the semi-major axis. Therefore, the center is not adjusted in every image and the distribution is likely to have a larger variance. This is because the difference between measurements with and without the adjustment is expected to be larger than the difference between two measurements without adjustment. Note that the variance is mainly related to sensor noise in the navigation states of the UAV and not uncertainty in the object detection algorithm. Hence, the variance would be reduced significantly if the navigation states were perfectly known. Note also that the poor accuracy in georeferencing, compared to the case studies in Chapter 5, is explained by the lack of time synchronization and camera misalignment calibration.

Figure 6.4(a) shows the total distance between the GPS measured position and the estimated position from the nonlinear tracking filter with and without the centroid adjustment. After the initial period, the estimate with the adjustment is more accurate, even though the difference is below 5 m for most of the time. The mean



(a)

(b)

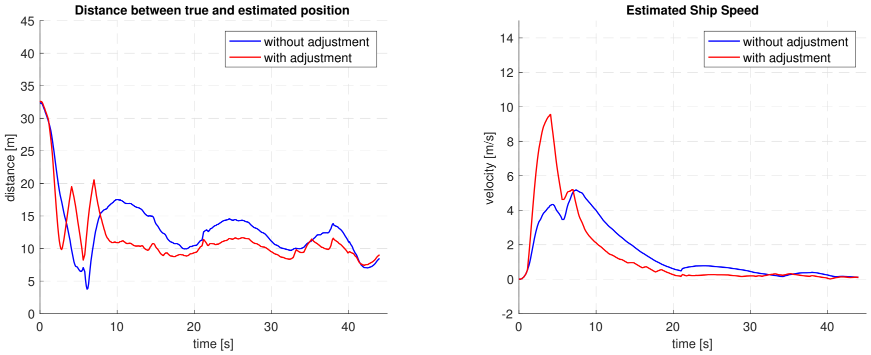
Figure 6.3: Georeferenced north and east positions with (a) and without (b) the centroid adjustment

Table 6.1: Results of georeferencing with and without the centroid adjustment during object detection. SD = standard deviation

Measure	**Reference**	**With adjustment**	**Without adjustment**
Mean north position	6.6 m	4.8 m	3.6 m
Mean east position	16.4 m	12.2 m	9.7 m
SD north position	x	16.4 m	15.2 m
SD east position	x	18.2 m	17.5 m

absolute distance between the true and estimated position is 11.54 m and 13.07 m with and without the centroid adjustment, respectively. However, if the initialization period is neglected (the first 10 s), the mean is 9.96 m and 12.21 m, respectively. The accuracy increases with the centroid adjustment and the adjustment seems to be beneficial. Nevertheless, it is important to emphasize the fact that the centroid adjustment is only necessary for targets of a certain length. The difference would be negligible for small vessels.

The estimated position is reasonable, especially because the accuracy of the navigation states is limited. The GPS receiver on the ship was not mounted exactly in the center of the ship and single-frequency GPS receivers can have an uncertainty of a few meters. Figure 6.4(b) shows the estimated target speed. The vessel was nearly at rest during the experiment (dynamic positioning) and the estimated speed is close to zero after the first period. An interesting observation is that the speed converges to zero more rapidly with the centroid adjustment. A speed of exactly zero is hard to achieve because of the magnitude of the measurement errors.



(a)

(b)

Figure 6.4: Estimation error in position (a) and estimated speed (b)

Nevertheless, the estimated speed is small so the performance is reliable.

6.4.2 Case study 2 - Linear vs. nonlinear tracking

The objective in the second case study is to compare the linear and nonlinear tracking formulation. This case study is based on experimental data gathered at two different flight experiments in the fall of 2017 with the thermal camera payload described in Section 2.3. These results have not been published before, but a part of the experimental data has been used in [47] and in [45]. The small marine vessel displayed in Figure 7.1 in Chapter 7 was used as target. The tracking systems described in Section 6.2 and 6.3 are compared by using the exact same set of measurements and navigation data. The target was detected automatically by the method in Section 4.3. Navigation data estimated by the nonlinear observer presented in Section 5.5 are used in the first flight, and the autopilot navigation data are used in the second flight to demonstrate that both solutions work. The payload has been synchronized in time and calibrated for misalignment errors as discussed previously.

Data association is neglected in this case study because only one target is considered and clutter is not present. Several measurements are available initially so track initialization is straightforward. Moreover, all measurements are used whether they fall within the validation region or not. This is because this type of tracking formulation struggles with being to optimistic as will be demonstrated later. Thus, using the validation region would lead to track loss and the need to re-initiate the track several times, unless it is tuned in a pessimistic manner. This is not desirable as the intention is to compare the accuracy of the linear and nonlinear tracking formulations. Note that when NEES is discussed later, it is only with respect to position. This is because a reliable reference for the velocities is unavailable.

Flight 1 - Tracking of high-speed target

The goal in the first flight was to track a high-speed target moving with nearly constant course. The UAV operated at an altitude of approximately 350 m and the path is shown in Figure 6.5.

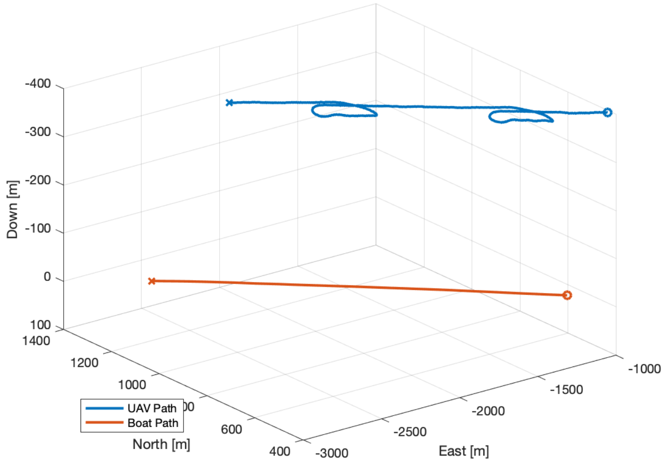


Figure 6.5: Path of the UAV and the high-speed target

The reference is a single-frequency GPS receiver that was on-board the target. Both solutions use the CV motion model. The target states are initialized with zero speed and position given by the first georeferenced measurement. The initial covariance for the north and east positions is $(10\,\mathrm{m})^2$, which is a typical georeferencing error as shown in Chapter 5. The initial covariance for the velocity is chosen as $(10\,\mathrm{m/s})^2$ and is large because the initial velocity is unknown. σ_v^2 in (6.2) is chosen as $(0.05\,\mathrm{m/s^2})^2$ and the low value is caused by the target motion, which fits well with the CV motion model. The covariance of the measurement noise for the linear formulation (georeferencing) is chosen as $(7\,\mathrm{m})^2$. The measurement noise covariance is chosen to be $(20\text{ pixels})^2$ for the EKF, which actually corresponds to $(7\,\mathrm{m})^2$ at the altitude the UAV kept.

Figure 6.6 shows the estimation error in position. It also shows the periods where the vessel was detected, which only cover a small part of the tracking period. Figure 6.7 shows the estimated path for both the KF and EKF. The estimated positions are similar and no significant difference is observed. This is supported by the average results presented in Table 6.2. The average estimation error is comparable, but the linear formulation with the KF has the best accuracy overall. The similar performance is expected since both filters are tuned the same way and use the same set of measurements. The difference is caused by the accuracy of the

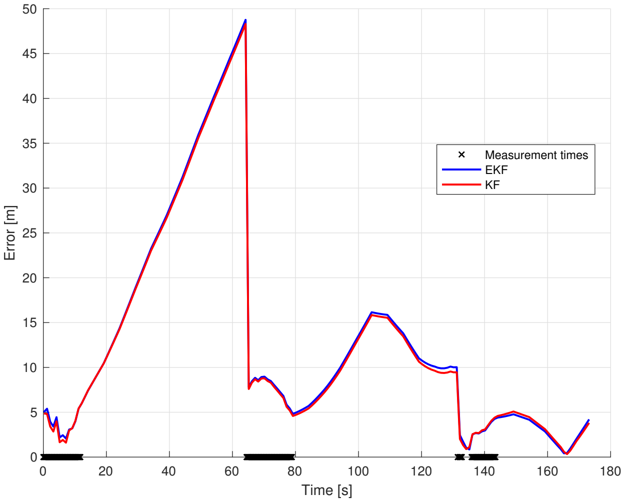


Figure 6.6: Estimation error in position

georeferencing algorithm compared to linearization of the measurement equations. The estimation error is below 10 m in periods where measurements are available and only grows a few meters when prediction is used after the second segment with measurements.

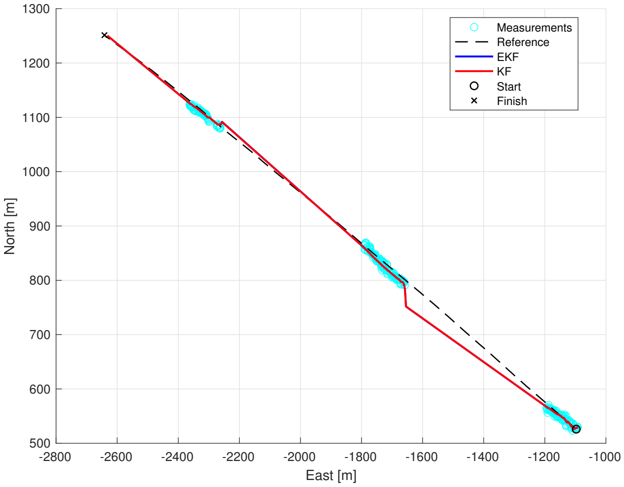


Figure 6.7: Estimated path and reference

Figure 6.8 shows the estimated speed and course. It is not possible to observe any difference between the EKF and KF, and both filters estimate the speed and course in an accurate manner. Figure 6.9 shows the NIS and NEES (defined in Section 3.5). The NEES is larger than the confidence bounds, which means that both fil-

Table 6.2: Average performance metrics in flight 1 - tracking of high-speed target

Parameter	**EKF**	**KF**
Mean estimation error [north, east]	$[-9.43\,\text{m}, 1.55\,\text{m}]$	$[-9.26\,\text{m}, 1.12\,\text{m}]$
Mean absolute estimation error (position)	13.01 m	12.81 m
Mean NEES	17.22	15.67
Mean NIS	0.50	0.46

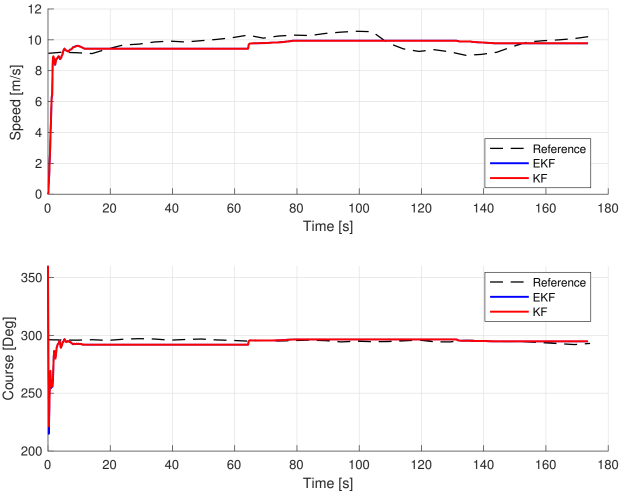


Figure 6.8: Estimated speed and course

ters are too optimistic. The estimated covariance is smaller than the corresponding estimation error. The NIS shows that the filter is pessimistic with respect to the measurements, and that the covariance of the measurement noise could have been reduced. This shows the fundamental challenge in tracking from a moving platform with uncertain camera pose. The NIS indicates that the covariance of the measurement noise could be reduced while NEES indicates that the covariance is too small and that the covariance matrices for both the measurement and process noise should be increased. This is caused by the uncertainty in the camera pose which is neglected in the design. This issue is investigated further in the next chapter.

Figure 6.10 shows the correlation in the innovation sequence and illustrates that they are correlated in time. This is also a violation of the consistency criteria. The correlation is actually a bit larger for the EKF and the vertical pixel coordinate, which is likely to be because of the linearization. Even though consistency is a challenge, the results show that a high-speed target can be tracked accurately. From an application-based point of view, the accuracy of the state estimates is more important than maintaining consistency in a single target scenario. The accuracy

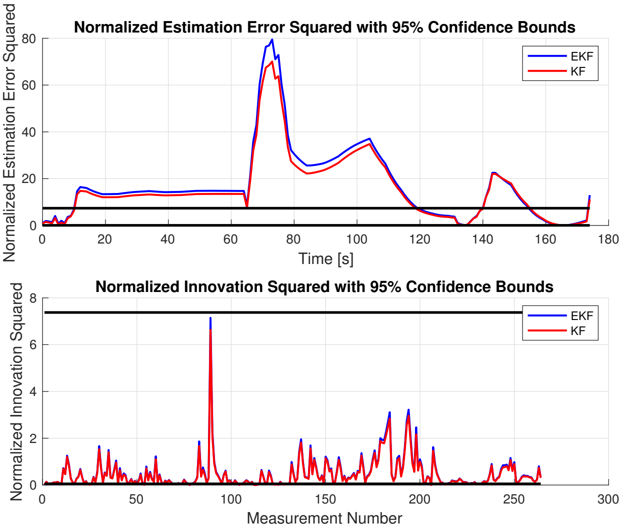


Figure 6.9: (a) Normalized estimation error squared and normalized innovation squared

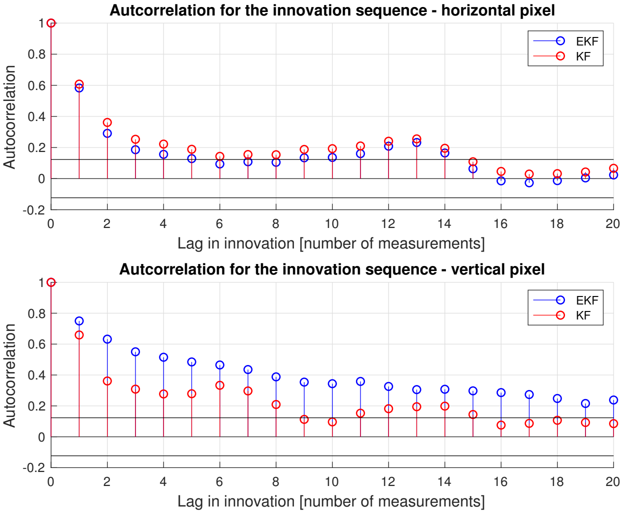


Figure 6.10: Autocorrelation for the innovation sequences

obtained in these results are better than previous results [43, 44], especially when taking the altitude of the camera and the speed of the target into account.

Flight 2 - Tracking of slowly moving target

The goal in the second flight was to track a slowly-moving target. The UAV operated at an altitude of nearly 300 m and the path is shown in Figure 6.11. The reference and filter parameters are almost equal to the previous flight. The only differences are that $\sigma_v^2 = (0.01\,\text{m/s}^2)^2$, and that the measurement noise covariance is increased to $(10\,\text{m})^2$ and $(30\text{ pixels})^2$. The increase in the measurement noise is because the measurements are more uncertain because of a different flight pattern with more variation in the attitude. The process noise is reduced because the vessel moved more slowly, and the focus here is to show what kind of accuracy one can expect in these types of experiments.

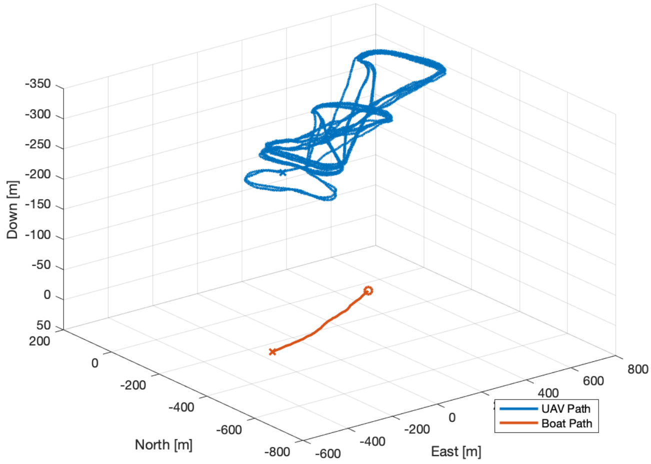


Figure 6.11: Path of the UAV and the slowly-moving target

Figure 6.12 shows the estimation error in position, and Figure 6.13 shows the estimated path for both the KF and EKF. Moreover, Figure 6.12 also shows the periods where the vessel was detected, which only cover a small part of the tracking period. The average results are presented in Table 6.3. The average estimation error is slightly smaller for the linear formulation overall, but both formulations provide accurate position estimates.

Figure 6.14 shows the estimated speed and course. These estimates are also similar to what was experienced in the previous flight and follow the references accurately. Figure 6.15 shows the NIS and NEES. Both filters are too optimistic. The difference between the filters are more prominent in this flight and that can be somewhat surprising. Both filters have been tuned in the same manner, but the NIS is larger for the EKF. This is most likely because the linearization is more inaccurate in this flight compared to the georeferencing equations. Moreover, it can be explained by

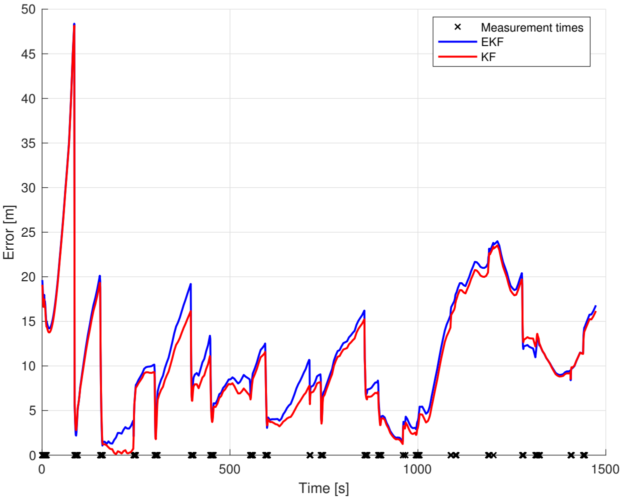


Figure 6.12: Estimation error in position

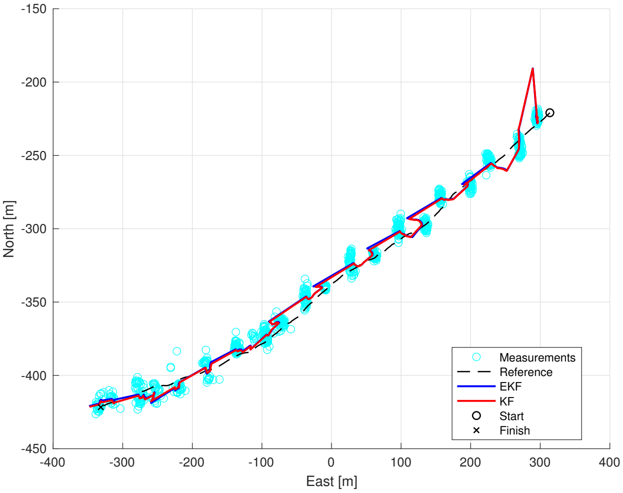


Figure 6.13: Estimated path and reference

the path and attitude variations that were more severe. The UAV followed straight-line segments in the first flight and circular motion in this flight. Circular motion with more variations in the attitude leads to larger errors in georeferencing.

Figure 6.16 shows the correlation in the innovation sequence. The correlation is perhaps even stronger in this flight. This is most likely explained by the target speed, which was much lower. Low target speed leads to even greater influence from the UAV pose and increases the correlation.

This case study has shown that both the linear and nonlinear tracking formulations

Table 6.3: Average performance metrics in tracking of slowly-moving target

Parameter	**EKF**	**KF**
Mean estimation error [north, east]	[2.73 m, −4.94 m]	[2.33 m, −4.40 m]
Mean absolute estimation error (position)	11.25 m	10.46 m
Mean NEES	48.1	29.9
Mean NIS	1.04	0.60

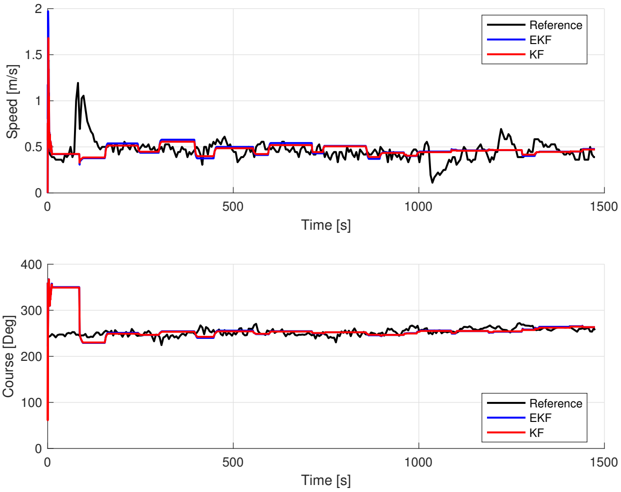


Figure 6.14: Estimated speed and course

can be used to track different targets with estimation error in position limited to a few meters. It is perhaps not surprising that the performance of the KF and EKF is similar, but it is important to validate that both solutions work in tracking of floating objects. The biggest challenge (when data association is neglected) is the uncertainty in the camera pose, which leads to consistency issues. The estimated covariance is smaller than the corresponding estimation error. How this can be mitigated is studied in Chapter 7. Moreover, both solutions are suboptimal due to the flat-earth assumption.

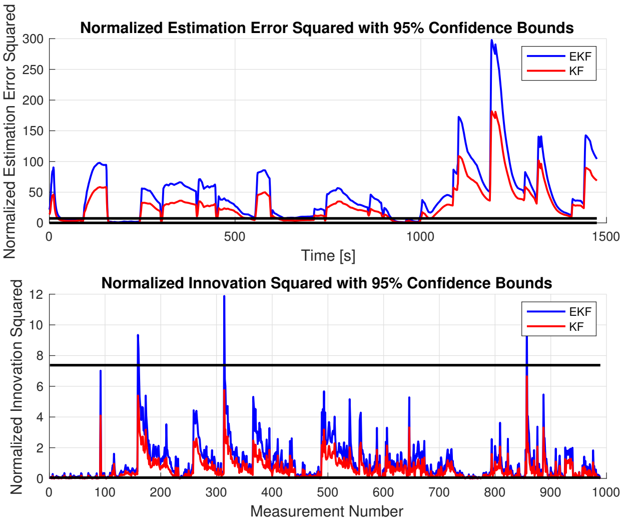


Figure 6.15: Normalized estimation error squared and normalized innovation squared

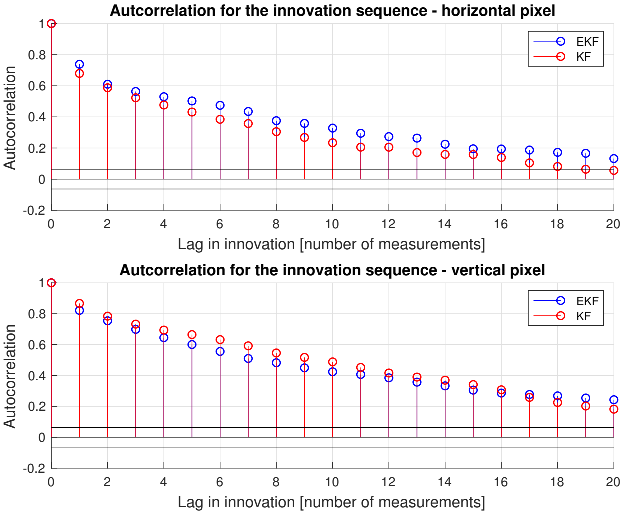


Figure 6.16: Autocorrelation for the innovation sequences

6.4.3 Case study 3 - Data association for linear and nonlinear tracking filters

The main objective in the third case study is to compare the different methods for target gating. The linear gating strategy described in Section 6.2.3 is compared with the two nonlinear strategies in Section 6.3.2. Evaluation of the methods is achieved through Monte Carlo simulations. It is based on [43]. A part of the data in this

Table 6.4: Covariance of the Gaussian white noise added to the UAV navigation states and the object detection algorithm in Case study 3

State	**Covariance Gaussian noise**
North and east positions UAV	$(2\,\mathrm{m})^2$
Down position UAV	$(5\,\mathrm{m})^2$
Roll and pitch UAV	$(1°)^2$
Yaw UAV	$(2°)^2$
Gimbal pan and tilt angles	$(1°)^2$
Uncertainty detection algorithm (image plane coordinates)	(2 pixels)^2

case study is based on experimental data. Autopilot navigation data were originally sampled at a field experiment at a frequency of 10 Hz. Images were gathered with the same frequency during the experiment. This case study uses 141 images that were captured over a time period of 14 seconds. The payload used in the field experiment is described in [65] and is the same one used in case study 1 in Section 6.4.1.

Existence of two targets at rest that are located 10 (north) and 15 meters (east) from each other (mean values) are simulated in this case study. Gaussian noise with a standard deviation of 10 meters is added to the north and east positions for both targets. Therefore, the true distance between the targets varies for each simulation. The true pixel positions of the targets (in the image) are calculated with the navigation states of the UAV and the NED positions of the objects for each image (with "backwards" georeferencing). Gaussian noise, with properties described in Table 6.4, is added to the true pixel positions and used as measurements of the pixel position. Moreover, white noise (properties described in Table 6.4) is added to the navigation states of the UAV so that the measurement models are affected by navigation uncertainty. Thus, the simulation includes uncertainty related to both the image detection algorithm and the navigation states of the UAV.

One object (referred to as the main object) is tracked so that the estimated position and covariance are available. The goal is to find out which of the two measurements that belong to the main object in each image. The measurement with the smallest normalized innovation squared is assigned to the main target. 141 images are processed for each Monte Carlo simulation and the association is either right or wrong for every image. The success rate (number of right associations divided by number of images) is the main performance measure. The states for the main object are propagated through 50 iterations in the Kalman filter initially (before association begins) to ensure that the estimates have somewhat converged before data association is conducted. The measurements for these iterations are generated as the true position where Gaussian noise with standard deviation of 10m in both the north and east positions is added. 10000 Monte Carlo simulations were conducted in total.

Table 6.5: Initial conditions and measurement noise for the different association methods in case study 3

Property	**Covariance**
North and east initial positions	$(10\,\mathrm{m})^2$
North and east initial velocities	$(\sqrt{20}\,\mathrm{m/s})^2$
Georeferencing uncertainty north and east positions	$(10\,\mathrm{m})^2$
Pixel uncertainty u in (6.4)	360 pixels^2
Pixel uncertainty v in (6.4)	160 pixels^2
Uncertainty azimuth angle (6.9)	$(6°)^2$
Uncertainty elevation angle (6.9)	$(3°)^2$

Table 6.6: Success rate for the data association methods in case study 3

Data association method	**Success rate**
The linear case (georeferencing)	96.64 %
The nonlinear case - EKF (6.8)	96.83 %
The nonlinear case - azimuth and elevation angles (6.11)	95.99 %

The initial conditions and measurement noise related to each association method are summarized in Table 6.5. The covariance of the measurement noise (the bottom four elements in Table 6.5) is large, but that is mainly because the accuracy of the navigation states is limited. In other words, the magnitude of the covariance is caused by the navigation states and not the accuracy of the object detection algorithm.

Table 6.6 presents the main results. The success rate for the different methods is similar, which means that all of the representations are applicable in practice. More importantly, the nonlinear methods have the same performance as the linear method. Note, that the azimuth and elevation approach is the least accurate. However, the performance is so similar that one cannot rule out the possibility of the design parameters being the reason for the difference.

Figures 6.17, 6.18 and 6.19 show results of a single Monte Carlo simulation. The success rate for the angle representation is 92.2 % in this particular simulation. The success rate for the nonlinear approach in pixel form and the linear representation is 91.5 %. Hence, in this simulation, the angle representation is able to associate one more image correctly compared to the other representations. One interesting consideration in Figure 6.17(a) is that even though the pixel and angles form have almost the same success rate, the mistakes in the association are not occurring at the exact same set of images. Therefore, it seems to be situations where the pixel representation has advantages and other situations where the angle form is better. Thus, it could be possible to get more robust measurement association by combining these methods in the nonlinear case. However, this is not investigated further in this thesis.

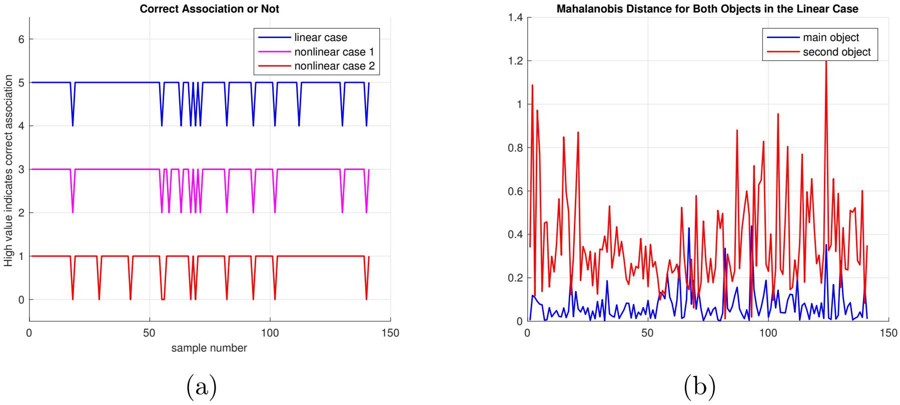


Figure 6.17: Results of a single simulation. (a) Illustration of when associations are correct and not. The values 1, 3 and 5 indicate successful associations for the different methods. The values 0, 2 and 4 indicate wrong associations. The nonlinear case 1 and 2 are the pixel form and angles representation, respectively. (b) Normalized innovation squared for the linear formulation.

Figure 6.17(b) and 6.18 show the NIS (Mahalanobis distance) for the three methods. The NIS for the measurements from the main object is obviously much smaller in general, but some outliers exist that cause erroneous measurement association. Figure 6.19 shows the estimated north and east positions for the main object and the measurements for the main and second object with georeferencing. Notice that the measurements are distributed on a large area in the north-east plane and that is mainly because of the noise related to the UAV navigation states. Noise in the image detection algorithm has a much smaller influence as discussed previously.

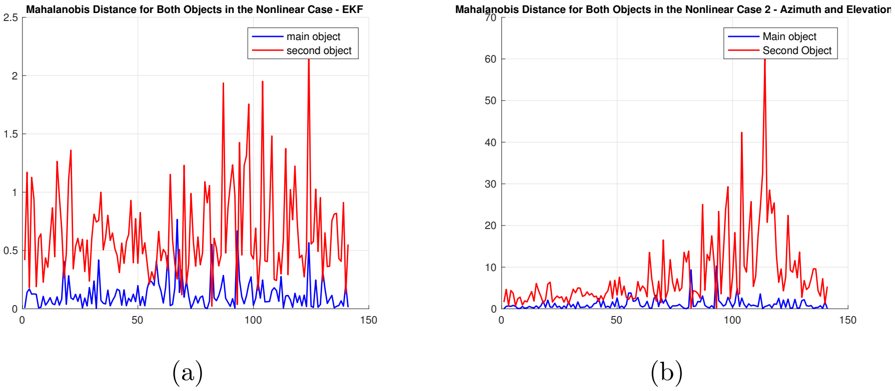
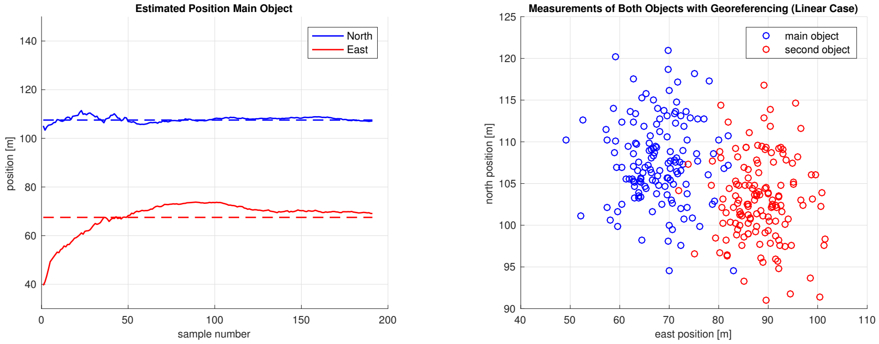


Figure 6.18: Results for a single simulation. (a) Normalized innovation squared in the nonlinear case with innovation in (6.8). (b) Normalized innovation squared in the nonlinear case with innovation from (6.11).



(a) (b)

Figure 6.19: Results for a single simulation. (a) Estimated north and east positions for the main object. (b) Measurements of the main object and the second object (with georeferencing).

6.4.4 Case study 4 - Tracking of multiple targets with data association

The fourth case study looks into a multi-target scenario. It is based on the work and data presented in [67]. The objective is to investigate the linear tracking formulation in a relevant multi-target scenario. The case study is based on experimental data gathered at a flight experiment in the fall of 2017. Two independent marine vessels operated in close proximity and the thermal camera payload was used to collect the necessary data. The intention is to show that two targets can be tracked efficiently at the same time. Moreover, a few issues that arise in multi-target tracking are covered. This means that detection, data association and the filtering part are all tested together. The specific system is described in [67].

Both targets were instructed to maintain a low speed of approximately 1 m/s. The paths of the UAV and the vessels are displayed in Figure 6.20. The UAV flew in an eight-figure pattern at an altitude of approximately 300 m. The vessel denoted boat 1 was told to cross the path of the other vessel to complicate measurement association. The tracking period lasted for 19 minutes and the crossing occurred after approximately four minutes. Two single-frequency GPS receives were used to measure the true position of the targets.

The detection algorithm described in Section 4.3 was used to detect targets in thermal images. Moreover, the global nearest neighbour approach (Section 3.3) was used for data association. Georeferencing was used to acquire target position measurements, and the linear formulation described in Section 6.2 was used to track the targets. 2400 detections were gathered in the tracking period. 1535 belonged to boat 1 and the remaining 865 to boat 2. This information was obtained by manual classification. 349 images had both vessels within the same image. The measure-

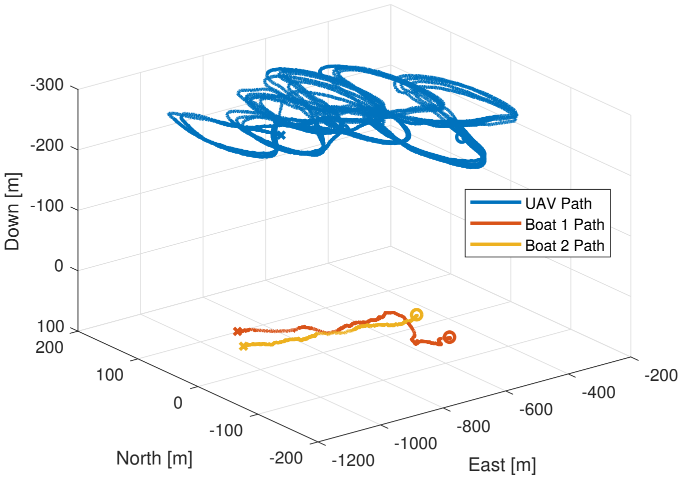


Figure 6.20: Paths of UAV and the vessels

ment noise standard deviation was chosen as $(0.05h)$m where h is the altitude of the UAV. Thus, the measurement noise standard deviation was approximately 17.5 m in both north and east. The process noise standard deviation was chosen as 0.2 m/s^2. The initial standard deviations in position and velocity were chosen as 17.5 m and 5 m/s, respectively. The tracks were initialized after two consecutive measurements in close proximity.

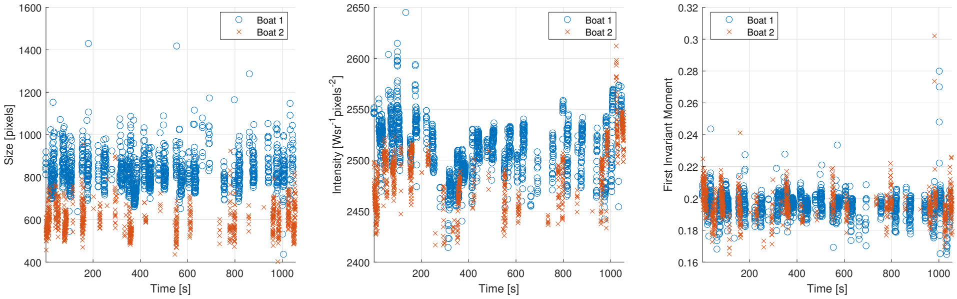


Figure 6.21: Distribution of image features (size, intensity and the first invariant Hu moment) for boat 1 (blue) and boat 2 (orange)

Figure 6.21 shows three independent image feature distributions for the two objects. The features are size, intensity and the first centralized invariant Hu moment [52]. These features describe the visual appearance of the two objects and can be used as an aid in measurement association. If the tracking system stores values for these features together with the state estimates, they can be used when new measurements are received. Figure 6.21 shows that the size of the object has the best spread in between the boats in this particular scenario. There is also a visible

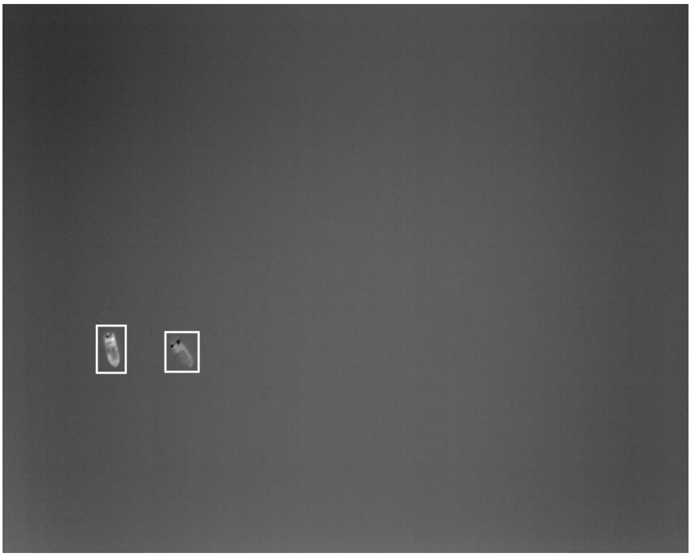


Figure 6.22: Example of thermal image with two vessels. This image was also used as example in Figure 4.9(d).

difference in intensity while the Hu moment is similar for both objects. This is explained by the similar geometric form and not surprising since the Hu moment is both size and rotation invariant. A thermal image of both objects is shown in Figure 6.22.

Figure 6.23 shows the estimates of both targets without using visual features in data association. Only the physical distance (NIS) is used to distinguish between the targets. Since the GNN data association approach is used, a measurement is associated to only one track. One can see that a track swap occurs at the crossing because measurements are wrongly associated. The tracking system never recovers from this situation. Moreover, the system never detects the swap and continues to the estimate the position of the wrong object. Nevertheless, one could argue that the system still accurately estimates the position of both objects, even though a swap has occurred. Thus, it is in a situation where it is necessary to distinguish between the objects that this behavior is critical. If it is sufficient to know that objects are present at the estimated locations, the system works satisfactorily.

Figure 6.23 shows that firm data association based on physical distance is problematic when crossings occur. This is a situation where it would be desirable to do a soft association instead. However, another possibility is to use the image features displayed in Figure 6.21 as a tool in data association. The results in Figure 6.24 shows such a scenario, where image features for each object are stored when a track is initialized. Moreover, data association is based on the GNN search where a weighted sum of both the physical and the feature distance is used. This procedure is described in detail in [67]. Note that image features also can be included in the

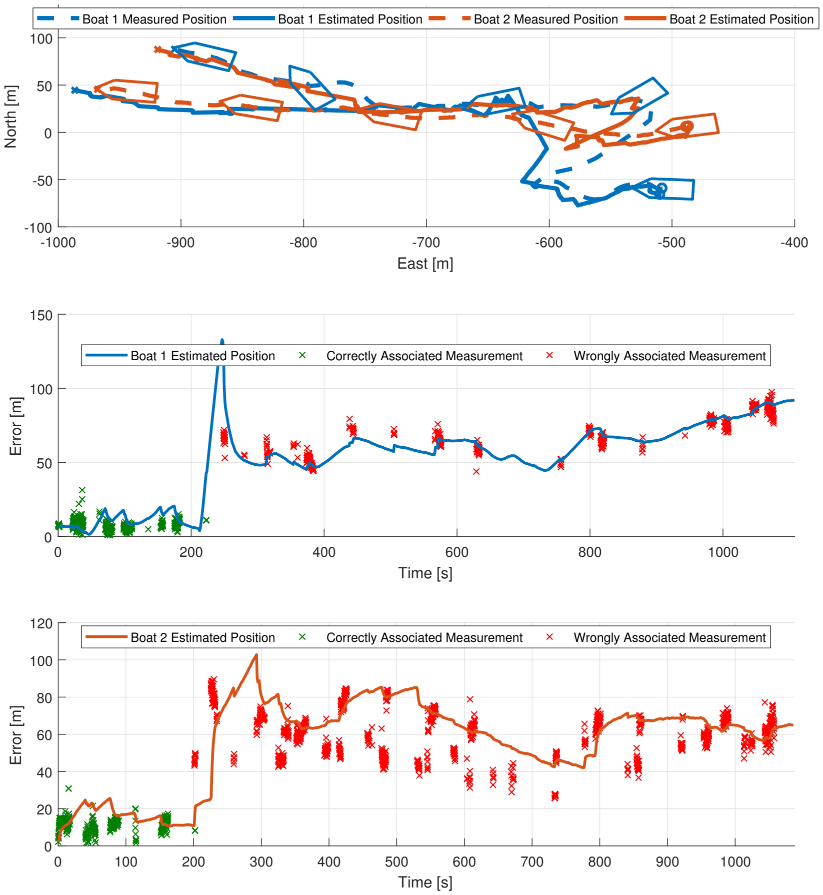


Figure 6.23: Tracking results without visual features in data association

state vector and estimated in the Kalman filter. Such a solution would incorporate image features directly into the NIS and is an interesting idea for future work.

The results presented in Figure 6.24 illustrate that both targets are tracked accurately. The position of both targets are mostly within 20 m of the true position. Five measurements are wrongly associated for the first vessel while no measurements are wrongly associated to the second vessel. This shows the main advantage with using image features since most of the measurements are associated correctly in contrast to the situation without image features. This case study has proved that the linear tracking formulation described in Section 6.2 works in a multi-target scenario with two vessels when image features are utilized. The nonlinear formulation

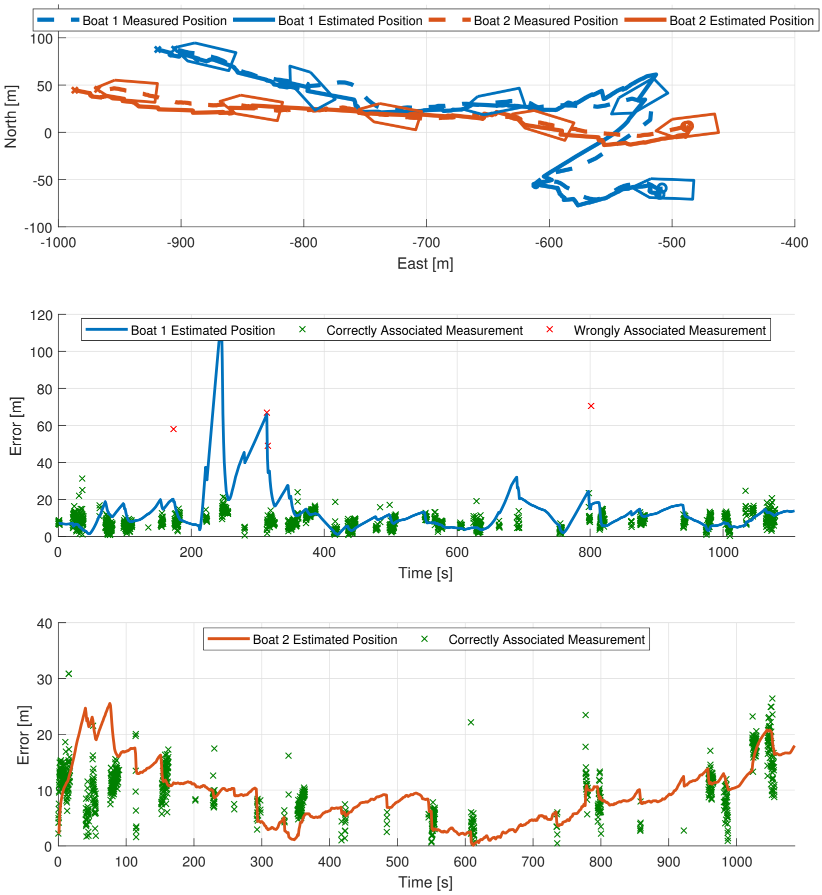


Figure 6.24: Tracking results with visual features in data association

is obviously also expected to work in the same manner.

Chapter 7

Tracking of Floating Objects in the Presence of Navigation Uncertainty

Chapter 6 considered tracking of floating objects where the pose of the thermal camera was assumed to be known without accounting for navigation uncertainty. However, such reasoning often leads to inconsistency in the tracking filter unless it is tuned in a conservative manner as shown in the previous chapter. This is because a large part of the measurement noise is caused by uncertainty in the camera pose and not only by uncertainty in the detection algorithm. Moreover, errors in the camera pose are most often correlated for images captured closely in time. Consequently, the white noise assumption, present in most state estimators, is violated and inconsistency occurs. This chapter investigates tracking of floating objects in the presence of navigation uncertainty and how these effects can be mitigated. This chapter is based on the work in [47] and [45]. The following topics are covered in this chapter:

- Section 7.1 gives an introduction to tracking in the presence of navigation uncertainty. It introduces the problem and discusses related work and existing solutions.
- Section 7.2 presents a method for relaxing the white-noise assumption by handling correlated noise in the tracking filter, both for motion and measurement models.
- Section 7.3 presents a navigation system based on the error-state formulation of the EKF. This is covered because it is important for understanding the foundation of the Schmidt-Kalman filter presented in Section 7.4.
- Section 7.4 presents a tracking solution based on the Schmidt-Kalman filter where the covariance of the navigation estimates is incorporated directly in the tracking filter. It uses the navigation system based on the error-state formulation presented in Section 7.3.

- Section 7.5 presents four independent case studies used to evaluate and validate the topics covered in this chapter.

7.1 Background & Related Work

Much work has been conducted on tracking of moving objects using optical sensors where the sensor pose is stationary. However, tracking of moving objects using sensors on a moving platform constitutes a different challenge. Several articles cover situations where it is assumed that the sensor pose is known or can be estimated with high precision [43, 90]. However, the sensor pose is never perfectly known and uncertainty degrades the tracking accuracy. In addition, the motion of the UAV exceeds the motion of typical objects on the sea surface and makes it harder to distinguish camera and target velocities.

Tracking of objects using UAVs equipped with optical sensors has been studied previously [46, 67, 89]. However, a common challenge is to obtain the desired accuracy while maintaining filter consistency. It is vital to know the orientation and the position of the camera precisely at all times to obtain the desired tracking accuracy. This is for example the case in a tracking system based on georeferencing as described in Section 6.2. The camera pose depends on the position and attitude of the UAV. Therefore, errors in the navigation system of the UAV affect the tracking system directly and can cause a drastic loss in performance when the navigation states are uncertain. This is especially troublesome because accurate covariance estimation is needed for measurement association.

Most tracking filters are based on measurement models where the measurement error is modeled as additive zero-mean Gaussian noise. This makes sense if the error is not systematic. The uncertainty in the detected object position in a thermal image can normally be accepted as white. However, in georeferencing, the pixel coordinates are converted to an Earth-fixed frame based on the navigation states of the UAV. This is a converted measurement approach and the georeferencing error is usually correlated for consecutive images due to correlations in the navigation error. Consequently, colored noise is induced in the measurement model. One possibility is to model colored noise directly in the motion and measurement models, and use georeferencing and a linear tracking filter [47]. This is described in Section 7.2 and includes strategies for modeling both the process and measurement noise as correlated processes.

Navigation uncertainty is also problematic when an EKF is used as in Section 6.3. In periods where the navigation estimates are uncertain, the accuracy of the Jacobian is reduced. The reduced accuracy might not be a significant issue, but only if the covariance estimates represent the true uncertainty. In practice, this means that the tracking system should automatically adapt to the accuracy of the current navigation estimates. This is hard to achieve without considering the quality of the navigation estimates directly in the tracking system. Moreover, the tracking system is usually assumed to be uncorrelated with the navigation system. This is violated if navigation errors remain constant. Therefore, there is a need to find a solution that incorporates the quality of the navigation estimates directly in the tracking system and this is investigated in Section 7.4.

Target tracking in the presence of navigation uncertainty has been studied for autonomous ships [11, 118, 125] equipped with sensors that can acquire range and bearing measurements. However, these articles have focused on a subset of the navigation space and used a radar as the tracking sensor. Moreover, experimental results are not presented in these articles. Fixed-wing UAVs have much faster dynamics than ships and the attitude can change quickly. Thus, compensation of navigation uncertainty is expected to be even more important for target tracking using UAVs. Section 7.4 covers compensation of navigation uncertainty in both camera position and attitude.

7.2 Handling Violation of the White Noise Assumption

This section deals with how colored noise can be added in both the motion and measurement models. It is assumed that the tracking system is based on georeferenced measurements as in Section 6.2. However, the methods discussed here are also applicable and relevant for the nonlinear tracking formulation described in Section 6.3.

7.2.1 Correlated noise in the motion model

The simplest way to incorporate colored noise in the motion model of the target is to choose a different model than the CV model presented in Chapter 6. Correlated noise can be modeled directly in the motion model as discussed in [98]. Note that correlated noise in the motion model is related to target maneuvers and not uncertainty in the camera pose. During maneuvers, such as turns or during linear acceleration, the target motion do not obey a CV model. Other models approximate the target behavior more accurately in these situations.

Finding a single model that represents target motion during different maneuvers is obviously impossible and explains why multiple model approaches were developed. However, the CV model is most often accurate and reliable on average. Moreover, it can be dangerous to over-parametrize the motion model as this can degrade the reliability of the predicted motion unless all states are estimated precisely. A constant acceleration model can for example be less accurate because it is hard to estimate the acceleration precisely, and estimation errors in acceleration give large drift in position in periods without measurements.

A motion model with correlated noise, that can be appropriate for marine vessels and floating objects, is the Singer acceleration model [98]. This is an alternative to the CV model where the acceleration is modeled as a zero-mean stationary Markov process. This model allows systematic acceleration changes that automatically converge back to zero with an adjustable time constant. It represents the motion of marine vessels in an appropriate manner because maneuvers tend to finish quite quickly before motion with constant course and velocity is resumed. The acceleration is modeled with autocorrelation function

$$E[a(t+\tau)a(t)] = \sigma^2 e^{-\alpha|\tau|} \tag{7.1}$$

where α is the inverse of the expected maneuver time constant. Low values indicate a long maneuver and vice versa. Marine vessels rarely change course so a small value is perhaps the most appropriate choice, but depends on the type of ship. σ is the variance of the acceleration and modeled as

$$\sigma^2 = \frac{a_M^2}{3}(1 + 4p_M - p_0) \tag{7.2}$$

where a_M is the maximum expected acceleration, p_0 is the probability of zero acceleration and p_M is the probability of the target accelerating (or decelerate) at the maximum rate.

The Singer model is a continuous-time model and is defined as (only one coordinate dimension shown for simplicity):

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \tag{7.3}$$

where the state vector $\mathbf{x}$ consists of the position, velocity and acceleration in the corresponding dimension. The model can obviously be extended to cover both of the horizontal coordinates. The system must be discretized for implementation and the discrete-time process noise has covariance matrix equal to

$$\mathbf{Q} = 2\alpha\sigma^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \tag{7.4}$$

Nonlinear motion models that account for turns or other maneuvers can also be chosen, but are not included in this thesis. These models do not work well in situations without maneuvers and must be combined with other models through multiple-model approaches. This is not assessed to be necessary or beneficial for the purpose of this thesis.

7.2.2 Correlated noise in the measurement model

Systematic errors in the camera pose lead to correlated noise in the measurement model as discussed in Section 7.1. This section describes how the measurement noise can be whitened. The method is based on the work presented in [47] and applied here in a tracking system based on georeferencing. Lets assume that georeferencing is used to acquire measurements in a linear tracking filter as in Section 6.2. The goal is to model the measurement noise as a correlated process. The main argument for this approach is to maintain the simplicity that georeferencing and a linear system provide, but account for correlated noise. Consider the system model

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{v}_k \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{w}_{c,k} \\ \mathbf{w}_{c,k+1} &= \mathbf{F}_c \mathbf{w}_{c,k} + \mathbf{w}_{w,k} \\ E[\mathbf{w}_{w,k}\mathbf{w}_{w,j}^\top] &= \mathbf{R}_w \delta_{kj} \\ E[\mathbf{v}_k \mathbf{v}_j^\top] &= \mathbf{Q}\delta_{kj} \\ E[\mathbf{v}_k \mathbf{w}_{w,j}^\top] &= \mathbf{0} \end{aligned} \tag{7.5}$$

where δ_{jk} is 1 when $k = j$ and zero otherwise. $\mathbf{x}$ is the state, $\mathbf{z}$ is the measurement and $\mathbf{v}$ is process noise. The state transition matrix $\mathbf{F}$ is chosen as in the CV model (6.1). The measurement matrix $\mathbf{H}$ is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{7.6}$$

where it is assumed that the first two elements in $\mathbf{x}$ are the north and east positions, and the remaining two are the corresponding velocities. Colored noise in the measurement is modeled with $\mathbf{w}_c$. The transition matrix for the measurement noise, $\mathbf{F}_c$, is chosen as the identity matrix since errors in the camera pose, which cause the correlated noise, are expected to be equal for two consecutive images captured closely in time.

So how can $\mathbf{w}_c$ be included in the tracking filter? One possibility is to augment the state vector with $\mathbf{w}_c$ and estimate the noise process. However, this is troublesome because the estimate of $\mathbf{w}_c$ can be used to cancel the noise in the measurement $\mathbf{z}$, which means that the filter believes that the measurement is perfect. Consequently, the covariance of the corresponding states is pushed to zero and is problematic for measurement association. Therefore, a better strategy is to use the so-called differenced measurement approach [4] and rewrite (7.5). A new measurement model $\mathbf{y}$ is defined as

$$\begin{aligned} \mathbf{y}_k &= \mathbf{z}_{k+1} - \mathbf{F}_c \mathbf{z}_k \\ &= \mathbf{H}\mathbf{x}_{k+1} + \mathbf{w}_{c,k+1} - \mathbf{F}_c\mathbf{H}\mathbf{x}_k - \mathbf{F}_c\mathbf{w}_{c,k} \\ &= \mathbf{H}\mathbf{F}_k\mathbf{x}_k + \mathbf{H}\mathbf{v}_k + \mathbf{w}_{w,k} - \mathbf{F}_c\mathbf{H}\mathbf{x}_k \\ &= \underbrace{(\mathbf{H}\mathbf{F}_k - \mathbf{F}_c\mathbf{H})}_{\mathbf{H}_k^*}\mathbf{x}_k + \underbrace{(\mathbf{H}\mathbf{v}_k + \mathbf{w}_{w,k})}_{\mathbf{w}_k} \\ &= \mathbf{H}_k^*\mathbf{x}_k + \mathbf{w}_k \end{aligned} \tag{7.7}$$

where the new measurement noise $\mathbf{w}$ is white, but correlated with the process noise since it includes $\mathbf{v}$:

$$E[\mathbf{v}_k \mathbf{w}_j^\top] = \mathbf{Q}\mathbf{H}^\top \tag{7.8}$$

The covariance of the new measurement noise is

$$E[\mathbf{w}_k\mathbf{w}_j^\top] = \mathbf{R}\delta_{kj} = (\mathbf{HQH}^\top + \mathbf{R}_w)\delta_{kj} \tag{7.9}$$

where $\mathbf{R}_w$ is the covariance of the expected error in the original measurement $\mathbf{z}$. To remove the correlation between the new measurement and the process noise, a new motion model is defined as:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{v}_k + \mathbf{T}[\mathbf{y}_k - \mathbf{H}^*\mathbf{x}_k - \mathbf{w}_k] \\ &= \underbrace{(\mathbf{F} - \mathbf{TH}^*)}_{F^*}\mathbf{x}_k + \underbrace{(\mathbf{v}_k - \mathbf{T}\mathbf{w}_k)}_{\mathbf{v}_k^*} + \mathbf{T}\mathbf{y}_k \\ &= \mathbf{F}^*\mathbf{x}_k + \mathbf{v}_k^* + \mathbf{T}\mathbf{y}_k\end{aligned} \tag{7.10}$$

where $\mathbf{T}$ is a transformation matrix used to remove the cross-correlation and $\mathbf{v}^*$ is the new process noise. It is chosen so that

$$\begin{aligned}E[\mathbf{v}_k^*\mathbf{w}_k] &= 0 \\ E[(\mathbf{v}_k - \mathbf{T}\mathbf{w}_k)(\mathbf{w}_k)] &= 0 \\ \mathbf{QH}^\top - \mathbf{TR} &= 0 \\ \mathbf{T} &= \mathbf{QH}^\top\mathbf{R}^{-1}\end{aligned} \tag{7.11}$$

The new process noise covariance matrix is:

$$\mathbf{Q}^* = E[\mathbf{v}_k^*(\mathbf{v}_k^*)^\top] = \mathbf{Q} - \mathbf{QH}^\top\mathbf{R}^{-1}\mathbf{HQ} \tag{7.12}$$

The new system (based on $\mathbf{y}$, $\mathbf{F}^*$, $\mathbf{H}^*$, $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{Q}^*$) is implemented as a regular Kalman filter, but note that the measurement model $\mathbf{y}_k$ includes both $\mathbf{z}_{k+1}$ and $\mathbf{z}_k$. Thus, when a new measurement $\mathbf{z}_{k+1}$ is obtained, the previous measurement $\mathbf{z}_k$ must be subtracted and the time between them must be used as the sampling interval in the Kalman filter. Note also that the new measurement $\mathbf{y}_k$ is a measure of the velocity since the difference between two consecutive position measurements is used.

This approach is valid if a new measurement is received closely in time after the previous measurement. If the new measurement is far from the previous (e.g. more than a second), the approach outlined here should not be used. In such a situation, the new position measurement is used directly to update the state estimate with the original system (7.5), where $\mathbf{w}_c$ is assumed to be white with covariance matrix $\mathbf{R}_w$. This is because correlated noise in the measurements is more unlikely if the time between measurements is more than a second.

The motivation behind this approach is to whiten the measurement noise and consequently improve consistency. Both of the approaches described in Section 7.2 are investigated experimentally in Section 7.5

7.3 UAV Navigation using the Multiplicative Extended Kalman Filter

Navigation is not the main focus in this chapter, but is addressed for the reader to understand the details of the tracking system presented in Section 7.4. In general, the goal of the navigation system is to estimate the position, velocity and attitude of the UAV as stated in Section 5.5. The navigation system can obviously also estimate other states such as acceleration, angular velocity and inertial sensor bias. Navigation for fixed-wing UAVs is covered in e.g. [36, 37, 51, 107].

A requirement for compensation of navigation uncertainty is to know something about the quality of the navigation estimates. Kalman filters and other stochastic filters usually represent the uncertainty with an estimate of the covariance matrix, and it is necessary to know the covariance to compensate for the uncertainty. Therefore, nonlinear observers are not suitable for this purpose since they normally don't provide information about the covariance.

It is assumed that the UAV is equipped with an inertial measurement unit (IMU) providing measurements of the acceleration (specific force) and angular rate at high frequency. In addition, it is assumed that a global navigation satellite system (GNSS) provides information about the position. Moreover, two independent GNSS receivers with RTK capability are assumed available as in Section 5.5. RTK functionality may not be required, but increases the estimation accuracy, particularly when the baseline between the antennas is small. Using multiple receivers is advantageous when heading measurements are unavailable. If one receiver is used, non-zero angular and linear velocities are needed.

The error-state formulation of the Kalman filter is utilized in the design of the navigation system. The IMU is used to predict a nominal navigation estimate and the error states are estimated through measurements of position from the GNSS receivers. The attitude error is estimated using a multiplicative extended Kalman filter (MEKF) where the modified Rodriguez parameters [77] represent the attitude error with the minimal amount of parameters, and is used to avoid singularities. This ensures that the covariance matrix of the attitude error has the minimal dimension and remains nonsingular. Note that Euler angles could be a feasible parametrization since the singularity at a pitch angle of 90 deg [7] is unlikely to be an issue for fixed-wing UAVs.

7.3.1 Preliminaries

The coordinate frames of interest were described in Section 5.2. In this section, the unit quaternion is used as attitude parameterization. The nominal attitude is represented with the Hamiltonian representation of the unit quaternion [106]. The relationship between the quaternion and the corresponding rotation matrix is

$$\mathbf{R}(\mathbf{q}) = (q_w^2 - \mathbf{q}_v^\top \mathbf{q}_v)\mathbf{I} + 2\mathbf{q}_v\mathbf{q}_v^\top + 2q_w\mathbf{S}(\mathbf{q}_v) \tag{7.13}$$

where q_w is the real part of the quaternion, $\mathbf{q}_v$ is the vector part of the quaternion, $\mathbf{I}$ is the identity matrix and $\mathbf{S}(\cdot)$ is the skew-symmetric matrix representing a cross-product [29]. The quaternion product used for composition of rotations is defined as

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} q_{1,w}q_{2,w} - \mathbf{q}_{1,v}^\top \mathbf{q}_{2,v} \\ q_{1,w}\mathbf{q}_{2,v} + q_{2,w}\mathbf{q}_{1,v} + \mathbf{S}(\mathbf{q}_{1,v})\mathbf{q}_{2,v} \end{bmatrix} \tag{7.14}$$

where $\mathbf{q}_1$ and $\mathbf{q}_2$ are unit quaternions. In the inertial navigation problem, the true attitude $\mathbf{q}$ is composed as

$$\mathbf{q} = \hat{\mathbf{q}} \otimes \delta\mathbf{q} \tag{7.15}$$

where $\delta\mathbf{q}$ is the attitude error and $\hat{\mathbf{q}}$ is the nominal estimate of the true quaternion. To avoid over-parametrization, the error state is represented as four times the Modified Rodrigues parameters (MRP) $\delta\mathbf{a}$ [77]:

$$\delta\mathbf{q} = \delta\mathbf{a} = 4\delta\mathbf{a}_{mrp} = 4\frac{\delta\mathbf{q}_v}{1 + \delta q_w} \tag{7.16}$$

The UAV position is also a combination of a nominal state estimate $\hat{\mathbf{p}}$ and an error state $\delta\mathbf{p}$. The true position is expressed as

$$\mathbf{p} = \hat{\mathbf{p}} + \delta\mathbf{p} \tag{7.17}$$

where the reference frame is neglected for simplicity.

7.3.2 Kinematics

The kinematic equations are used to design the filter equations in the state estimator. They are based on the strapdown equations for the states of interest, which are:

- The UAV NED positions $\mathbf{p}_{nb}^n$
- The UAV NED velocities $\mathbf{v}_{nb}^n$
- The attitude between {n} and {b} represented by $\mathbf{q}_b^n$

Moreover, the angular velocity of {b} relative to {n} is $\boldsymbol{\omega}_{nb}^b$. The kinematic equations are:

$$\begin{aligned} \dot{\mathbf{p}}_{nb}^n &= \mathbf{v}_{nb}^n \\ \dot{\mathbf{v}}_{nb}^n &= \mathbf{R}_b^n(\mathbf{q}_b^n)\mathbf{f}_{nb}^b + \mathbf{g}^n \\ \dot{\mathbf{q}}_b^n &= \frac{1}{2}\mathbf{q}_b^n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{nb}^b)\mathbf{q}_b^n \end{aligned} \tag{7.18}$$

where $\mathbf{f}^b_{nb}$ is the specific force acting on the UAV and $\mathbf{g}^n = [0, 0, g]^\top$ is the gravity vector in NED. Moreover, $\mathbf{\Omega}(\boldsymbol{\omega}^b_{nb})$ is

$$\mathbf{\Omega}(\boldsymbol{\omega}^b_{nb}) = \begin{bmatrix} 0 & -(\boldsymbol{\omega}^b_{nb})^\top \\ \boldsymbol{\omega}^b_{nb} & -\mathbf{S}(\boldsymbol{\omega}^b_{nb}) \end{bmatrix} \tag{7.19}$$

It is assumed that NED is inertial in these equations. This is reasonable as long as the rotation rate of the earth is small compared to the errors in the angular rate sensors. The IMUs used in this thesis typically have such behavior. For higher-level IMUs, this is not a good approximation.

7.3.3 Inertial measurement unit models

The nominal state estimates are based on measurements from inertial measurement units. They typically measure specific force with accelerometers and angular rate with gyros. The following equations are common for modeling measurements from IMUs [29], and are used here:

$$\begin{aligned} \mathbf{f}^b_{\text{imu}} &= \mathbf{R}^b_n(\mathbf{q}^b_n)(\dot{\mathbf{v}}^n_{nb} - \mathbf{g}^n) + \mathbf{b}^b_{\text{acc}} + \mathbf{w}^b_{\text{acc}} \\ \boldsymbol{\omega}^b_{\text{imu}} &= \boldsymbol{\omega}^b_{nb} + \mathbf{b}^b_{\text{ars}} + \mathbf{w}^b_{\text{ars}} \end{aligned} \tag{7.20}$$

where $\mathbf{b}^b_{\text{acc}}$ and $\mathbf{b}^b_{\text{ars}}$ are biases on the accelerometers and angular rate sensors, respectively. $\mathbf{w}^b_{\text{acc}}$ and $\mathbf{w}^b_{\text{ars}}$ are zero-mean noise terms.

7.3.4 Error-state navigation system

The fundamental idea of the error-state formulation is to use the high-frequency INS to estimate the nominal state and use low-frequency measurements from GNSS to estimate the error states in a multiplicative Kalman filter. The error-states are used to correct the nominal states whenever they are updated. [106] describes the error-state Kalman filter in detail.

The nominal state vector is

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{p}}^n_{nb}, & \hat{\mathbf{v}}^n_{nb}, & \hat{\mathbf{q}}^n_b, & \hat{\mathbf{b}}^b_{\text{acc}}, & \hat{\mathbf{b}}^b_{\text{ars}} \end{bmatrix}^\top \tag{7.21}$$

and includes the position, velocity and attitude of the UAV. Moreover, the biases in the IMUs are also included and must be estimated. The error-state vector is

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{p}^n_{nb}, & \delta\mathbf{v}^n_{nb}, & \delta\mathbf{a}, & \delta\mathbf{b}^b_{\text{acc}}, & \delta\mathbf{b}^b_{\text{ars}} \end{bmatrix}^\top \tag{7.22}$$

where each element in the nominal state vector $\hat{\mathbf{x}}$ has a corresponding error state. Note that the error state for the attitude is the four times modified Rodrigues parameters as described in Section 7.3.1. The true state is related to the nominal state and the error state by the following equation:

$$\mathbf{x} = \hat{\mathbf{x}} \otimes \delta\mathbf{x} \tag{7.23}$$

where $\otimes$ is the generic composition used to describe the relationship between the true, nominal and error states. The generic composition can e.g. be the quaternion product for the attitude, and a simple summation for the linear states as shown in Section 7.3.1.

The kinematic equations used to estimate the nominal states are based on the strapdown and IMU equations:

$$\begin{aligned}
\dot{\hat{\mathbf{p}}}^n_{nb} &= \hat{\mathbf{v}}^n_{nb} \\
\dot{\hat{\mathbf{v}}}^n_{nb} &= \mathbf{R}^n_b(\hat{\mathbf{q}}^n_b)(\mathbf{f}^b_{\text{imu}} - \hat{\mathbf{b}}^b_{\text{acc}}) + \mathbf{g}^n \\
\dot{\hat{\mathbf{q}}}^n_b &= \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}^b_{\text{imu}} - \hat{\mathbf{b}}^b_{\text{ars}})\hat{\mathbf{q}}^n_b \\
\dot{\hat{\mathbf{b}}}^b_{\text{acc}} &= -\mathbf{T}^{-1}_{\text{acc}}\hat{\mathbf{b}}^b_{\text{acc}} \\
\dot{\hat{\mathbf{b}}}^b_{\text{ars}} &= -\mathbf{T}^{-1}_{\text{ars}}\hat{\mathbf{b}}^b_{\text{ars}}
\end{aligned} \tag{7.24}$$

where $\mathbf{T}_{\text{acc}}$ and $\mathbf{T}_{\text{ars}}$ are time constant matrices used to model the bias processes.

The error state is estimated with two independent measurements of the NED positions acquired with GNSS. A single position measurement can be expressed as

$$\begin{aligned}
\mathbf{z} &= \mathbf{p}^n_{\text{GNSS}} \\
&= \mathbf{p}^n_{nb} + \mathbf{R}^n_b\mathbf{r}^b \\
&= \hat{\mathbf{p}}^n_{nb} + \delta\mathbf{p}^n_{nb} + \hat{\mathbf{R}}^n_b(\mathbf{I} + \mathbf{S}(\delta\mathbf{a}))\mathbf{r}^b \\
&= \hat{\mathbf{p}}^n_{nb} + \delta\mathbf{p}^n_{nb} + \hat{\mathbf{R}}^n_b\mathbf{r}^b - \hat{\mathbf{R}}^n_b\mathbf{S}(\mathbf{r}^b)\delta\mathbf{a}
\end{aligned} \tag{7.25}$$

where $\mathbf{r}^b$ is the lever arm from the origin of the body-fixed frame to the GNSS antenna. Consequently, the measurement matrix with respect to the error states is

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{R}(\hat{\mathbf{q}}^n_b)\mathbf{S}(\mathbf{r}_{b,1}) & \mathbf{0}_{3\times6} \\ \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{R}(\hat{\mathbf{q}}^n_b)\mathbf{S}(\mathbf{r}_{b,2}) & \mathbf{0}_{3\times6} \end{bmatrix} \tag{7.26}$$

where $\mathbf{0}_{m\times n}$ is a matrix of zeros with m rows and n columns, $\mathbf{I}_3$ is the identity matrix of dimension 3 and $\mathbf{r}_{b,i}$ is the lever arm of receiver i. Note that the Jacobian with respect to the error states must be used in the filter as described in [106]. The equations presented here are discretized for the experimental implementation used in the results.

7.4 Target Tracking using the Schmidt-Kalman Filter

This section looks into tracking of floating objects using the Schmidt-Kalman filter (SKF). The objective is to derive a tracking formulation where the navigation uncertainty is incorporated in the tracking filter. This can be achieved with the SKF and methods presented previously in this thesis. The first part of this section gives an introduction to the SKF before specific equations are derived. The final part of this section links the SKF together with the tracking formulation using the EKF described in Section 6.3.

7.4.1 Introduction to the Schmidt-Kalman filter

The accuracy of tracking systems depends on the accuracy of the position and orientation of the tracking sensor, which normally are assumed perfectly known. Even though this might be reasonable in some applications (e.g. when only the relative position is of interest), it limits the accuracy and consistency of tracking systems in other applications. This is particularly relevant when targets are tracked in an Earth-fixed coordinate frame from an agile moving platform.

Simultaneous localization and mapping (SLAM) [13] is closely related to tracking in the presence of navigation uncertainty. The objective in SLAM is to simultaneously estimate the pose of a vehicle and make a map of the surroundings. A common assumption in SLAM is that landmarks are static. Thus, the regular SLAM architecture cannot cover a situation with moving objects as will be experienced in a tracking scenario. Nevertheless, SLAM could potentially be extended to consider moving landmarks as suggested in [116]. However, a fundamental part of SLAM is that target observations are allowed to influence the navigation estimates of the vehicle. For this to work, it is critical that the observations are reliable and that the motion models of the targets are precise. Erroneous data association and poor target measurements can degrade the navigation estimates and in worst-case lead to conditions that affect the airworthiness of the UAV. Moreover, measurements from optical sensors can be inaccurate and only contain bearing information. Thus, letting the target tracking system directly affect the navigation system is not desirable in this type of application, especially since the navigation estimates are estimated accurately by other sensors.

A more attractive option is to use a framework where the navigation system of the UAV operates independently, but where the uncertainty of the navigation estimates is allowed to affect the tracking system. This can be achieved with the SKF [27], where information is allowed to flow from the navigation system to the tracking system, but not in the opposite way. The SKF is designed so that the target state vector is augmented with nuisance parameters that represent errors in the navigation states. These error states are not estimated in the tracking filter, but their uncertainty influences the tracking system. The uncertainty of the nuisance parameters is extracted from the covariance matrix in the navigation system.

The main issue when neglecting the influence of navigation errors is inconsistency in the tracking filter. Normally this is observed as the filter being optimistic and estimating a covariance that is much lower than the corresponding mean square error. This was experienced in Section 6.4.2. It is possible to counteract this behavior by increasing the covariance matrices for the noise, but this is an ad-hoc technique that disguises the underlying problem and must be tuned towards a particular scenario or set of data. Therefore, it is hard to generalize this strategy to fit new and unknown data.

The SKF is a sub-optimal approach in contrast to SLAM where information can flow from the tracking system to the navigation system as well. Nevertheless, the SKF is more robust when considering model mismatch and erroneous data association so it is preferred in this research. The SKF formulation for target tracking is also described in [86, 124].

7.4.2 The Schmidt-Kalman filter

Let $\mathbf{x}_k^t$ denote the target state vector at time index k. It includes the position and velocity of the target in the north-east frame. The usual way to estimate the tracking states is to use a regular Kalman filter as described in Section 3.2. The regular KF equations do not account for uncertainty in parameters affecting the motion and measurement models. Instead, all of the uncertainty is modeled as additive white noise. This is not a reliable way to model the uncertainty when navigation errors are present. To compensate for errors in the sensor pose, the SKF is utilized.

General system model

Consider a situation where the sensor pose is defined as

$$\mathbf{x}^o = \hat{\mathbf{x}}^o \otimes \delta\mathbf{x}^o \tag{7.27}$$

where $\hat{\mathbf{x}}^o$ represents a nominal known estimate of the sensor pose and $\delta\mathbf{x}^o$ is the error in the sensor pose (follows the notation of the error-state Kalman filter). Moreover, assume that the uncertainty of the error state is known through a covariance matrix $\mathbf{P}^o$. The motivation behind this formulation is to augment the tracking system with $\delta\mathbf{x}^o$, and let $\mathbf{P}^o$ affect the tracking estimates. Only the error states that directly affect the tracking system are considered. The augmented state transition model is written as

$$\begin{bmatrix} \mathbf{x}_k^t \\ \delta\mathbf{x}_k^o \end{bmatrix} = \begin{bmatrix} \mathbf{F}_k^t & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_k^o \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1}^t \\ \delta\mathbf{x}_{k-1}^o \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^t \\ \mathbf{v}_k^o \end{bmatrix} \tag{7.28}$$

where $\mathbf{F}_k^t$ is the motion model of the target (e.g. the CV model), and $\mathbf{F}_k^o$ is the motion model of the error states. $\mathbf{v}_k^t$ and $\mathbf{v}_k^o$ are additive white noise for the target and navigation uncertainty, respectively. The covariance matrix of the augmented system is expressed as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^t & \mathbf{C} \\ \mathbf{C}^\top & \mathbf{P}^o \end{bmatrix} \tag{7.29}$$

where $\mathbf{P}^t$ is the target covariance (estimated in the filter) and $\mathbf{C}$ is the cross-covariance. Note that the state transition models of the error state and target are assumed to be uncoupled because the target behavior should not affect the trajectory of the error state and vice versa.

Assume that the augmented system has a measurement model that depends on both the target estimates and the navigation states of the UAV:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}^t, \hat{\mathbf{x}}^o, \delta\mathbf{x}^o) + \mathbf{w} \tag{7.30}$$

where $\mathbf{z}$ is the measurement vector and $\mathbf{w}$ is additive zero-mean white Gaussian noise. The nominal navigation estimate $\hat{\mathbf{x}}^o$ is known perfectly from the navigation system. When the measurement model is nonlinear, the Jacobian of (7.30) is used in the EKF. The idea in the SKF is to calculate the Jacobian with respect to both the target states $\mathbf{x}^t$ and the error states $\delta\mathbf{x}^o$. Therefore, the linearized measurement model is expressed as

$$\mathbf{z}_k = \mathbf{h}(\hat{\mathbf{x}}_k^t, \hat{\mathbf{x}}_k^o, \delta\mathbf{x}_k^o) + \begin{bmatrix}\mathbf{H}_k^t & \mathbf{H}_k^o\end{bmatrix} \begin{bmatrix}\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t \\ \delta\mathbf{x}_k^o - \delta\mathbf{x}_k^o\end{bmatrix} + \mathbf{w}_k + \text{h. o. t.} \tag{7.31}$$

where $(\hat{\mathbf{x}}_k^t, \hat{\mathbf{x}}_k^o, \delta\mathbf{x}_k^o)$ is the linearization point and given by the most recent estimates. $\mathbf{H}_k^t$ and $\mathbf{H}_k^o$ are measurement Jacobians differentiated with respect to the target state vector and the error state, respectively.

Update equations for the Schmidt-Kalman filter

A general system model was considered in (7.27) - (7.31). This part derives specific equations for the SKF using these equations. How to define the system model for tracking of floating objects is described in the next section. The SKF equations are based on the augmented system in (7.28)-(7.29), and the KF and EKF equations from (3.5) and (3.7), respectively.

The a priori (time) update for the augmented system is

$$\begin{aligned}
\mathbf{x}_{k|k-1}^t &= \mathbf{F}_k^t \mathbf{x}_{k-1|k-1}^t \\
\delta\mathbf{x}_{k|k-1}^o &= \mathbf{F}_k^o \delta\mathbf{x}_{k-1|k-1}^o \\
\mathbf{P}_{k|k-1}^t &= \mathbf{F}_k^t \mathbf{P}_{k-1|k-1}^t (\mathbf{F}_k^t)^\top + \mathbf{Q}_k^t \\
\mathbf{C}_{k|k-1} &= \mathbf{F}_k^t \mathbf{C}_{k-1|k-1} (\mathbf{F}_k^o)^\top
\end{aligned} \tag{7.32}$$

where $\mathbf{Q}_k^t$ is the covariance matrix of the process noise in the target motion model. The covariance of the navigation uncertainty, $\mathbf{P}^o$, is estimated in the navigation system. The Kalman gain in the augmented filter is divided into two components, one that affects the target estimates and one that affects the navigation error:

$$\mathbf{K}_k = \begin{bmatrix}\mathbf{K}_k^t \\ \mathbf{K}_k^o\end{bmatrix} = \mathbf{P}_{k|k-1} \begin{bmatrix}(\mathbf{H}_k^t)^\top \\ (\mathbf{H}_k^o)^\top\end{bmatrix} \mathbf{S}_k^{-1} \tag{7.33}$$

where the innovation covariance is defined as

$$\begin{aligned}\mathbf{S}_k &= \begin{bmatrix}\mathbf{H}^t_k & \mathbf{H}^o_k\end{bmatrix} \mathbf{P}_{k|k-1} \begin{bmatrix}(\mathbf{H}^t_k)^\top \\ (\mathbf{H}^o_k)^\top\end{bmatrix} + \mathbf{R}_k \\ &= \begin{bmatrix}\mathbf{H}^t_k & \mathbf{H}^o_k\end{bmatrix} \begin{bmatrix}\mathbf{P}^t_{k|k-1} & \mathbf{C}_{k|k-1} \\ \mathbf{C}^\top_{k|k-1} & \mathbf{P}^o_{k|k-1}\end{bmatrix} \begin{bmatrix}(\mathbf{H}^t_k)^\top \\ (\mathbf{H}^o_k)^\top\end{bmatrix} + \mathbf{R}_k \\ &=\mathbf{H}^t_k\mathbf{P}^t_{k|k-1}(\mathbf{H}^t_k)^\top + \mathbf{H}^t_k\mathbf{C}_{k|k-1}(\mathbf{H}^o_k)^\top + \\ &\quad \mathbf{H}^o_k\mathbf{C}^\top_{k|k-1}(\mathbf{H}^t_k)^\top + \mathbf{H}^o_k\mathbf{P}^o_{k|k-1}(\mathbf{H}^o_k)^\top + \mathbf{R}_k\end{aligned} \tag{7.34}$$

and $\mathbf{R}_k$ is the covariance of the measurement noise. The SKF formulation appears by choosing a suboptimal gain where $\mathbf{K}^o_k$ is forced to be zero so that the error state $\delta\mathbf{x}^o$ is constant over the measurement update. The target gain is

$$\mathbf{K}^t_k = (\mathbf{P}^t_{k|k-1}(\mathbf{H}^t_k)^\top + \mathbf{C}_{k|k-1}(\mathbf{H}^o_k)^\top)\mathbf{S}^{-1}_k \tag{7.35}$$

and the a posteriori state estimates are given as

$$\begin{aligned}\mathbf{x}^t_{k|k} &= \mathbf{x}^t_{k|k-1} + \mathbf{K}^t_k(\mathbf{z}_k - \mathbf{H}^t_k\mathbf{x}^t_{k|k-1}) \\ \mathbf{x}^o_{k|k} &= \mathbf{x}^o_{k|k-1}\end{aligned} \tag{7.36}$$

The a posteriori estimates of the covariance and the cross-covariance differ from the regular Kalman filter because a suboptimal gain is chosen. Therefore, it is not possible to use the simplified formula for the a posteriori estimate. The Joseph form, which is valid for any gain, is used instead:

$$\begin{aligned}\mathbf{P}_{k|k} = &\left(\mathbf{I} - \begin{bmatrix}\mathbf{K}^t_k \\ \mathbf{0}\end{bmatrix} \begin{bmatrix}\mathbf{H}^t_k & \mathbf{H}^o_k\end{bmatrix}\right) \begin{bmatrix}\mathbf{P}^t_{k|k-1} & \mathbf{C}_{k|k-1} \\ \mathbf{C}^\top_{k|k-1} & \mathbf{P}^o_{k|k-1}\end{bmatrix} \\ &\left(\mathbf{I} - \begin{bmatrix}\mathbf{K}^t_k \\ \mathbf{0}\end{bmatrix} \begin{bmatrix}\mathbf{H}^t_k & \mathbf{H}^o_k\end{bmatrix}\right)^\top + \begin{bmatrix}\mathbf{K}^t_k\mathbf{R}_k(\mathbf{K}^t_k)^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\end{aligned} \tag{7.37}$$

Moreover, the following a posteriori estimates for the target covariance and the cross-covariance are acquired

$$\begin{aligned}\mathbf{P}^t_{k|k} =&(\mathbf{I} - \mathbf{K}^t_k\mathbf{H}^t_k)\mathbf{P}^t_{k|k-1}(\mathbf{I} - \mathbf{K}^t_k\mathbf{H}^t_k)^\top - \\ &\mathbf{C}_{k|k-1}(\mathbf{H}^o_k)^\top(\mathbf{K}^t_k)^\top + \mathbf{K}^t_k\mathbf{H}^t_k\mathbf{C}_{k|k-1}(\mathbf{H}^o_k)^\top(\mathbf{K}^t_k)^\top \\ &- \mathbf{K}^t_k\mathbf{H}^o_k\mathbf{C}^\top_{k|k-1} + \mathbf{K}^t_k\mathbf{H}^o_k\mathbf{C}^\top_{k|k-1}(\mathbf{H}^t_k)^\top(\mathbf{K}^t_k)^\top \\ &+ \mathbf{K}^t_k\mathbf{H}^o_k\mathbf{P}^o_{k|k-1}(\mathbf{H}^o_k)^\top(\mathbf{K}^t_k)^\top + \mathbf{K}^t_k\mathbf{R}_k(\mathbf{K}^t_k)^\top \\ \mathbf{C}_{k|k} =& \mathbf{C}_{k|k-1} - \mathbf{K}^t_k(\mathbf{H}^t_k\mathbf{C}_{k|k-1} + \mathbf{H}^o_k\mathbf{P}^o_k)\end{aligned} \tag{7.38}$$

This concludes the main structure of the SKF. The key step is to use the suboptimal gain. Note that the tracking system must maintain the target estimates and corresponding covariance in addition to the cross covariance. The navigation system is responsible for maintaining the covariance of the error state.

7.4.3 Navigation error models

Target motion and measurement models are discussed in Section 7.4.4. This section discusses models for the error state $\delta\mathbf{x}^o$, which includes errors in the position and orientation of the UAV (and consequently the tracking sensor). The covariance of the navigation error, $\mathbf{P}^o$, is assumed to be available from the navigation system described in Section 7.3. The error state $\delta\mathbf{x}^o$ includes the position and attitude of the UAV

$$\delta\mathbf{x}^o = \begin{bmatrix}\delta\mathbf{p}^o \\ \delta\mathbf{a}^o\end{bmatrix} \tag{7.39}$$

where $\delta\mathbf{p}^o$ is the error state in position and usually parametrized in NED. $\delta\mathbf{a}^o$ is the error state in attitude and represented by the four times modified Rodriguez parameters (see Section 7.3). $\mathbf{F}^o$, in (7.32), is chosen as the identity matrix because the goal is not to estimate the navigation error, but to compensate for the uncertainty. Thus, it is not necessary to update the a priori estimate or the a posteriori estimates of the error state vector. Note that $\mathbf{F}^o$ in the SKF differ from the state transition matrix in the navigation system because it is not desired to estimate the error states in the SKF. This is also a fundamental difference from the SLAM approach where the navigation states are estimated within the same filter as the tracking estimates. The next part discusses typical behavior of the error states and how the equations in Section 7.4.2 are affected by the properties of the error states.

Zero-mean and uncorrelated navigation state errors

One possibility is to assume that the navigation system is consistent, which results in estimation errors that are zero-mean and uncorrelated. This gives uncertainty in the UAV pose which is zero mean. Consequently, the measurement noise in the tracking filter is white. Moreover, the cross-correlation between the target estimates and the navigation error, in (7.29), is zero ($\mathbf{C} = 0$). Therefore, the SKF equations can be simplified and the only effect on the tracking system is a so-called covariance inflation (also called consider covariance), where the innovation covariance is reduced to

$$\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k^t\mathbf{P}_{k|k-1}^t(\mathbf{H}_k^t)^\top + \mathbf{H}_k^o\mathbf{P}_{k|k-1}^o(\mathbf{H}_k^o)^\top \tag{7.40}$$

and the a posteriori target covariance estimate is

$$\begin{aligned}\mathbf{P}_{k|k}^t &= (\mathbf{I} - \mathbf{K}_k^t\mathbf{H}_k^t)\mathbf{P}_{k|k-1}^t(\mathbf{I} - \mathbf{K}_k^t\mathbf{H}_k^t)^\top \\ &\quad + \mathbf{K}_k^t\mathbf{H}_k^o\mathbf{P}_{k|k-1}^o(\mathbf{H}_k^o)^\top(\mathbf{K}_k^t)^\top + \mathbf{K}_k^t\mathbf{R}_k(\mathbf{K}_k^t)^\top\end{aligned} \tag{7.41}$$

where the second term is the covariance inflation. The practical effect of this design is quite similar to the methodology in [47] and in Section 7.2.

Constant navigation state bias

A second and more likely situation is navigation errors that introduce a constant bias in the target measurements. This can be because of a misalignment in the mounting between the IMU and the camera. It can also be caused by a constant bias that is not accounted for or estimated correctly in the navigation system. Moreover, a real navigation system is not based on ideal sensors so it is likely that estimation errors are correlated in time [46]. The covariance $\mathbf{P}^o$ can change in such a situation even though $\mathbf{F}^o$ is the identity matrix. Moreover, this gives a nonzero cross-covariance between the target estimates and the navigation errors, and introduces a correlation. The constant bias formulation gives the equations in Section 7.4.2 where $\mathbf{F}^o$ is the identity matrix.

Other navigation state error models

Error states that behave like slowly-varying biases or higher-order models can also occur. Nevertheless, if it is assumed that the navigation system is initialized properly and has been running for a few minutes before the tracking objective starts, it is likely that sensor biases and other disturbances (in the navigation filter) have converged. Large variations in the navigation error are not expected after the initial period. Therefore, it is not considered to be beneficial to model the navigation error with higher-order models, especially without having a strong indication about that being necessary. That could for instance be the case in dead reckoning due to GNSS dropout.

In the rest of this chapter the error states are modeled as a constant bias. It also covers a situation where the navigation error varies slowly. Rapid changes in the error states are not expected so this is a reasonable choice. This design allows the uncertainty of the error states to affect the tracking system and a correlation between the tracking system and navigation system is maintained.

7.4.4 Tracking architecture using the Schmidt-Kalman filter

The general idea of tracking in the presence of navigation uncertainty was derived in Section 7.4.2. This section defines specific models for tracking of floating objects using monocular optical sensors. This includes motion and measurement models and is based on the work in Chapter 6.

In this chapter, the target motion model is simply chosen as the CV model described in Section 6.2.1. It worked well in the case studies in the previous chapter and this tracking architecture does not require changes to this part of the system. Other motion models can obviously also be used within the SKF formulation. The measurement model for the SKF is based on the nonlinear tracking formulation in Section 6.3. This is because it is necessary to find the Jacobian with respect to the error states of the navigation system. Thus, the measurement model is equal to the one in Section 6.3, but parametrized with extra parameters representing the error states. The measurement model is based on the pinhole camera model described in Section 6.3.1:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z_t^c} \begin{bmatrix} x_t^c \\ y_t^c \end{bmatrix} \tag{7.42}$$

The camera-fixed position of a target detected in a thermal image is related to the camera pose and target position by the following equation:

$$\mathbf{p}_{ct}^c = \mathbf{R}_n^c (\mathbf{p}_{nt}^n - \mathbf{p}_{nb}^n) \tag{7.43}$$

where $\mathbf{R}_n^c$ is the rotation matrix between {c} and {n}, and depends on the orientation of the camera with respect to NED. $\mathbf{p}_{nt}^n$ is the target position in NED and $\mathbf{p}_{nb}^n$ is the NED positions of the UAV. It is necessary to rewrite (7.43) to include the error states before the Jacobian is calculated. This is achieved by rewriting the camera-fixed coordinates of the target as

$$\mathbf{p}_{ct}^c = \mathbf{R}_b^c \mathbf{R}_{\hat{b}}^b \mathbf{R}_n^{\hat{b}} (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \tag{7.44}$$

where $\mathbf{R}_b^c$ is a known and constant rotation matrix between the camera-fixed frame and the body-fixed frame (as long as the gimbal has a static orientation). $\mathbf{R}_{\hat{b}}^b$ is a rotation matrix given by the error state of the attitude, and $\mathbf{R}_n^{\hat{b}}$ is the nominal rotation matrix between the body frame and NED. $\hat{\mathbf{p}}_{nb}^n$ is the nominal NED positions of the UAV estimated by the navigation system and $\delta \mathbf{p}_{nb}^n$ is the error state in the UAV position. Note that the rotation matrix based on the error state ($\mathbf{R}_{\hat{b}}^b$) represents the difference between the estimated body frame (by the nominal state) and the true body frame.

To simplify the Jacobian calculation, it is beneficial to rewrite (7.44) in the following manner:

$$\begin{aligned} \mathbf{p}_{ct}^c &= \mathbf{R}_b^c \mathbf{R}_{\hat{b}}^b \mathbf{R}_n^{\hat{b}} (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \\ &= \mathbf{R}_b^c \underbrace{(\mathbf{I} + \mathbf{S}(\delta \mathbf{a}))^\top}_{\mathbf{R}_{\hat{b}}^b} (\mathbf{R}_{\hat{b}}^n)^\top (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \\ &= \mathbf{R}_b^c (\mathbf{R}_{\hat{b}}^n)^\top (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \\ &\quad - \mathbf{R}_b^c \mathbf{S}(\delta \mathbf{a}) (\mathbf{R}_{\hat{b}}^n)^\top (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \\ &= \mathbf{R}_b^c (\mathbf{R}_{\hat{b}}^n)^\top (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \\ &\quad + \mathbf{R}_b^c \mathbf{S} \Big((\mathbf{R}_{\hat{b}}^n)^\top (\mathbf{p}_{nt}^n - \hat{\mathbf{p}}_{nb}^n - \delta \mathbf{p}_{nb}^n) \Big) \delta \mathbf{a} \end{aligned} \tag{7.45}$$

The final two expressions are equal, but both can be used when finding the Jacobian. The last expression is beneficial when differentiating the measurement model with respect to the error state of the attitude ($\delta \mathbf{a}$) and the expression before is beneficial when finding the Jacobian with respect to the error state of the position. Note that (7.45) needs to be inserted in (7.42) before the Jacobian is calculated. Finding the Jacobian theoretically is challenging, but is achievable with a computer program. Nevertheless, for real-time calculation, it is typically more efficient to find the Jacobian numerically.

The Jacobian is calculated with respect to the target position ($\mathbf{p}^n_{nt}$), the target velocity ($\mathbf{v}^n_{nt}$), the UAV position error state ($\delta\mathbf{p}^n_{nb}$) and the UAV attitude error state ($\delta\mathbf{a}$). Hence, the Jacobian has dimension 2×10 (two rows and ten columns) where column three and four only have zeros because the measurement model does not depend on the target velocity. Removing the error states from the SKF leads to a regular EKF as described in Section 6.3.

7.5 Case Studies

This section presents four independent case studies covering the topics discussed in this chapter. Case study one and two investigate tracking of floating objects using georeferencing with compensation of correlated noise in the motion and measurement models as described in Section 7.2. These results are based on the work in [47]. The third and fourth case studies investigate the effect of compensating for navigation uncertainty with the SKF as described in Section 7.4. It is based on the work in [45]. All of the results are based on experimental data gathered in several flight experiments.

7.5.1 Case study 1 - Tracking of high-speed vessel with correlated noise

This case study investigates the effectiveness of the tracking systems presented in Section 7.2 for a high-speed vessel. The thermal camera payload described in Section 2.3 and Appendix A was used to collect data for this case study. The marine vessel shown in Figure 7.1 was used as target.

The target was detected in 264 images and the detections are divided into three separate periods with a significant amount of time in between the periods. The floating object detection algorithm from Section 4.3 was used to detect the target automatically. The target was moving at high speed while trying to maintain a constant course. The speed varied somewhat because of varying water and wave resistance. The gathered data cover a time period of approximately 3 minutes and are used to illustrate how well a high speed target can be tracked with a limited amount of measurements and the methodology in Section 7.2. Data association is neglected and all measurements are used as target detections. No more than one detection occurred in a single image. The same data were used in Section 6.4.2.

Three different tracking filters are evaluated in this case study. Georeferencing of the image detections is used to acquire measurements and all filters use the same set of measurements. The three tracking filters are:

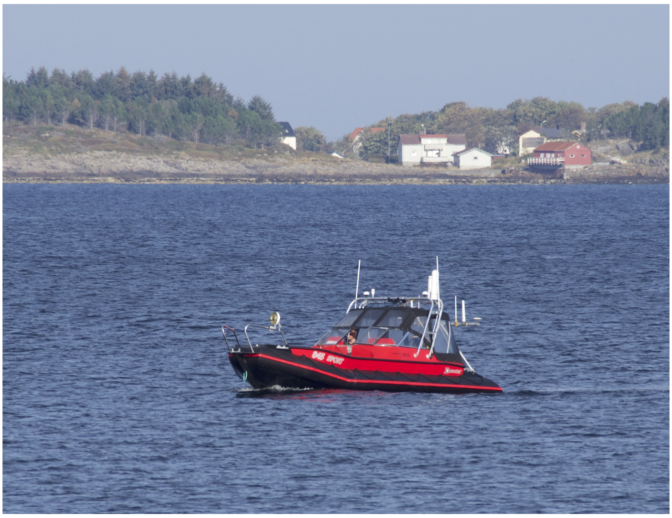


Figure 7.1: The surface vessel used as target in all of the case studies in Chapter 7

- Filter 1 - Kalman filter with the constant velocity motion model (6.1) and (7.6) as measurement model. This is a solution based on the linear formulation described in Section 6.2 and used as benchmark.
- Filter 2 - Kalman filter with the Singer motion model (7.1)-(7.2) and (7.6) as measurement model (note that (7.6) must be extended with 2 columns of zeros because acceleration is a part of the state vector). This filter is used to compare the Singer motion model with the constant velocity motion model during typical vessel motion.
- Filter 3 - Kalman filter with the constant velocity motion model (6.1), and where colored noise is modeled in the measurement model as outlined in (7.7)-(7.12). This filter seeks to minimize the influence of colored noise in the measurements due to navigation errors in the georeferencing algorithm.

The filter parameters are summarized in Table 7.1. The process noise is adjusted to have the same influence in all tracking filters, but Filter 2 is different because the Singer motion model has a distinct set of adjustable parameters. The process noise standard deviation is chosen to be small ($0.05\,\mathrm{m/s^2}$) for the constant velocity motion model because the target tried to maintain constant course and speed, which fits with the model.

The initial velocity is chosen to be zero because no prior information about the velocity exist. The position is initialized with the first measurement. The initial standard deviation in position is 10 m, which is larger than the measurement noise to let the first few measurements correct the initial position quickly. The standard deviation of the initial velocity is chosen much larger (100 m/s) because the velocity is unknown initially. The standard deviation of the measurement noise is chosen to be constant and equal to 7 m in both north and east, which is a typical

Table 7.1: Filter parameters in case study 1

Parameter	**Filter 1**	**Filter 2**	**Filter 3**
$\mathbf{P}(0)$	diag(10^2, 10^2 100^2, 100^2)	diag(10^2, 10^2, 100^2, 100^2, 0.01^2, 0.01^2)	diag(10^2, 10^2, 100^2, 100^2)
$\mathbf{R}$	diag (7^2, 7^2)	diag(7^2, 7^2)	diag(7^2,7^2)
$\mathbf{Q}$	$\sigma_v = 0.05$ (see (6.2))	$a_M = 0.01$, $\alpha = 1/10$, $p_M = 0.05$, $p_0 = 0.7$ (see (7.2)-(7.4))	$\sigma_v = 0.05$ (see (6.2))

georeferencing error at the given altitude. The mean absolute georeferencing error in this case study is approximately 6 m.

The paths of the UAV and the vessel are displayed in Figure 7.2. The UAV operated at an altitude of 350 m. Figure 7.3 shows the target position measured with a single-frequency GPS receiver with update rate of 1 Hz together with the distribution of georeferenced measurements. The measurements are clustered into three different groups because the target was outside the field of view of the camera for extended periods. The UAV flew over and past the target before it took a turn to do a new flyover. This was repeated and gave three periods with target observations. Since the frame rate of the camera is 7.5 Hz, many measurements were collected when the target was within the field of view of the camera.

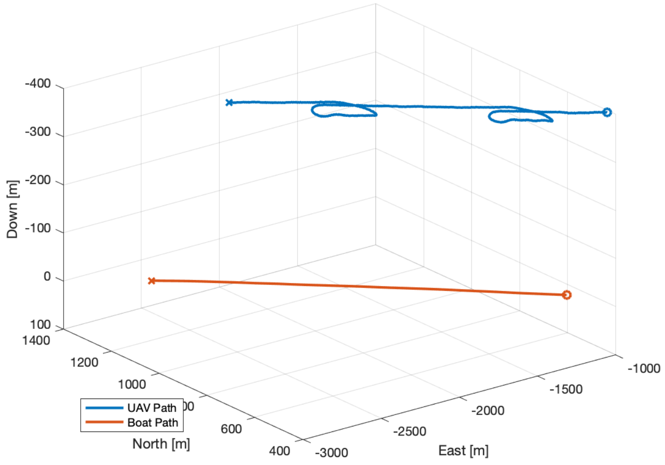


Figure 7.2: The paths of the UAV and the target

Figure 7.4 shows the root-mean-square estimation error in position for all tracking filters together with the measurement errors. The single-frequency GPS receiver

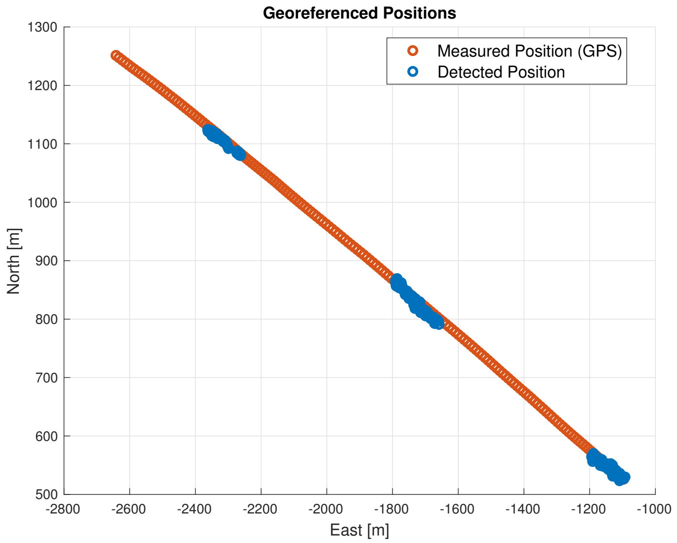


Figure 7.3: Georeferenced positions together with true position measured by GPS

on-board the target is used as a reference for the ground truth position. This is not necessarily a perfect reference so the results should be interpreted with that in mind (especially the estimation error and NEES). The measurement error is obviously equal for all tracking filters since the same measurements are used. Figure 7.4 also shows the times when measurements are available (black crosses). Note that this only corresponds to a total time of approximately 35 s. Therefore, the target estimates are based on prediction for most of the time and explains why the estimation error is larger initially. Filters 1 and 2 have the smallest estimation error overall, but Filter 3 is closing in near the end. The estimation error in position is below 5 m for all filters in the end, and mostly below 10 m when measurements are available.

Figure 7.5 shows the estimated speed and course together with the reference. All filters are able to estimate the speed and course accurately after the initial period. The main difference is that Filter 3 converges more slowly, which may be a drawback with this filter. This is also the reason for why the predicted position of the third filter drifts more than the other two initially. The large initial covariance in velocity is the reason for why the speed and heading oscillate particularly much in the beginning. Filter 3 is oscillating more than the other two, which is explained by the whitening of the measurements. The whitening process increases the covariance of the measurement noise. Consequently, the covariance of the state estimates decreases slower than for the other filters.

Figure 7.6 shows NEES (only in position since an accurate true reference for the north and east velocities is unavailable) and NIS for all filters. The NEES exceeds the upper confidence bound for Filter 1 and Filter 2, and this means that the estimated covariances are smaller than the corresponding estimation error. Thus, the

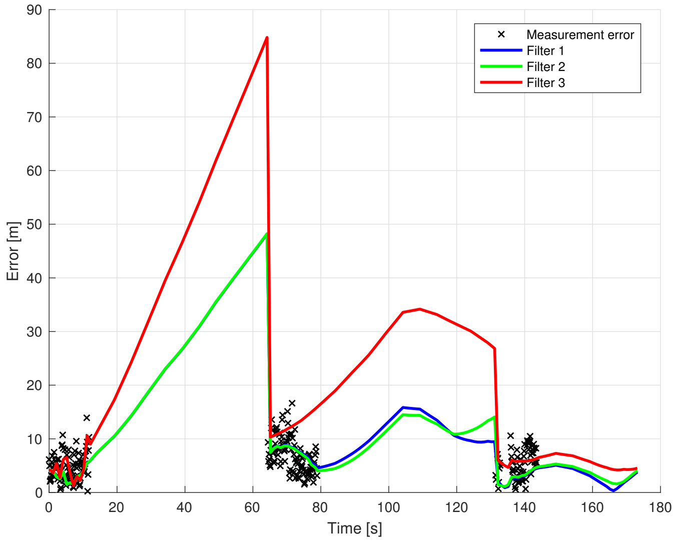


Figure 7.4: The absolute estimation error in position and absolute measurement errors

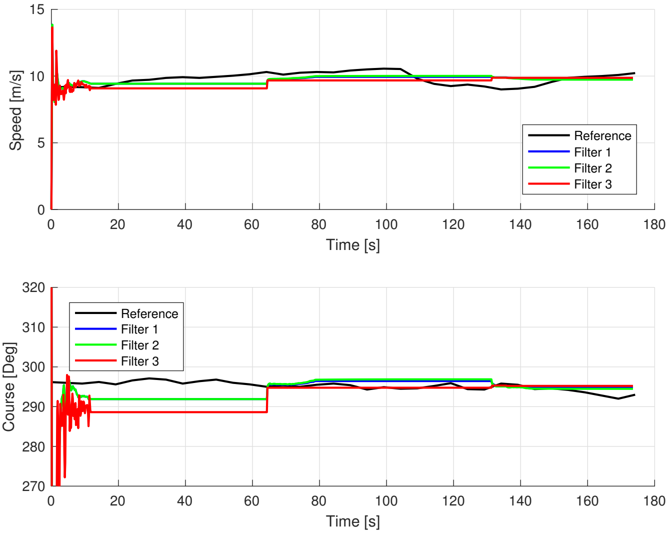


Figure 7.5: Estimated speed and course

filters are too optimistic and assess the estimates to be more trustworthy than they really are. This is in line with the experiences in the previous chapter. The third filter has estimates of the covariance which represent the true error more correctly since NEES is within the bounds, even though the filter perhaps is too pessimistic in the beginning. This is obviously explained by the large initial covariance in velocity. In retrospect, the initial covariance in velocity is unrealistically large and should have been reduced.

The NIS is similar for the different tracking filters and within the confidence bounds. The NIS values are closer to the lower limit and this suggests that the covariance of the measurement noise could have been reduced since NIS increases with a reduction in the measurement noise covariance. However, this would increase NEES even more and lead to covariance estimates that are further from the truth. Since NIS is similar and a significant difference is experienced in NEES, a bias is most likely present in the estimates. This is supported by the values in Table 7.2 which summarizes the average results for the entire tracking period. The mean estimation error in north is more than 8 m from the reference for all filters, and is most likely because of a bias in the measurements caused by colored noise in the navigation errors. Filter 3 is better equipped to handle such a situation as the NEES shows, even though the root-mean-square estimation error is comparable. Note that the mean absolute estimation error in position is larger than the mean absolute measurement error because the estimates drift in periods without measurements as illustrated in Figure 7.4. Thus, the statistics in Table 7.2 must be interpreted with care.

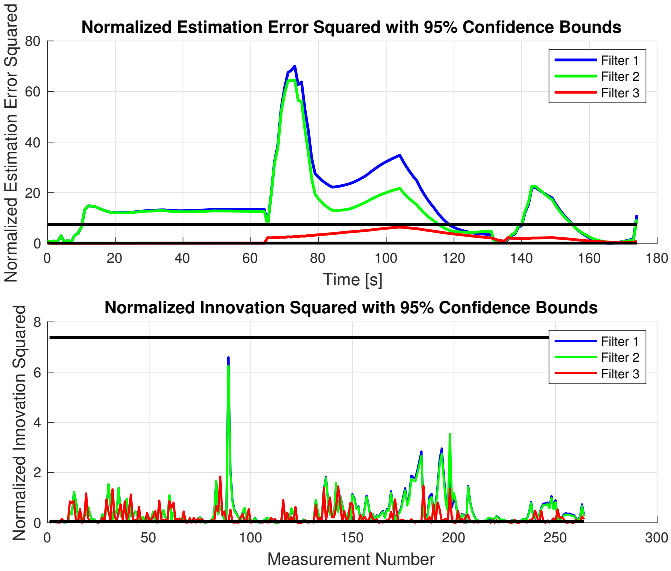


Figure 7.6: The normalized estimation error squared and the normalized innovation squared for the different tracking filters

Figure 7.7 shows the correlation in the innovation sequences and the confidence bounds. The innovation sequence is correlated for all tracking filters, but in a much lower degree for the third filter. This is expected since the intention with the third filter is to whiten the measurement noise and improve filter consistency. Therefore, considering NIS, NEES and the correlation in the innovation sequences, only Filter 3 is close to fulfilling the criteria for consistency. Note also that the

Table 7.2: Average performance metrics in case study 1. ME = mean error. MAE = mean absolute error.

Parameter	**Filter 1**	**Filter 2**	**Filter 3**
ME measurements [north, east]	[−3.77 m, −0.94 m]	[−3.77 m, −0.94 m]	[−3.77 m, −0.94 m]
MAE georeferenced position	5.87 m	5.87 m	5.87 m
ME position estimates [north, east]	[−9.2 m, 1.1 m]	[−8.6 m, 0.9 m]	[−21.6 m, 5.0 m]
MAE estimated position	12.81 m	12.84 m	25.12 m
Mean NEES	15.65	13.08	1.88
Mean NIS	0.46	0.44	0.21

measurement noise must be tuned in a pessimistic manner for Filters 1 and 2 to keep NIS within the bounds, which is important in data association. The measurement noise covariance should obviously be reduced for the third filter, but was chosen to be equal to the other filters to show the effect of this design with respect to NEES. Thus, the third filter could perform better by reducing the measurement noise covariance and this is supported by the low NIS and NEES values.

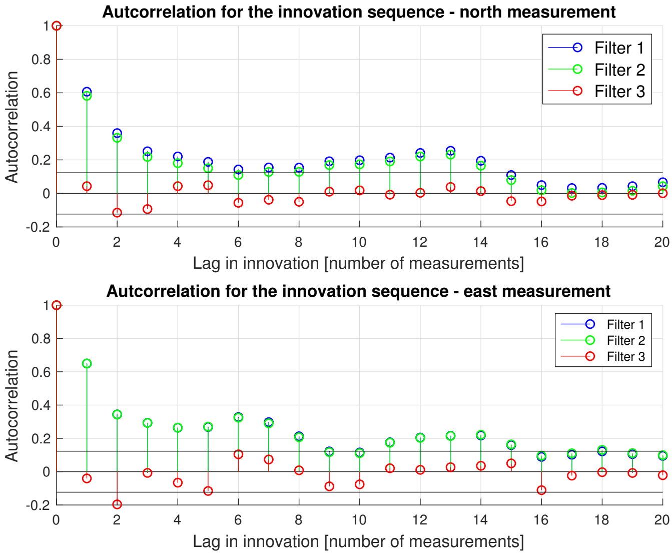


Figure 7.7: The autocorrelation for the innovation sequences

This case study has proved that it is beneficial to model colored noise in the measurement model to mitigate the effect of colored noise. The NEES is within the confidence bounds for Filter 3, and the innovation sequences are almost uncorrelated in contrast to the other solutions. However, the other solutions perform better when considering the estimation error, but this could most likely be avoided by reducing the measurement noise in the third filter as mentioned already. All filters could potentially benefit from increasing the process noise covariance (this reduces NEES), but at a cost of reducing the accuracy of the estimates. The trade-off between accuracy and consistency is particularly challenging without accounting for colored noise, and is the main weakness of the solutions discussed in the previous chapter.

A clear advantage with the Singer motion model has not been demonstrated in this case study, but that is not too surprising because the target did not maneuver much. Moreover, the solution using the Singer motion model performed equally well as the solution based on the CV model so it can be used in tracking of floating objects.

7.5.2 Case study 2 - Tracking of drifting vessel with correlated noise

The second case study investigates the same topic as in the previous case study, but for a slowly-moving target instead. An independent flight experiment was conducted to collect thermal images of the same target. 2200 images of the target were recorded in many periods spread over approximately 25 minutes. All detections were used as target measurements and no more than one object was detected in a single image. 2200 images correspond to a continuous observation of the target in 300 s. Thus, measurements are only available for 300 s in the total tracking period of 1500 s. This case study demonstrates how the system works for a longer time period with varying amount of measurements.

The UAV was controlled by the Pixhawk autopilot during the experiment. The path was chosen manually based on a map where the position of the target was visible. Automatic path design using information from the tracking system could have been utilized [104], but choosing the flight pattern manually was preferred to ensure that the desired data were captured during different maneuvers to give more variety in the data. The same set of filters as in the previous case study are used here.

The filter parameters are summarized in Table 7.3 and will not be discussed as thoroughly as in case study 1. The main difference is the reduction in the covariance of the process noise (the standard deviation is a fraction of 5 smaller), which is explained by the lower velocity of the target. This also explains the lower initial covariance in the velocity. The standard deviation of the measurement noise is increased to 13 m because the georeferencing errors are larger in this case study. This is most likely because the UAV followed a path similar to the figure eight (see Figure 7.8), which gives more variation in the attitude. The UAV followed primarily straight-line segments in the previous case study, which normally reduce

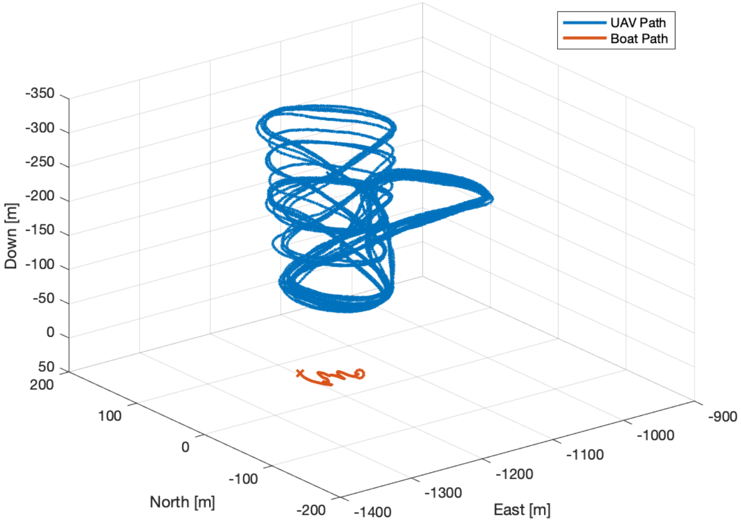


Figure 7.8: The paths of the UAV and target

Table 7.3: Filter parameters in case study 2

Parameter	**Filter 1**	**Filter 2**	**Filter 3**
$\mathbf{P}(0)$	diag(13^2, 13^2, 1, 1)	diag(13^2, 13^2, 1, 1, 0.01, 0.01)	diag(13^2, 13^2, 1, 1)
$\mathbf{R}$	diag (13^2, 13^2)	diag(13^2, 13^2)	diag(13^2, 13^2)
$\mathbf{Q}$	$\sigma_v = 0.01$	$a_M = 0.002$, $\alpha = 1/10$, $p_M = 0.05$, $p_0 = 0.7$	$\sigma_v = 0.01$

the georeferencing error.

The paths of the UAV and the vessel are displayed in Figure 7.8. The UAV operated at a varying altitude of 200 m to 300 m. The target was drifting in the water and moved only a short distance even though the experiment lasted for 25 min. Figure 7.9 shows the same path of the target, which was measured with a single-frequency GPS receiver at 0.5 Hz, together with the position estimates from the tracking filters. The root-mean-square estimation error in position and the measurement errors are displayed in Figure 7.10. The accuracy is more similar in this case study, although Filter 3 still has the largest estimation error on average.

Figure 7.11 shows NEES (in position) and NIS for all filters. The NEES is larger than the upper confidence limit for Filter 1 and Filter 2, and implies that the estimated covariance is lower than the corresponding estimation error. Thus, Filters

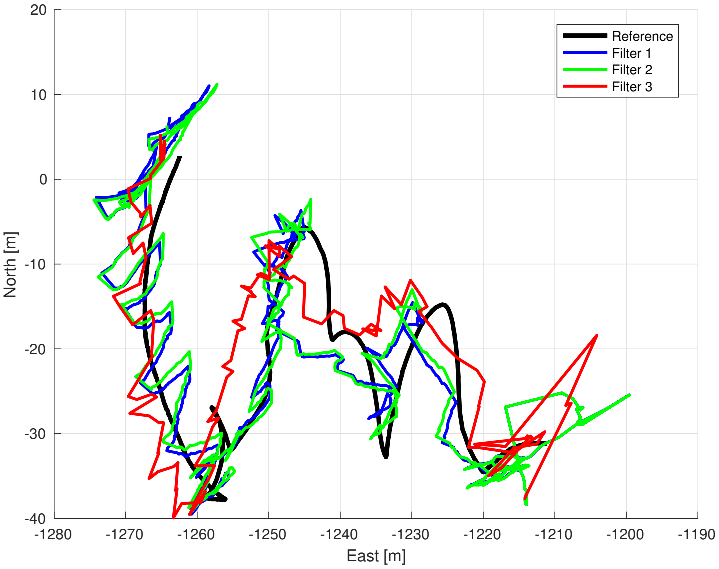


Figure 7.9: Estimated position for all tracking filters

Table 7.4: Average performance metrics in case study 2. ME = mean error. MAE = mean absolute error.

Parameter	**Filter 1**	**Filter 2**	**Filter 3**
ME measurements [north, east]	[1.59 m, −0.32 m]	[1.59 m, −0.32 m]	[1.59 m, −0.32 m]
MAE georeferenced position	9.68 m	9.68 m	9.68 m
ME position estimates [north, east]	[0.8 m, −0.4 m]	[0.9 m, −0.4 m]	[0.2 m, −0.5 m]
MAE estimated position	5.08 m	5.12 m	6.22 m
Mean NEES	11.53	10.12	1.34
Mean NIS	0.69	0.67	0.12

1 and 2 are too optimistic as experienced in the previous case study. The NIS is similar for Filters 1 and 2, but smaller for the third filter. This is supported by the values in Table 7.4, which summarizes the average results in case study 2. The estimation error is larger for Filter 3 on average (6.22 m compared to 5.08 m and 5.12 m for Filters 1 and 2, respectively), but the estimate of the covariance is more accurate since NEES is within the confidence bounds for almost the entire tracking period. Moreover, the estimation error could be reduced for Filter 3 by reducing the measurement noise covariance as discussed in the previous case study.

Figure 7.12 shows the correlation in the innovation sequences. The innovation se-

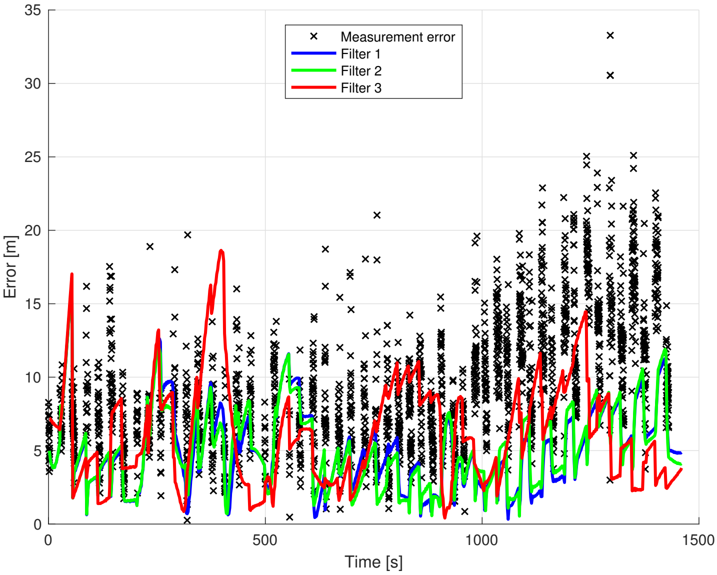


Figure 7.10: The estimation and measurement errors

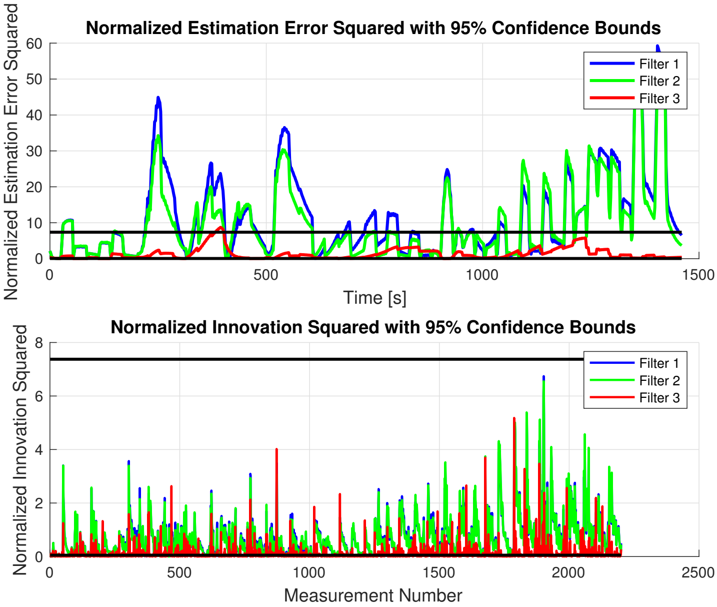


Figure 7.11: The normalized estimation error squared and the normalized innovation squared

quence is correlated for all tracking filters, but in less degree for the third filter. The influence of colored measurement noise is obvious for Filters 1 and 2, which is expected as they neglect it in the design. Considering correlation, NIS and NEES, only Filter 3 is close to fulfilling the criteria for consistency.

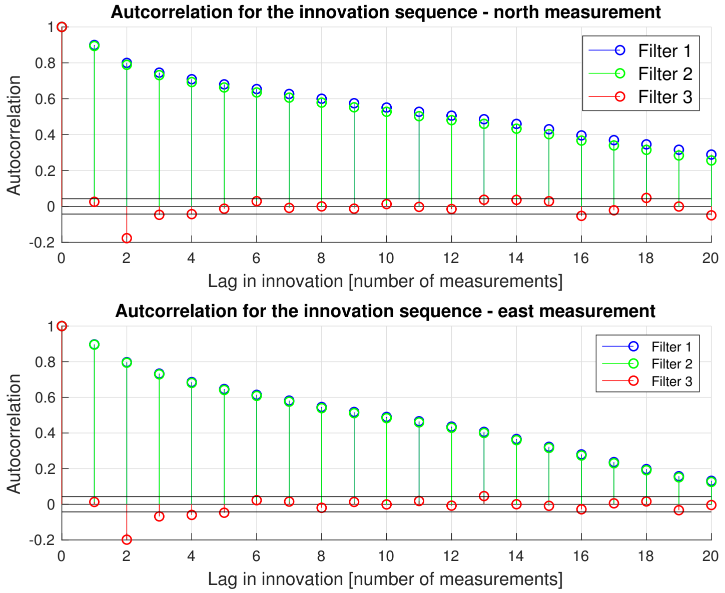


Figure 7.12: The autocorrelation for the innovation sequences

Both this case study and the previous have shown that colored noise is propagated into the tracking system through georeferencing, and that Filter 3 is better equipped to mitigate colored noise. The same difference in performance has not been illustrated between Filters 1 and 2. Therefore, a clear benefit with the Singer motion model has not been demonstrated. Nevertheless, it is not a major surprise because the target was not maneuvering extensively. In a situation with more maneuvers, the Singer model is expected to surpass the constant velocity motion model in accuracy.

7.5.3 Case study 3 - Tracking of high-speed vessel with the SKF

The objective in the third case study is to compare the EKF derived in Section 6.3 with the SKF presented in Section 7.4 when tracking of a high-speed target is of interest. Thus, the data from Section 7.5.1 and Section 6.4.2 are revisited. The same tuning is used as in Section 6.4.2 where the EKF was studied. The only difference is that the covariance of the measurement noise is increased to (30 pixels)^2 in both image dimensions (from 20^2) because the same measurement noise covariance is desired in this case study and the next. The measurement noise covariance could have been reduced significantly here, but results in poor performance in the next case study. A generic tuning is desired to prove that the system works in several scenarios.

Both the EKF and SKF use the CV motion model with σ_v from (6.2) equal to $(0.05\,\mathrm{m/s^2})^2$. The initial covariance in the north and east positions are $(10\,\mathrm{m})^2$ and the corresponding velocities have covariance $(10\,\mathrm{m/s})^2$. The initial covariance in velocity is not very far from the values proposed in [75] ($\frac{v^2_{\mathrm{max}}}{3}$) for initialization based on a single position measurement, and corresponds to a maximum possible target velocity of 17 m/s. Most vessels operate at much lower speeds, but the initial covariance should account for the maximum possible target velocity. The paths of the UAV and the target were displayed in Figure 7.2. Clutter and false alarms are not considered in this case study, and only one target is present.

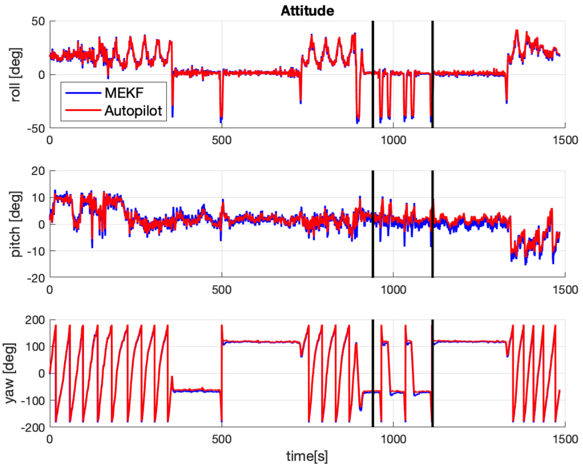


Figure 7.13: Estimated UAV attitude by the MEKF. The segment between the black vertical lines is the tracking period.

The navigation states of the UAV were estimated by the MEKF described in Section 7.3. The estimated attitude and velocity are shown in Figures 7.13 and 7.14, respectively. The estimates are compared with the estimates from the Pixhawk autopilot because a true reference is unavailable. The estimates from the MEKF are comparable to the estimates from the autopilot, and a significant difference is not observed. Tuning of the MEKF is not described because it is less relevant for the purpose of this case study.

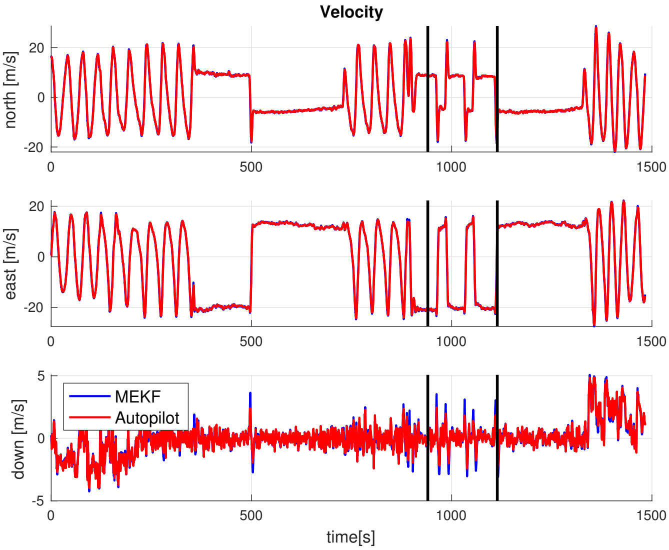


Figure 7.14: Estimated UAV velocity by the MEKF. The segment between the black vertical lines is the tracking period.

A more interesting factor is the magnitude of the estimated covariance in the navigation uncertainty since it is used in the tracking system. The covariance is nearly constant in the tracking period. The mean estimated standard deviations for the north, east and down positions are 8 cm, 8 cm and 7 cm, respectively. The mean estimated standard deviations for the roll, pitch and yaw angles are 1.35°, 1.30° and 2.65°, respectively. The uncertainty in the position is obviously less influential than the attitude [46]. The estimated standard deviation is small in position because position measurements were provided with RTK capability.

Figure 7.15 shows the attitude error when the autopilot is used as reference. Moreover, confidence bounds given by two times the estimated standard deviation are also displayed. Figure 7.16 shows the same error and confidence bounds for the estimated velocity. The roll and pitch errors are mainly within the confidence bounds. The yaw error is not equally small, particularly in the tracking period. Mainly straight-line paths were followed during the tracking period. Consequently, it is more difficult to estimate the heading in wind because of sideslip. The autopilot uses a magnetometer as heading reference in contrast to the MEKF which only has the two-antenna GNSS solution. The difference in estimated heading influences the estimation error in velocity since the attitude affects the velocity as shown in the kinematic equations in Section 7.3.4. Note that the autopilot is not necessarily more reliable than the MEKF so the difference can both be because of the MEKF and the autopilot.

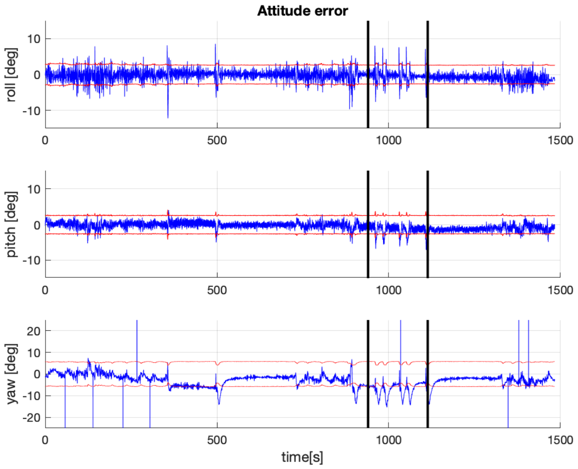


Figure 7.15: UAV attitude error for the MEKF when the autopilot is used as reference. The confidence bounds are given by two times the standard deviation. The segment between the black vertical lines is the tracking period.

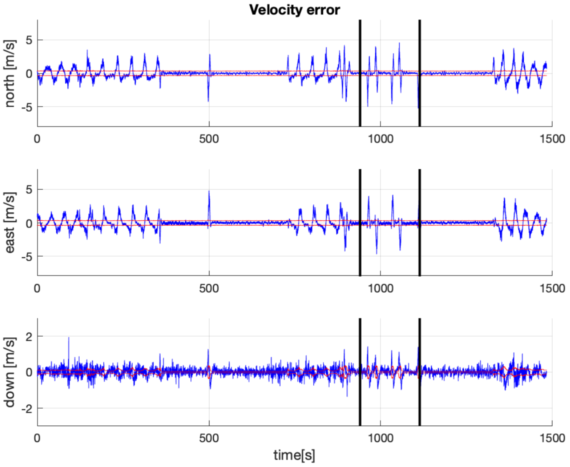


Figure 7.16: UAV velocity error for the MEKF when the autopilot is used as reference. The confidence bounds are given by two times the standard deviation. The segment between the black vertical lines is the tracking period.

The estimated total speed of the UAV is shown in Figure 7.17. The impact of wind can be observed because of the severe variations in total speed. They are mainly

Table 7.5: Average performance metrics in tracking of high-speed target in case study 3. ME = mean error. MAE = mean absolute error

Parameter	**EKF**	**SKF**
ME position estimates [north, east]	$[-4.1\,\mathrm{m},\ -1.2\,\mathrm{m}]$	$[-2.9\,\mathrm{m},\ -1.3\,\mathrm{m}]$
MAE estimated position	9.4 m	8.7 m
Mean NEES	7.90	0.40
Mean NIS	0.42	0.37

caused by headwind and tailwind, even though the total speed also decreases in turns. Therefore, it is likely that a significant crab angle was present during the flight and affects the accuracy of the estimated heading angle.

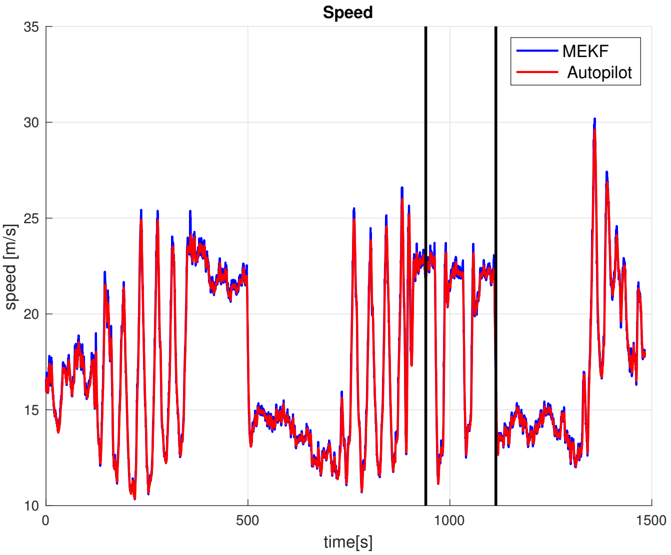


Figure 7.17: Estimated total speed of the UAV. The segment between the black vertical lines is the tracking period.

The estimation errors in target position are shown in Figure 7.18. The estimated target paths are shown in Figure 7.19. The main results of this case study are summarized in Table 7.5. The accuracy of the estimated position is comparable for the EKF and the SKF, but the SKF is more accurate overall. The mean estimation error is closer to zero and the average absolute estimation error is also smaller.

The estimated target speed and course are displayed in Figure 7.20. It is hard to observe a major difference, but the estimated speed and course from the SKF are more accurate initially and this explains why the estimated position of the EKF drifts more in the beginning of the period (the first segment without measurements).

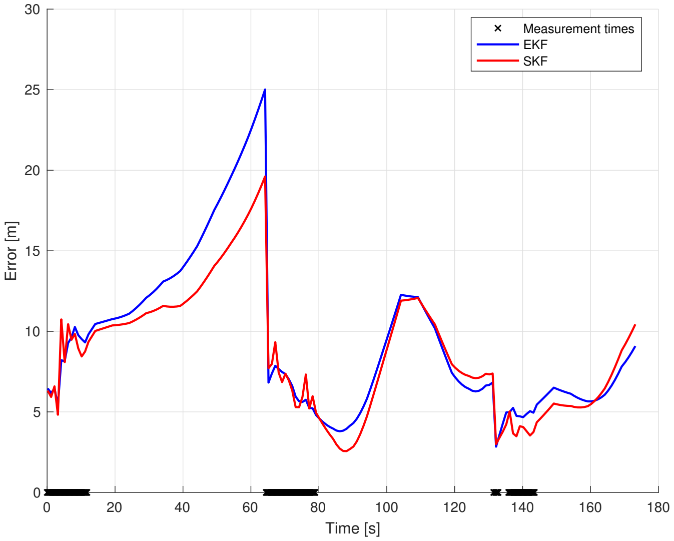


Figure 7.18: Estimation error in target position

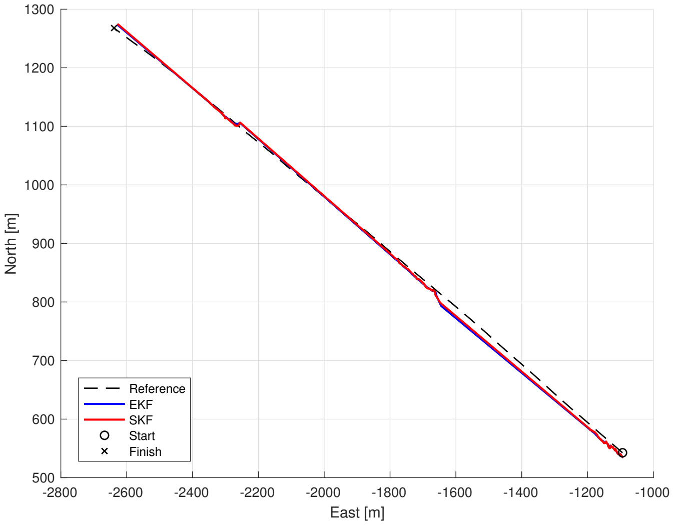


Figure 7.19: Estimated target path and reference

The largest difference is experienced in Figure 7.21 where the NIS and NEES (in position) are shown. The NEES is outside of the confidence bounds for the EKF in periods without measurements and the filter is too optimistic with a wrong covariance estimate. The SKF on the other hand has the opposite behavior. In fact, the SKF is actually too pessimistic. Therefore, the measurement noise covariance could have been reduced as mentioned in the previous case studies. This is also supported by the NIS, which is small. It is obviously possible to reduce the NEES for the EKF

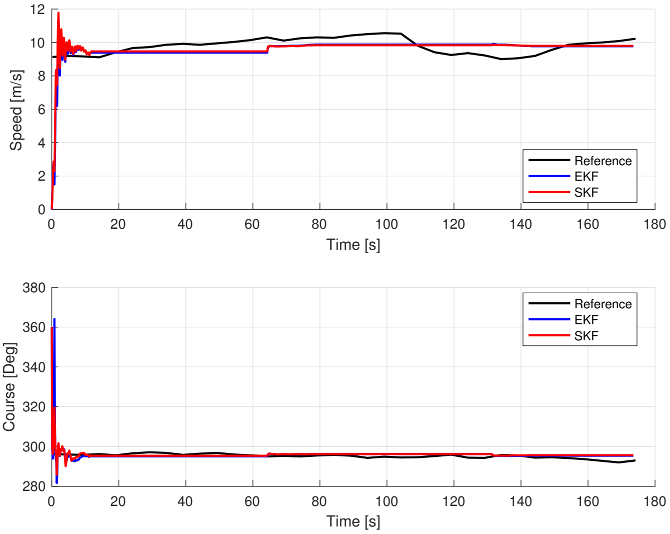


Figure 7.20: Estimated target speed and course together with reference for the extended Kalman filter (EKF) and the Schmidt Kalman filter (SKF)

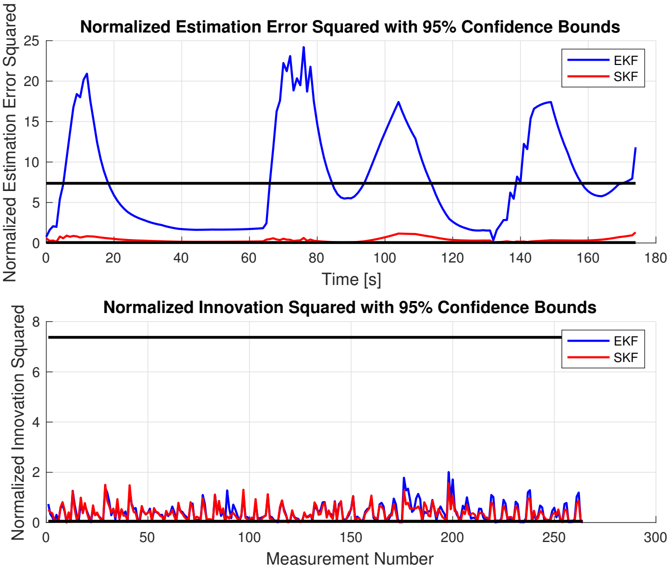


Figure 7.21: Normalized estimation error squared and normalized innovation squared for the extended Kalman filter (EKF) and Schmidt Kalman filter (SKF)

by increasing the measurement noise covariance. However, the measurement noise standard deviation must be increased to 60 pixels (from 30) to maintain consistency. This is much larger than the expected measurement noise from the object detection algorithm, and a solution that must be tailored to every new set of data.

Consequently, it is not robust in the same sense as the SKF even though such a solution does not necessarily affect the accuracy negatively. The results presented in this case study are further discussed at the end of the next case study.

7.5.4 Case study 4 - Tracking of slowly-moving vessel with the SKF

The final case study in this thesis has the same objective as the previous one, but the SKF is investigated for a slowly-moving boat. Data from two independent flight experiments are used. The first flight consists of target detections in 600 images (corresponds to 80 seconds with detections) spread over a period of 750 seconds. The detections are spread into 10 segments. The second flight consists of 2400 target detections (corresponds to 320 seconds) over a period of 1500 seconds. The detections are spread into segments in the second flight as well. Both the EKF and SKF are tuned in the exact same manner as in the previous case study. Remember that the previous case study concerned a high-speed target so the behavior of the targets are different even though the same tuning is used.

Flight 1

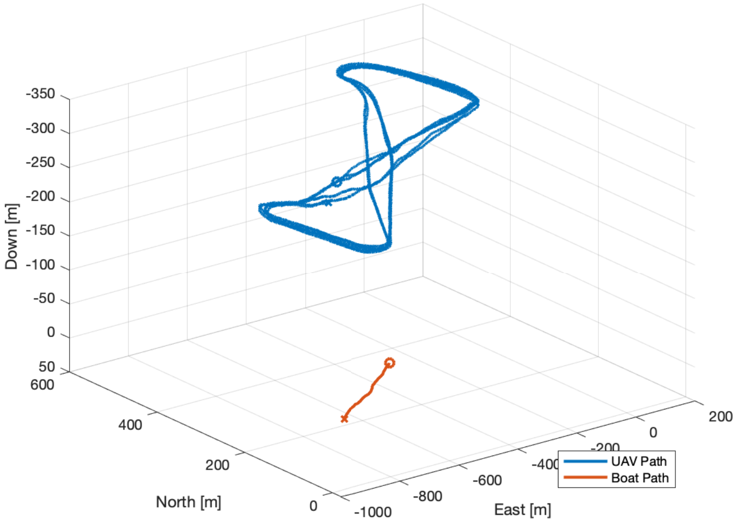


Figure 7.22: The paths of the UAV and target in flight 1

The tracking segment investigated here is a part of the data used in Section 6.4.2 for the slowly-moving target. The paths of the UAV and the vessel are displayed in Figure 7.22. The UAV operated at an altitude of 300 m and in a figure-eight pattern. Consequently, the attitude of the UAV varied more during this flight than in the previous case study where straight-line segments were followed during target observations.

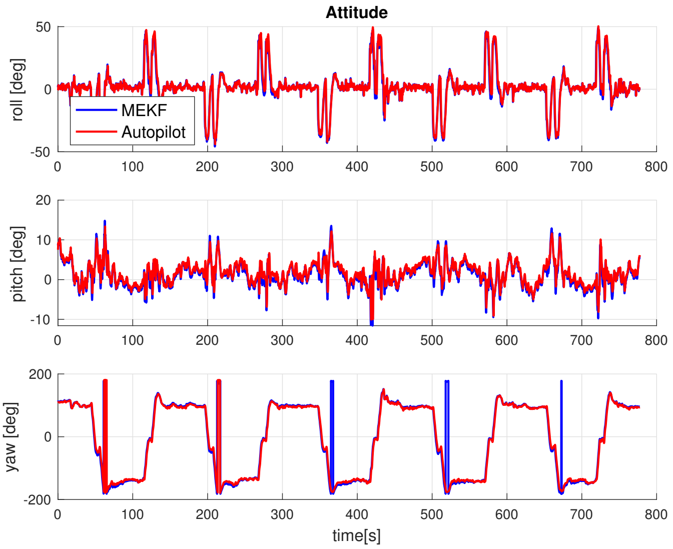


Figure 7.23: Estimated UAV attitude in flight 1

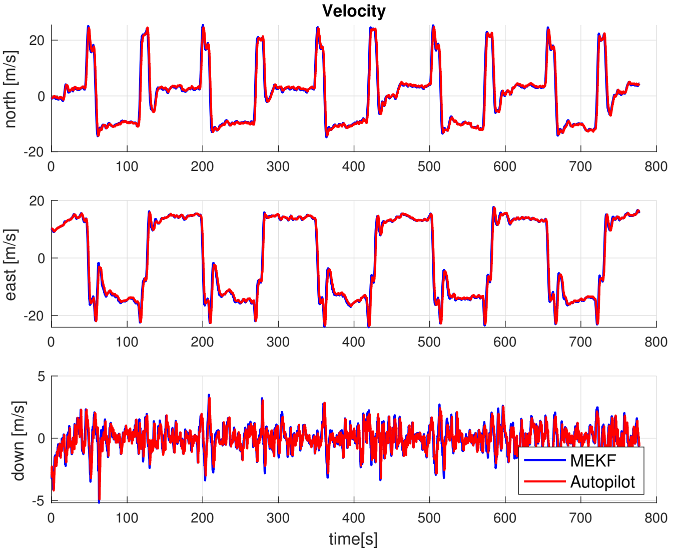


Figure 7.24: Estimated UAV velocity in flight 1

The UAV attitude and velocity were estimated by the MEKF and shown in Figures 7.23 and 7.24, respectively. The large visible difference in yaw is caused by jumps between $-\pi$ and π. The mean estimated standard deviations for the error states in the north, east and down positions are 43 cm, 43 cm and 68 cm in the tracking period, respectively. The mean estimated standard deviations for the roll, pitch and yaw angles are 1.22°, 1.24° and 2.63°, respectively. The standard deviation in position is larger here than in the previous case study because the IMU was weighted more in the navigation filter. This was caused by the flight pattern which was dominated by turns.

Figures 7.25 and 7.26 show the UAV attitude and velocity errors, respectively. The autopilot is used as reference. The roll error is larger in this flight, but the majority of the estimates are within two times the standard deviation. The largest errors occur during turns. This can be because of a small synchronization error between the autopilot and the MEKF since the roll angle changes quickly in turns. The yaw error is also non-zero in parts of the flight and the largest errors coincide with when the roll error is severe. Therefore, the attitude errors in roll, pitch and yaw are correlated. The velocity errors are mostly within two times the standard deviation. Velocity errors outside of the confidence bounds occur at times where the attitude error is most severe.

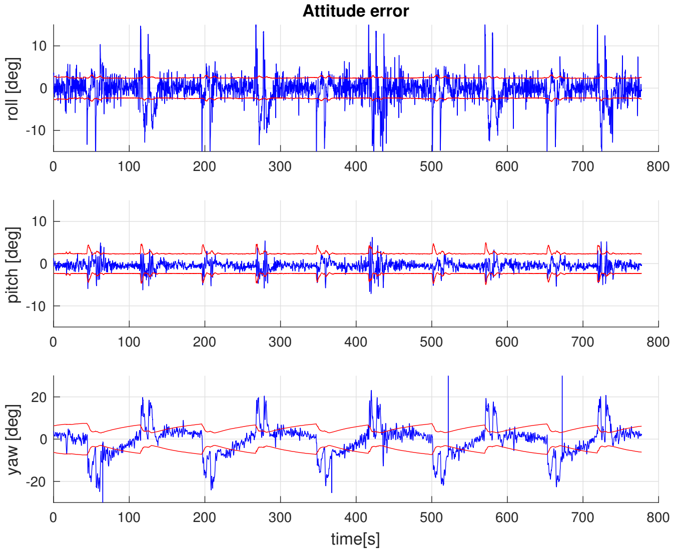


Figure 7.25: UAV attitude error for the MEKF in flight 1. The autopilot is used as reference and the confidence bounds are given by two times the standard deviation.

The estimated total speed is shown in Figure 7.27. The impact of wind can be observed in this flight as well. The total speed is about 15 m/s in headwind and about 24 m/s with tailwind. The total estimated speed is comparable for the MEKF and the autopilot, even though the velocity errors in NED are non-zero. This confirms that the main difference between the MEKF and the autopilot is related to the attitude.

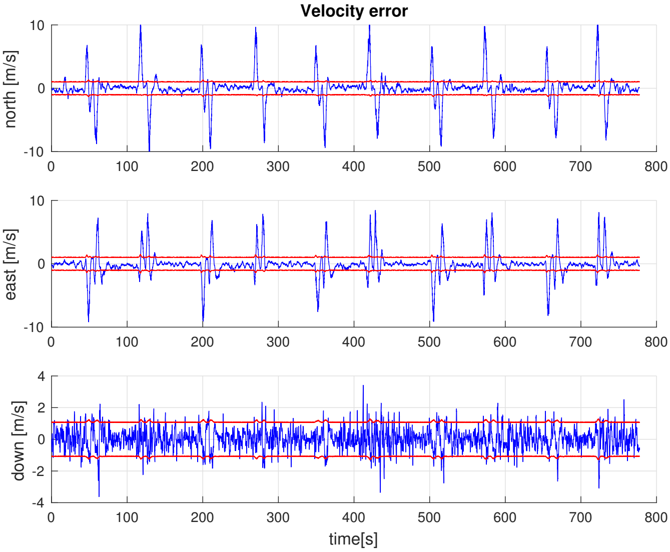


Figure 7.26: UAV velocity error for the MEKF in flight 1. The autopilot is used as reference and the confidence bounds are given by two times the standard deviation.

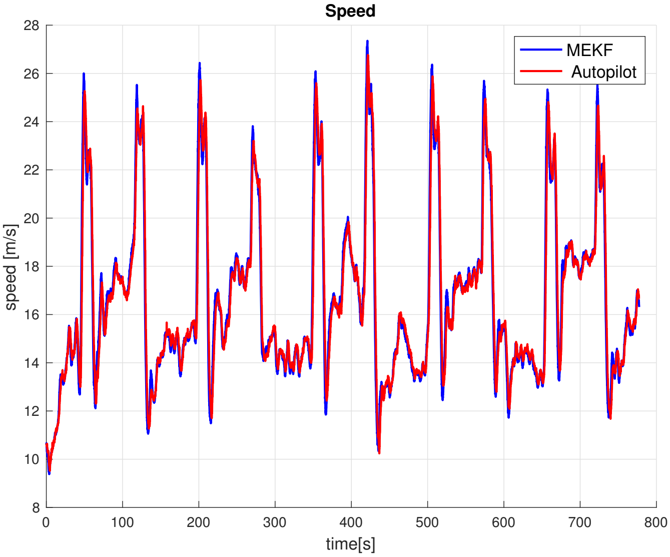


Figure 7.27: Estimated total speed of the UAV in flight 1

The estimation errors in target position are shown in Figure 7.28. The estimated target path is shown in Figure 7.29. The main results of this flight are summarized in Table 7.6. The SKF is more accurate than the EKF. The EKF drifts in periods without measurements while the SKF manages to predict the position quite accurately. This means that the velocity is estimated more accurately for the SKF.

This is also supported by the estimation errors in Table 7.6.

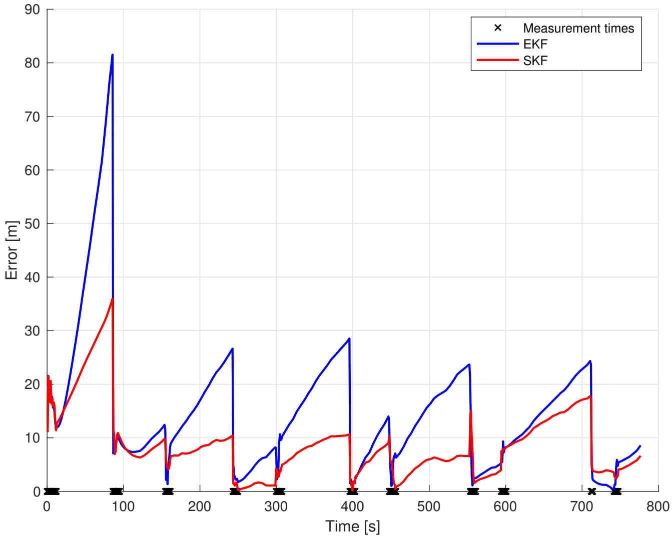


Figure 7.28: Estimation error in target position in flight 1

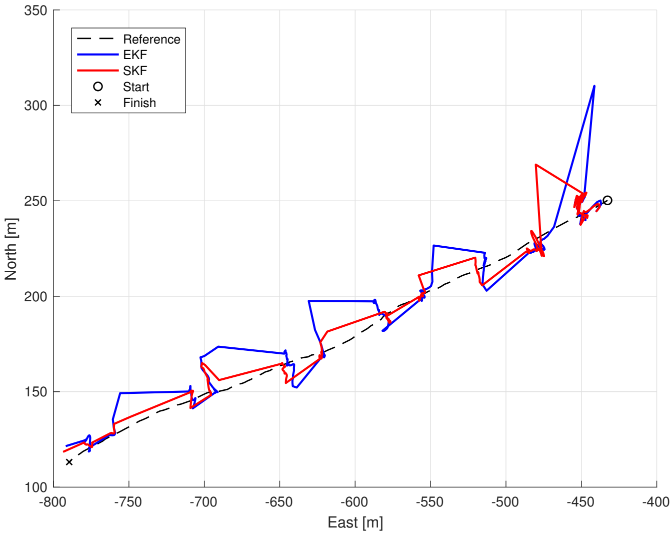


Figure 7.29: Estimated target path in flight 1

The estimated target speed and course are displayed in Figure 7.30. The EKF manages to estimate the total speed accurately, but the estimated course is wrong and it never converges in the same sense as for the SKF. This explains why the EKF drifts more in periods without measurements and this behavior was also observed

Table 7.6: Average performance metrics in case study 4 and flight 1. ME = mean error. MAE = mean absolute error

Parameter	**EKF**	**SKF**
ME position estimates [north, east]	[10.5 m, 1.76 m]	[4.44 m, 0.08 m]
MAE estimated position	15.0 m	8.79 m
Mean NEES	17.8	1.07
Mean NIS	1.84	1.84

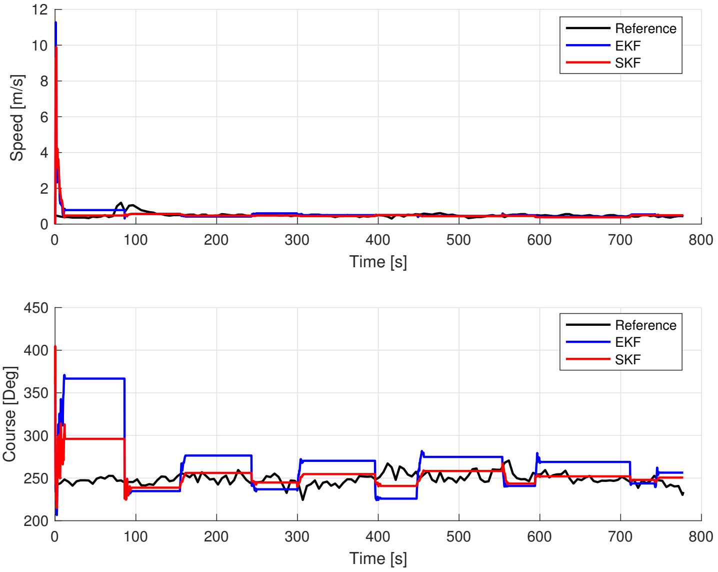


Figure 7.30: Estimated target speed and course together with reference for the extended Kalman filter (EKF) and the Schmidt Kalman filter (SKF) in flight 1

in the previous case study. Since both the EKF and SKF use the same motion model and the same set of measurements, one can conclude that the SKF is more accurate in this flight.

The NIS and NEES (in position) are displayed in Figure 7.31. The NIS is actually similar for both filters and within the confidence bounds for most measurements. However, a few measurements are outside of the confidence bounds, but that is not necessarily an issue for consistency. Nevertheless, it can be an issue for the validation region in measurement association if measurements are discarded. The largest NIS values occur when there is a long time since the previous measurement and that is the expected behavior.

The NEES is not similar. The EKF has NEES that is much larger than the upper confidence bound in every period solely based on prediction. This means that the filter is too optimistic and that the estimated covariance in position is much smaller than the corresponding estimation error. The SKF on the other hand is within the confidence bounds except for a short initial period. This supports the findings in the previous case study and shows that the SKF is able to maintain consistency when the regular design with the EKF struggles. Thus, the benefit of accounting for navigation uncertainty is demonstrated in this flight as well.

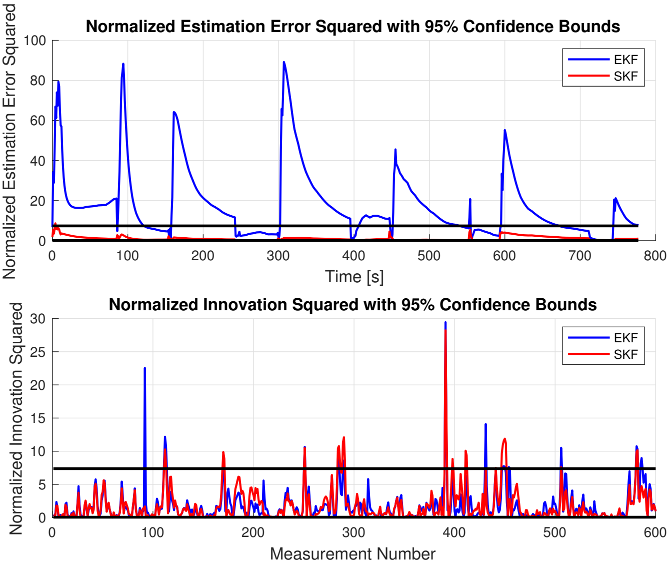


Figure 7.31: Normalized estimation error squared and normalized innovation squared for the extended Kalman filter (EKF) and Schmidt Kalman filter (SKF) in flight 1

Flight 2

The second flight uses a part of the data investigated in Section 5.6.2. The paths of the UAV and target are displayed in Figure 7.32. The UAV operated at an altitude of 200 m to 300 m in a figure-eight pattern. Thus, the UAV motion was similar to flight 1, but the altitude was lower in the first half of the tracking period. In addition, the target moved slower during this flight and only drifted in the water.

The UAV attitude and velocity were estimated by the MEKF and are shown in Figures 7.33 and 7.34, respectively. The visible difference in yaw is caused by the same jumps between $-\pi$ and π as in flight 1. The mean estimated standard deviations for the north, east and down positions are 87 cm, 87 cm and 1.42 m, respectively. The mean estimated standard deviations for the roll, pitch and yaw angles are 1.73°, 1.78° and 2.48°, respectively. Figures 7.35 and 7.36 show the attitude and velocity errors, respectively. The autopilot is used as reference. The roll and pitch errors are smaller in this flight. The yaw error is still quite large, but smaller than

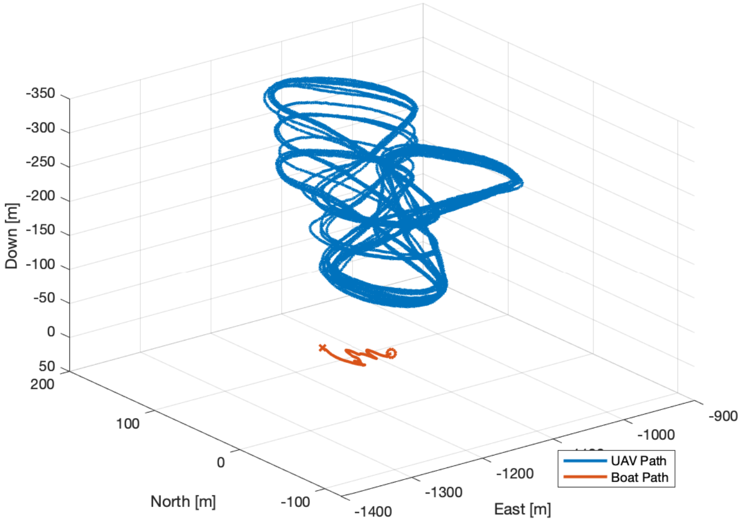


Figure 7.32: The paths of the UAV and target in flight 2

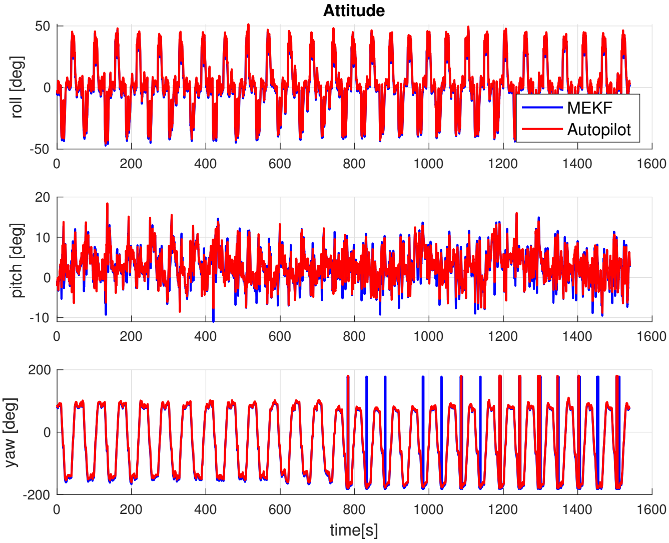


Figure 7.33: Estimated UAV attitude in flight 2

in flight 1. The velocity errors are also smaller in this flight and explained by the increased similarity for the attitude estimates. The velocity errors outside of the confidence bounds are not substantial and mainly occur because of turns.

The estimated total speed of the UAV is shown in Figure 7.37. The MEKF esti-

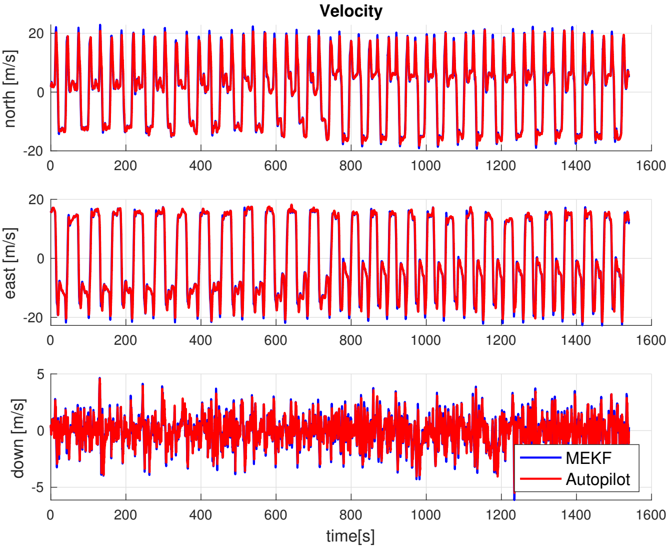


Figure 7.34: Estimated UAV velocity in flight 2

Table 7.7: Average performance metrics in case study 4 and flight 2. ME = mean error. MAE = mean absolute error

Parameter	**EKF**	**SKF**
ME position estimates [north, east]	[−3.32 m, 3.65 m]	[−4.83 m, 3.11 m]
MAE estimated position	8.80 m	8.04 m
Mean NEES	40.4	2.99
Mean NIS	1.75	1.09

mates a larger total speed than the autopilot. The wind effect was perhaps a bit smaller in this flight since the difference between the largest and smallest values is smaller. Reduced wind is a likely explanation for the reduced difference in the estimated attitude between the autopilot and the MEKF. Note that the attitude error indicates that a bias might be present in roll and yaw.

The estimation errors in target position are shown in Figure 7.38. The main results of this flight are summarized in Table 7.7. The performance, with respect to tracking accuracy, is comparable for the EKF and SKF. The EKF has the average estimation error closest to zero, but the SKF has the smallest average absolute estimation error. The estimation error in target position shows that the estimation error for the SKF oscillates less than for the EKF. This was also observed in the previous flight even though the drift was more significant there. The average absolute estimation error is smaller in this flight. That can both be because of the reduced altitude of the UAV and the vessel speed, which was smaller during this experiment.

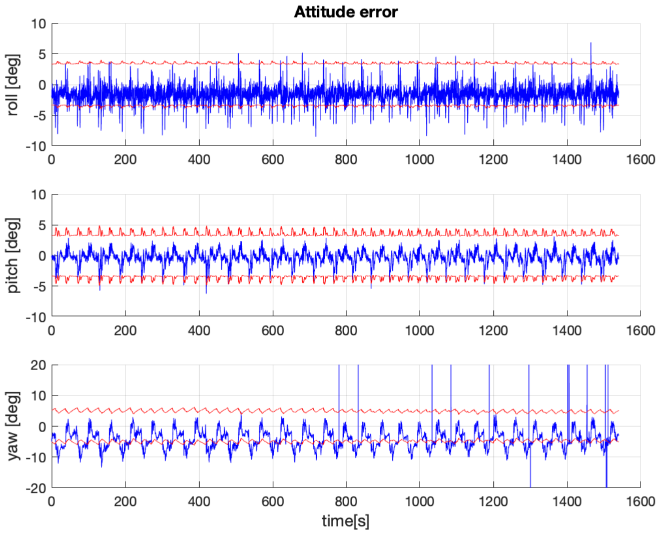


Figure 7.35: UAV attitude error for the MEKF in flight 2. The autopilot is used as reference and the confidence bounds are given by two times the standard deviation.

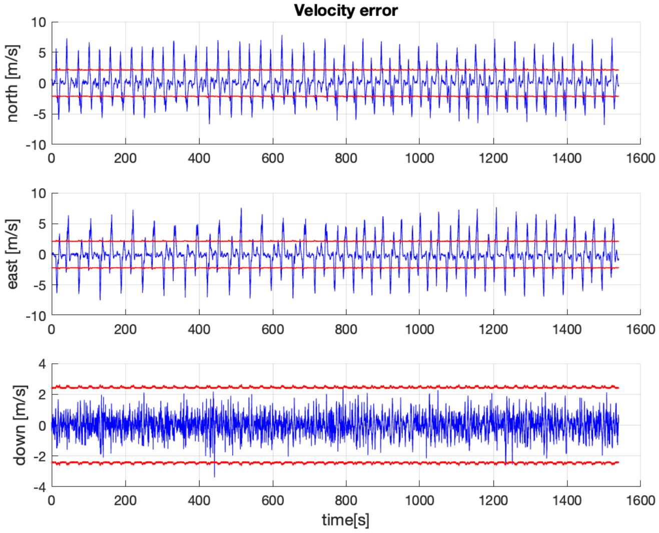


Figure 7.36: UAV velocity error for the MEKF in flight 2. The autopilot is used as reference and the confidence bounds are given by two times the standard deviation.

The estimated target paths of both the EKF and SKF are shown in Figure 7.39. The position of the EKF oscillates more and that was expected based on the oscillations

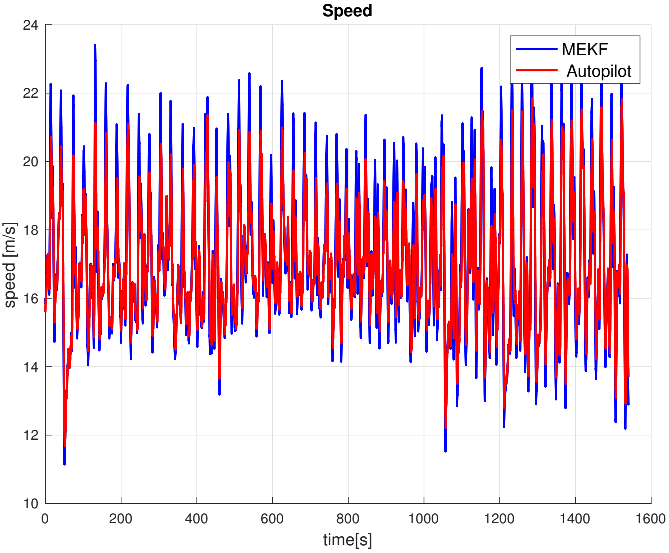


Figure 7.37: Estimated total speed of the UAV in flight 2

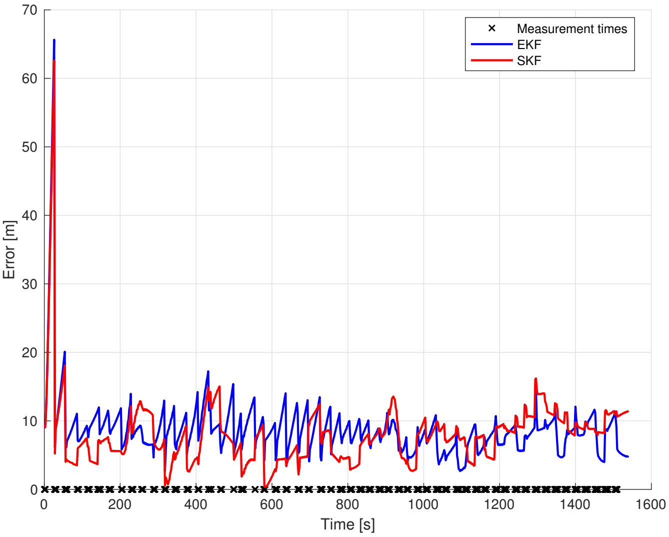


Figure 7.38: Estimation error in target position in flight 2

in the estimation error. The initial estimation error is due to the large initial covariance in the velocity and the poor accuracy in the first few measurements, which lead to poor velocity estimates initially. Note that the vessel only moved approximately 200 m during the tracking period.

Figure 7.40 shows the estimated target speed and course. The estimated speed is lower for the SKF and explains why the position estimates oscillate less. The

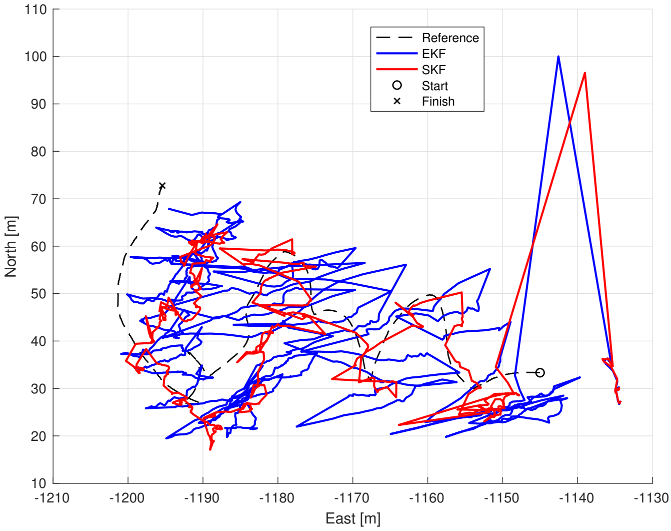


Figure 7.39: Estimated target path in flight 2

EKF never manages to find the correct speed, but closes in near the end. Drift in the position estimates is avoided because enough measurements are available. Both the EKF and the SKF struggle to estimate the correct course. The SKF converges halfway into the tracking period and is more accurate than the EKF. The EKF never manages to find the correct course. Note that both the speed and course references were gathered with a single-frequency GPS receiver. Moreover, the speed of the vessel was so low that the course might be inaccurate at times and not properly defined. The course reference could have been smoothed, but the raw data are shown here because it was used in the previous case studies. Nevertheless, these results show that the SKF estimates the speed and course more accurately, but that it does not affect the accuracy of the estimated position in a significant manner due to the total amount of measurements.

Figure 7.41 shows NIS and NEES (in position) for both tracking filters. The NEES is never within the confidence bounds for the EKF, which means that the estimated covariance is too small in this flight as well. Thus, this is a trend observed in several case studies and a weakness with the EKF. The SKF on the other hand, has NEES within the confidence bounds for most of the time and has a better performance with respect to consistency. This flight also shows a significant difference in NIS in contrast to the first flight. The NIS for the EKF is too large for several measurements and explained by a small innovation covariance. Thus, the covariance of the measurement noise should have been increased for the EKF. The SKF also has a few measurements outside of the confidence bounds. Nevertheless, this behavior is normally accepted since the majority of the measurements are within the bounds. Note that NIS and NEES, for the SKF, indicate that that the filter parameters fit well with the target behavior in this flight. This is because both of these measures are neither too low or too high with respect to the confidence bounds.

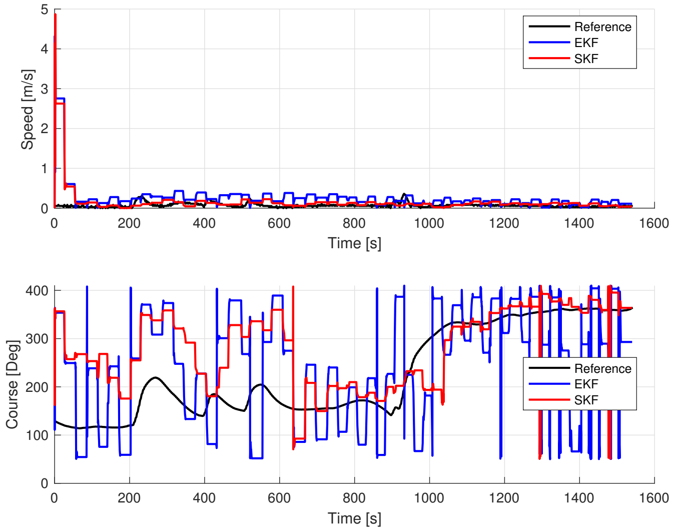


Figure 7.40: Estimated target speed and course together with reference for the extended Kalman filter (EKF) and the Schmidt Kalman filter (SKF) in flight 2

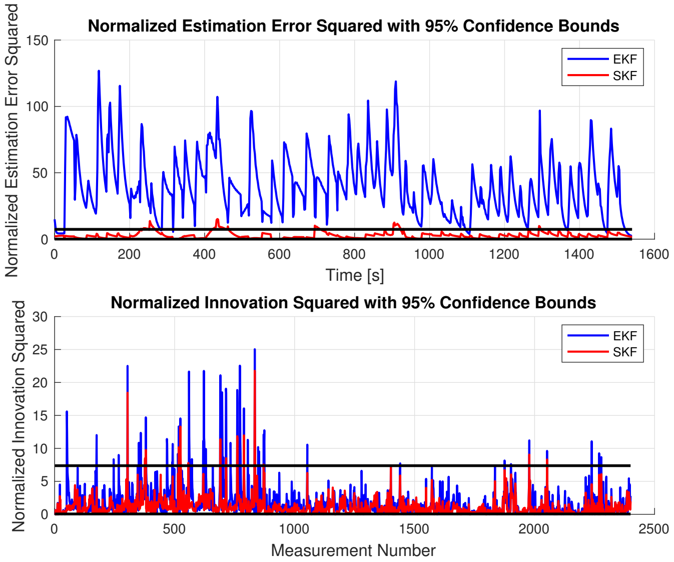


Figure 7.41: Normalized estimation error squared and normalized innovation squared for the extended Kalman filter (EKF) and Schmidt Kalman filter (SKF) in flight 2

A comprehensive amount of data have been used to compare the EKF with the SKF in case studies 3 and 4. A few conclusions can be made based on the results:

- The SKF is more accurate and is especially better when it comes to estimating the speed and course.
- The position estimates with the EKF are accurate, but are weak for long periods solely based on prediction.
- The major difference between the SKF and EKF is observed for the NEES where the EKF is overconfident. The estimated covariance is much smaller than the corresponding estimation error and this is an issue in data association. It is possible to increase the measurement noise covariance, but it is hard to generalize such a solution to fit new data. The SKF on the other hand, works for a fixed set of filter parameters as shown in case studies 3 and 4. Moreover, it is more robust and reliable with respect to NEES.
- Even though the SKF is superior in some means, the necessary covariance of the measurement noise is larger than the expected uncertainty in the detection algorithm. Consequently, the navigation uncertainty is not mitigated perfectly. This is most probably because the constant bias assumption is naive at times. The navigation error is not constant throughout a long flight. Moreover, the SKF is not necessarily able to remove the correlation in the innovations in the same sense as the method in Section 7.2.
- The same tuning was used in both case studies and different results were achieved. The tuning matched the data in the final case study better and it highlights the challenge one have when tracking unknown targets. The behavior is unknown beforehand so it is impossible to find parameters that are ideal in many different scenarios. Nevertheless, it is possible to identify values that are acceptable in several scenarios as shown in these case studies.
- The covariance of the navigation estimates play a major rule on the result. This is particularly observable in the NEES where the major difference between the EKF and SKF is casued by the navigation uncertainty. Moreover, the attitude is again the critical factor and a standard deviation of about 1° to 2° in roll, pitch and yaw seems to have the desired effect. The influence of the position uncertainty is negligible unless the standard deviation is large (5 meters or greater). Therefore, it is confirmed that the attitude is the critical factor for tracking of floating objects with optical sensors.

Chapter 8

Conclusions & Future Possibilities

This thesis has investigated tracking of floating objects using a monocular optical sensor. Seven chapters have been dedicated to describe and investigate this topic. The first chapter gave an introduction to the subject and tried to motivate and explain why the work in this thesis is useful. The subsequent two chapter gave a fundamental introduction to sensors, UAVs and target tracking. Chapter 4-7 considered everything from detection of floating objects to georeferencing and ended up with specific methods for tracking of floating objects. Moreover, numerous case studies have been designed and carried out to investigate the matter at hand. This thesis has, with one exception, been based on experimental data. The main focus has been on improving the accuracy of existing solutions while also highlighting issues related to optimality and consistency. The results of this thesis were described in 13 major case studies used to investigate different topics.

This chapter concludes this research and ties the results of the different case studies together. It has been divided into the following two parts:

- Section 8.1 concludes this thesis. The concluding remarks are based on the experimental results and summarize the main trends. Strengths and limitations with this thesis are also discussed thoroughly.

- Section 8.2 discusses future possibilities and issues that should be investigated further. This part is based on the experience gathered throughout this research and gives recommendations for next steps that follow naturally from this work.

8.1 Conclusions

This thesis has looked into several aspects regarding tracking of floating objects using fixed-wing UAVs. The primary sensor used for this purpose has been a thermal camera operating in the LWIR part of the infrared spectrum. A thermal camera was chosen because of the distinct thermal signature of typical objects at sea. Nevertheless, the methods presented in this thesis are equally applicable for other optical sensors, such as a visual spectrum camera. Only the detection algorithm must be changed in such a scenario. The thesis has focused on demonstrating different methods through analysis of experimental data, and investigating the accuracy in real applications. Moreover, real-time feasibility has been a requirement throughout this work, even though most of the results have been post-processed after the experiments.

8.1.1 Overview

Chapter 1 introduced the scope of this thesis from a remote sensing perspective. It was emphasized why UAVs equipped with optical sensors are a significant capacity in missions where mapping and surveillance are the main objectives. Research challenges were addressed and the main objectives and contributions of this thesis were specified.

Chapter 2 went on to describe UAVs and optical sensors in more detail. Common terms and expressions were defined and fundamental information regarding the electromagnetic spectrum and optical sensors were presented. A tailor-made payload for detection and tracking of floating objects was developed as a part of this research and the payload was described in this chapter. Sensor calibration and time synchronization were also addressed, and are factors that are often neglected in research. These topics are perhaps the most important considerations when dealing with experimental data.

Chapter 3 gave a basic introduction to target tracking. Related literature was described and recent trends within this field were highlighted. Moreover, the different parts of a general tracking system were introduced and everything from Bayesian tracking of a single target to data association and multiple target tracking were described. Track maintenance and consistency criteria were also addressed. This chapter ended the first part of this thesis, which gave the necessary background for the methods discussed in the rest of this research.

The second part of this thesis started with Chapter 4 where object detection in optical images was discussed. Related literature and existing work were presented and connected to detection of floating objects. The chapter went on to discuss feature detectors and how they can be used to find objects at sea. Feature detectors cannot be used as a standalone solution. However, features at the sea surface are rare in thermal imagery, and can for instance be used to guide another detector into parts of an image that should be investigated more thoroughly. The chapter also described a specific method for detection of floating objects in thermal imagery. The method utilizes the typical difference in emissivity for floating objects compared to the sea surface.

The final part of Chapter 4 contained two independent case studies. The first was used to illustrate how floating objects are detected. The second was used to illustrate that a similar procedure works in a different scenario and environment with a visual spectrum camera. The results showed that the floating object detection algorithm has a detection probability close to 1 in detection of marine vessels. Moreover, clutter and false positives were rare and not expected in general. The lack of false positives and clutter is a huge advantage in this scenario because these factors complicate tracking systems, and is a fundamental challenge in radar tracking. Because of these experiences, state estimators that can handle large amount of clutter have not been studied extensively in this thesis and other aspects have been prioritized. This is different from most existing literature that focus mostly on tracking systems that can handle a large amount of false positives.

Chapter 5 concerned georeferencing of pixels in optical images, which was used to map targets detected in a single image to an Earth-fixed coordinate frame. Related literature was described and a specific algorithm for georeferencing of images without ground reference markers was derived. Moreover, the need for compensation of mounting misalignments was stressed and proved with a sensitivity analysis. A specific method for detecting camera mounting misalignments was derived. The relationship between target velocity and optical flow was also derived in this chapter under a set of assumptions. This relationship is a useful theoretical result, but is sensitive to timing errors and uncertainty in the navigation states of the UAV. A navigation system based on a nonlinear observer was introduced as an alternative to using the autopilot navigation data.

The final part of the fifth chapter presented three independent case studies. The firs two investigated georeferencing of a static object and of a moving marine vessel. Both of these case studies showed that the mean position was within two meters from the true position, and the performance was comparable to results achieved at much lower altitudes previously. This is a significant step forward because the error usually is proportional with the altitude of the UAV. Moreover, the necessity of time synchronization and calibration of misalignment errors were demonstrated. The final case study looked into target velocity reconstruction using optical flow. The reconstructed velocity was noisy, but the mean velocity was reasonable, especially because the autopilot navigation data were used without time synchronization or misalignment calibration.

Chapter 6 investigated tracking of floating objects without considering navigation uncertainty. Both a linear and nonlinear solution were presented with everything from modeling, to calculation of the validation region and tuning of filter parameters. This chapter was based on the previous three chapters and used methods from Chapter 3 to create the tracking systems. In addition, the floating object detection method and the georeferencing algorithm were utilized in the tracking system. A comprehensive amount of results were presented in four independent case studies based on experimental data from different flight experiments. Data association was investigated and the linear and nonlinear architectures were compared experimentally. A case study considering multiple targets was also presented and showed why it can be beneficial to use object properties such as size, shape and emissivity in data association. The accuracy of the results was convincing compared to existing results in the literature and supported the accuracy of the georeferencing results in Chapter 5. However, a few issues related to colored noise, consistency and optimality were experienced and described.

Chapter 7 looked into these issues which mainly were related to navigation uncertainty. Therefore, target tracking in the presence of navigation uncertainty was the main topic of this chapter. Two different strategies were presented. The first strategy was used to account for colored noise in the motion and measurement models. The second strategy incorporated the uncertainty of the navigation estimates directly in the tracking system by using the Schmidt-Kalman filter. This architecture requires access to the covariance of the navigation estimates. Consequently, an error-state Kalman filter (MEKF) was introduced and used to provide navigation estimates with a corresponding covariance. The chapter ended with four independent case studies that compared the methods presented in this chapter with the methods from Chapter 6. Some of the main issues related to colored noise and consistency were mitigated with these methods. Moreover, the SKF estimated the target states more accurately than the EKF.

8.1.2 Challenges and lessons learned

Tracking of floating objects using UAVs with a monocular optical sensor has been studied previously, but not very extensively. This thesis has investigated this topic further and tried to develop clever ways to face the most common challenges. A fundamental requirement stated in the beginning of the thesis was related to real-time feasibility. This has served as guideline when considering whether a method is appropriate or not.

This thesis has utilized a detection algorithm for floating objects that can be executed on the on-board computer. It has proven to be robust and reliable in detection of floating objects. Nevertheless, it is limited to work in situations where an object is present in front of a homogeneous background. It is not working equally well over land and must be tuned differently in such a situation. Object detection in visual spectrum imagery is dominated by deep learning, but detection in thermal images is not developed that far currently. Running convolutional neural networks on small embedded computers is resource-demanding in general even though some examples exist. Moreover, a substantial amount of training data must be gathered and that is difficult to acquire. Thermal images only consist of an intensity channel and do not possess color information. It is not given that deep learning is the way to go in floating object detection, but it is something that should be studied in the future.

The main weakness in the detection algorithm is the parameters that must be adjusted manually. Especially the minimum required area could potentially prevent the detector from finding small objects such as people in the water. This obviously also depends on the altitude of the UAV, which directly affects the size of objects mapped to the image frame. Reducing the required size of an object could result in some noise being interpreted as an object and increase the amount of false positives. However, this is not critical if the tracking system is adjusted to handle clutter. Because of the emissivity differences between the sea surface and objects, it is also possible to detect floating objects that do not have a temperature difference compared to the sea surface. However, the algorithm is mainly tested for marine vessels so it should be developed further to handle other types of targets.

How to improve the accuracy of georeferencing has been a fundamental research question during this work. The main bottleneck in georeferencing of optical images is not the accuracy of the detection algorithm. It is related to knowing the sensor pose accurately at the time of image capture, and involves both calibration and time synchronization. Even though these challenges are closely related to engineering, they raise some interesting research questions. Calibration should be reliable and also validated somehow. Moreover, requirements related to necessary accuracy in mounting and time synchronization are not clearly defined in the literature. Some argue that the only reliable way to validate a georeferencing system is to conduct flight experiments. This is quite demanding because it requires planning and resources. This thesis has mainly evaluated the georeferencing platform experimentally, but the expected accuracy was also investigated with a motion capture system.

Tracking of objects on the sea surface has been the most prominent objective in this work. The Kalman filter and the extended Kalman filter have been used as a base. The lack of literature for this particular application led to the use of simple state estimators initially. Thus, a weakness is that more advanced alternatives have not been investigated experimentally. The reliability of the detection algorithm has made simple estimators applicable. However, this also limits the usefulness in other scenarios such as tracking of ground objects over land because more clutter must be expected. Both the linear and nonlinear tracking architectures resulted in challenges related to tuning and optimality. This is the fundamental challenge investigated in this thesis and some of the issues have been mitigated in Chapter 7. Moreover, the lack of clutter has made initialization of the tracking filters simple, especially when georeferencing has been utilized. This has worked well for tracking of marine vessels, but is obviously more challenging in other situations. Nevertheless, many of the issues that have been revealed in this work are also relevant for tracking of ground objects because they are related to how images are captured and processed. Consequently, these issues are relevant if other tracking solutions are used instead.

The thesis has investigated tracking of a single target extensively with a large amount of experimental results. A multi-target scenario was also studied in Section 6.4.4, but it is challenging to design and collect data with several targets. Thus, more work should be conducted on data association and multi-target tracking in the future. A firm data association method was utilized, but the robustness of this method has not been investigated as thoroughly as desired. In radar tracking, it is more common to use soft data association and allow multiple measurements to affect a single target. Comparing firm and soft data association in tracking of floating objects is a topic that should be studied further. Firm association could lead to track swaps more quickly, but is at the same time more suitable for optical sensors in contrast to radars. Image classification and object properties can be used to aid the data association procedure. Consequently, firm data association is more trustworthy in this scenario than in radar tracking.

8.2 Future Work

Mapping and surveillance of the sea surface are research topics that is of further interest in the near future. Autonomous ships need real-time information about the proximity of the ship to make decisions. UAVs can be of assistance for larger ships that need longer time to adjust their path and course. Future work include development of better and more accurate methods for all parts of the tracking system, including detection, georeferencing, data association and the tracking filter itself.

Even though the detection algorithm has performed satisfactorily, it is necessary to study other alternatives in the future. Object detection based on convolutional neural networks is becoming feasible on embedded computers and should be studied for thermal images. Neural networks are not necessarily going to be equally effective in thermal imagery as for visual spectrum images, but is an open research question. Moreover, the ability to detect other objects on the sea surface such as marine mammals, icebergs and fishing equipment need to be addressed. Therefore, evaluating the existing method on new data and compare it with methods based on deep learning would increase the scope for when the detector is applicable.

The main pitfalls in georeferencing have been studied extensively in this thesis from a system perspective. Calibration and time synchronization have been addressed, and are always going to be important. The influence of the UAV path has not been studied in details and is also a factor that affects the accuracy of georeferencing. The georeferencing error is smaller in level flight than during maneuvers with extensive roll or pitch motions. This is because the ray from the camera to the ground increases with roll and pitch motions and makes navigation errors more critical as shown in Tables 5.1-5.3. Consequently, path planning is of interest as it is desired to observe objects in level flight. Note that the observability of the navigation system typically increases with dynamical behavior. Therefore, increased observability during roll and pitch motions might counteract the expected growth in the georeferencing error as long as a constant bias is avoided (e.g. mounting misalignments).

Path-planning is obviously also of interest to ensure that existing targets are revisited within an appropriate amount of time. Since the georeferencing error depends on the pose of the UAV, it is interesting to see if it is possible to tune the filter parameters in tracking systems in a way that depends on the motion of the UAV. Measurement noise that is proportional to the altitude of the UAV has been mentioned, but could be extended further to depend on the magnitude of the roll and pitch angles. Increased georeferencing error is also expected closer to the image boundaries so the measurement noise could also depend on the object position in the image.

Colored noise and consistency issues have been a recurring theme in the experimental work. The Schmidt-Kalman filter worked particularly well and is a natural choice because navigation uncertainty is incorporated directly in the tracking filter. Colored noise originates from the non-Gaussian nature of the navigation uncertainty. Tracking filters based on sampling methods that handle other probability distributions could be of interest. For example the unscented Kalman filter is computationally feasible and is perhaps equipped to handle tracking from UAVs in a good manner. That is the natural next step for the filtering part of the tracking system. Another interesting possibility is to incorporate target visibility and existence in the tracking filter through methods such as JIPDA [82]. Modeling target visibility is particularly interesting for tracking with UAVs because the motion of the camera causes the target to move in and out of the field of view.

Data association and multi target tracking must be studied further. Using image features in measurement association is a promising possibility. Object properties could even be estimated in the tracking filter so that the properties are incorporated directly when designing the validation region. Another possibility is to add association based on image features as a safety layer after the classical association method, and require that they agree before a firm association is conducted. Soft data association could potentially also be studied if experimental results show issues in other scenarios than the ones considered in this thesis. This can for example be achieved through tracking filters such as the joint probabilistic data association filter [4].

To further generalize the tracking system, multiple model approaches could be considered. This does not necessarily include finding other motion models than the constant velocity model. A viable alternative is to use several filters based on the constant velocity model where the magnitude of the process noise covariance varies. This could further improve the accuracy of the estimation error during phases with prediction and the robustness when designing the validation region. Large portions of the tracking period are often based on prediction as shown in the results, even if the path is optimized to keep the targets within the field of view.

Effective tuning of the process and measurement noise is also of further interest. This thesis has shown results with different tuning strategies, but could be studied further to find other methods. Empirical tests for identifying suitable values for the process noise covariance could be developed. Moreover, tuning of the measurement noise covariance is also of further interest, even though much experience have been acquired in this thesis.

All of the experimental results have shown that consecutive measurements in the tracking filters have errors that are correlated. Moreover, the camera has a frame rate of 7.5 Hz, which means that many measurements are available whenever an object is within the field of view of the camera. The detection algorithm usually manages to detect objects in a reliable manner. Consequently, many measurements are available every second when an object is in the field of view. This pushes the estimates in a specific direction because the measurement errors are typically correlated. Therefore, an interesting possibility is to increase the time between measurements in the tracking filter by only using every other detection (or even fewer). This would likely reduce the problem with correlated measurement noise and in best case improve both consistency and accuracy. The downside is increased time for the estimates to converge. Moreover, if the noise is modeled correctly, a large amount of noise and several measurements should not be problematic.

Appendices

Appendix A

Technical Specifications for UAV Payload

This appendix gives the technical specifications for the sensors in the thermal camera payload described in Section 2.3. Each sensor that is relevant for this thesis is described. The payload also consists of wiring, 3D-printed mounting brackets for every sensor and voltage converters. These parts of the payload are not described, but also required a significant amount of time to design, produce and assemble. The specifications of the synchronization board are described in [1] and not given here.

A.1 Thermal Camera

The core of the thermal camera payload is obviously the thermal camera. The payload was designed so that a FLIR Tau2 thermal camera could be mounted. The UAV lab at NTNU has several cameras with different focal lengths so the one used in most of the experiments is described here. Additional information about the camera can be found in [28]. The camera is displayed in Figure A.1.

Figure A.1: FLIR tau 2 thermal camera. ©FLIR

Table A.1: Technical specification for the FLIR 2 thermal camera

Name	FLIR Tau 2 640
Thermal imager	Uncooled VOx microbolometer
Focal length	19 mm (9mm and 13mm also available)
Digital resolution	640 x 512
Pixel depth	14 bit
Pixel pitch	17 µm
Frame rate (export regulated)	7.50 Hz
Sensitivity range	7.5 µm to 13.5 µm (LWIR)
Sensitivity	$<$ 50 mK
Scene range	−25 °C to 100 °C
Size without lens	44.5 mm x 44.5 mm x 30 mm
Weight	72 g
Input voltage	4 V to 6 V
Power dissipation in steady state	$<$ 1.2 W
Humidity	5 % to 95 % non-condensing
Operating temperature range	−40 °C to 80 °C
Operational altitude	$<$ 12 000 m
Shock	200g shock pulse with 11 ms sawtooth

A.2 Thermalgrabber

The Thermalgrabber [111] is used to get full access to the 14 bit raw digital data of the Tau2 thermal camera core. It is connected directly to the camera core and displayed in Figure A.2. It has low latency streaming of raw data that are stored by the on-board computer.

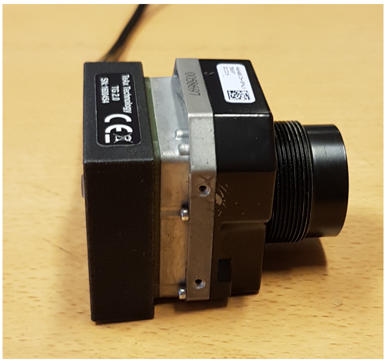


Figure A.2: Teax technology thermalgrabber mounted on the back of the thermal camera

Table A.2: Technical specification for Thermalgrabber

Name	ThermalCapture Grabber USB
Manufacturer	TeAx technology
Size	46 mm x 42 mm x 15 mm
Weight	25 g
Input voltage	5 V ±0.2 V DC
Current consumption	Typically 150 mA from USB
Operating temperature range	−10 °C to 45 °C

A.3 On-board Computer

The on-board payload computer is responsible for storing data and control the sensors in the payload. The on-board computer is manufactured by Hardkernel and called Odroid-XU4 [39]. It is displayed in Figure A.3.

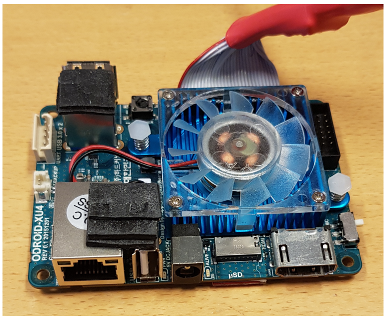


Figure A.3: Odroid-XU4 on-board computer

Table A.3: Technical specification for Odroid-XU4

Name	Odroid-XU4
Manufacturer	Hardkernel
CPU	2.0 GHz quad core
Processor type	ARM Cortex A15
RAM	2Gb
Size	83 mm x 58 mm x 20 mm
Weight	38 g
Input voltage	5 V ±0.2 V DC
Current need	Typically 4 A recommended to drive peripherals
Operating temperature range	−20 °C to 70 °C
Storage	Micro SD or eMMc

A.4 Retractable Gimbal

The gimbal is used to protect the camera during landing and give the camera a specific orientation. It is a R-BTC-88 retractable gimbal manufactured by microuav [95]. It is displayed in Figure A.4.

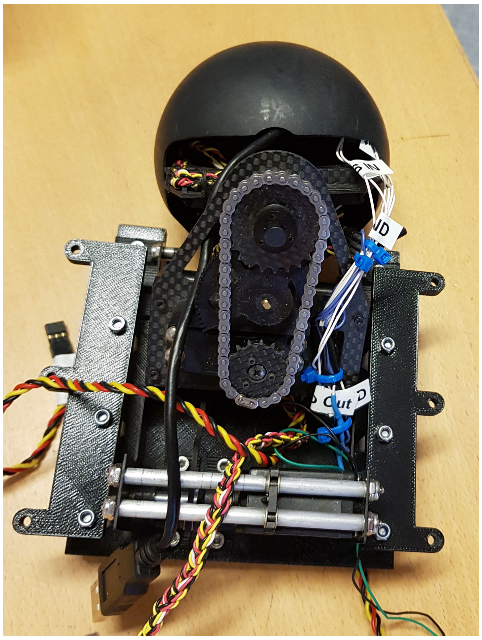


Figure A.4: R-BTC-88 retractable pan/tilt gimbal mounted in 3D-printed rack

Table A.4: Technical specification for the gimbal

Name	R-BTC-88
Manufacturer	Microuav
Size	12.3 cm x 8.9 cm x 9.1 cm
Weight	275 g
Azimuth range	−180° to 180°
Elevation range	80°
Azimuth and elevation speed	200 °/s
Voltage	5 V
Control	PWM
Operating temperature range	−30 °C to 80 °C

A.5 Adis 16490 IMU

The Adis 16490 inertial measurement unit [2] is used to measure specific force and angular rate. These measurements are used when estimating the navigation states of the UAV with the nonlinear observer in Section 5.5 and the MEKF in Section 7.3. The IMU is displayed in Figure A.5.

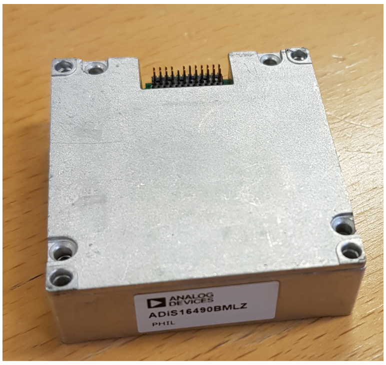


Figure A.5: Adis 16490 inertial measurement unit

Table A.5: Technical specification for the Adis 16490 IMU

Name	Analog Devices Adis 16490
Gyroscope	Triaxial digital with ±100 °/s dynamic range
Gyroscope in run bias stability	1.8 °/h
Angular random walk	$0.09\,°/\sqrt{\text{h}}$
Accelerometer	Triaxial digital with ±8 g dynamic range
Accelerometer in run bias stability	3.6 µg
Velocity random walk	$0.008\,\text{m/s}/\sqrt{\text{h}}$
Delta measurements	Triaxial delta angle and delta velocity output available
Axis to axis misalignment error	±0.05°
Axis to package misalignment error	±0.25°
Size	47 mm x 44 mm x 14 mm
Weight	42 g
Input voltage	3 V to 3.6 V
Operating temperature range	−40 °C to 105 °C
Shock	2000g shock survivability

A.6 STIM 300 IMU

The STIM 300 inertial measurement unit [101] is used to measure specific force and angular rate, and is an alternative to ADIS16490. It contains three MEMS gyros, three high stability accelerometers and three inclinometers. The IMU is displayed in Figure A.6.

Figure A.6: STIM 300 inertial measurement unit ©Sensonor

Table A.6: Technical specification for the STIM 300 IMU

Name	Sensonor STIM 300
Gyroscope	Triaxial digital with ±400 °/s dynamic range
Gyroscope in run bias stability	0.3 °/h
Angular random walk	$0.15\,^{\circ}/\sqrt{\text{h}}$
Accelerometer	Triaxial digital with ±10 g dynamic range
Accelerometer in run bias stability	0.05 mg
Velocity random walk	$0.07\,\text{m}/\text{s}/\sqrt{\text{h}}$
Axis to axis misalignment error	±0.2 mrad
Axis to package misalignment error	±1 mrad
Size	44.8 mm x 38.6 mm x 21.5 mm
Weight	55 g
Input voltage	4.5 V to 5.5 V
Operating temperature range	−40 °C to 85 °C
Shock	1500g shock survivability

A.7 RTK GNSS Receivers

The final sensor described in this Appendix is the ublox NEO-M8T GNNS receiver [113]. Two independent receivers are used on-board the UAV and a base station enables real-time kinematic positioning. The receiver is displayed in Figure A.7.

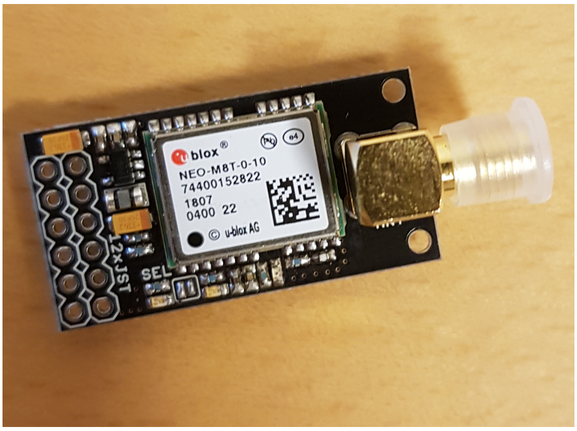


Figure A.7: Ublox NEO-M8T GNSS receiver

Table A.7: Technical specification for the NEO-M8T GNSS receiver

Name	Ublox NEO-M8T
GNSS	BeiDou, Galileo, GLONASS, GPS / QZSS
Number of concurrent GNSS	3
Global coverage	Yes
Interfaces	UART, USB, SPI and DDC
Update rate	10 Hz
Size (without carrier board)	12.2 mm x 16.0 mm x 2.4 mm
Weight	55 g
Input voltage	2.7 V to 3.6 V
Operating temperature range	−40 °C to 85 °C
Dynamics	$<$ 4g
Altitude	$<$ 50 000 m
Velocity	$<$ 500 m/s

Bibliography

[1] S. M. Albrektsen and T. A. Johansen. Syncboard - a high accuracy sensor timing board for uav payloads. *The International Conference on Unmanned Aircraft Systems*, pages 1706–1715., 2017. doi: 10.1109/ICUAS.2017.7991410.

[2] Analog. Analog devices adis16490 inertial measurement unit, 2019. URL `https://www.analog.com/en/products/adis16490.html`. Last accessed 2019-08-15.

[3] Y. Bar-Shalom and X.-R. Li. *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs Storrs, CT, 1995.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software.* John Wiley & Sons, 2004.

[5] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor. Vision-based target geo-location using a fixed-wing miniature air vehicle. *Journal of Intelligent and Robotic Systems*, 47(4):361–382, 2006.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

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

[8] B.K.P.Horn and B.G.Schunk. Determining optical flow. *Artif. Intell.*, 17: 185–204, 1981.

[9] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, Aug 1988. doi: 10.1109/9.1299.

[10] F. Bourgeois and J.-C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971. doi: 10.1145/362919.362945.

[11] E. F. Brekke and E. F. Wilthil. Suboptimal kalman filters for target tracking with navigation uncertainty in one dimension. In *2017 IEEE Aerospace Conference*, pages 1–11, March 2017. doi: 10.1109/AERO.2017.7943601.

[12] M. Bryson and S. Sukkarieh. Building a robust implementation of bearing-only inertial slam for a uav. *Journal of Field Robotics*, 24(1-2):113–143, 2007.

[13] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016. doi: 10.1109/TRO.2016.2624754.

[14] G. Cheng and J. Han. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:11 – 28, 2016. doi: https://doi.org/10.1016/j.isprsjprs.2016.03.014.

[15] K. Choi and I. Lee. A sequential aerial triangulation algorithm for real-time georeferencing of image sequences acquired by an airborne multi-sensor system. *Remote Sensing*, 5(1):57–82, 2013. doi: 10.3390/rs5010057.

[16] I. Colomina and P. Molina. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92:79–97, 2014. doi: 10.1016/j.isprsjprs.2014.02.013.

[17] Colregs. Convention on the international regulations for preventing collisions at sea, international maritime organization (imo), 1972.

[18] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1): 12–28, 2007. doi: 10.2514/1.22452.

[19] M. Daakir, M. Pierrot-Deseilligny, P. Bosser, F. Pichard, C. Thom, Y. Rabot, and O. Martin. Lightweight uav with on-board photogrammetry and single-frequency gps positioning for metrology applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127:115 – 126, 2017. doi: 10.1016/j.isprsjprs.2016.12.007.

[20] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu. Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications. *Journal of Communications*, 9:687–692, 09 2014. doi: 10.12720/jcm.9.9.687-692.

[21] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou. Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 145:3–22, 2018.

[22] P. Doherty and P. Rudol. A uav search and rescue scenario with human body detection and geolocalization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4830:1–13, 2007.

[23] S. J. Dumble and P. W. Gibbens. Efficient terrain-aided visual horizon based attitude estimation and localization. *Journal of Intelligent and Robotic Systems*, 78(2):205–221, 2015.

[24] O. Egeland and J. T. Gravdahl. *Modeling and simulation for automatic control.* Marine Cybernetics Trondheim, Norway, 2002.

[25] L. Elkins, D. Sellers, and W. R. Monach. The autonomous maritime navigation (amn) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles. *Journal of Field Robotics*, 27(6):790–818, 2010.

[26] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel. A complementary filter for attitude estimation of a fixed-wing uav. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 340–345, 2008. doi: 10.1109/IROS.2008.4650766.

[27] J. Farrell. *Aided Navigation: GPS with High Rate Sensors.* McGraw-Hill, Inc., New York, NY, USA, 1 edition, 2008.

[28] FLIR. Tau 2 longwave infrared thermal camera core, 2019. URL `https://www.flir.eu/products/tau-2/`. Accessed: 2019-08-28.

[29] T. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, 2011.

[30] L. Fusini, J. Hosen, H. H. Helgesen, T. A. Johansen, and T. I. Fossen. Experimental validation of a uniformly semi-globally exponentially stable non-linear observer for gnss- and camera-aided inertial navigation for fixed-wing uavs. In *Proc. of the International Conference on Unmanned Aircraft Systems*, pages 851–860, 2015.

[31] L. Fusini, T. I. Fossen, and T. A. Johansen. Nonlinear observers for gnss-and camera-aided inertial navigation of a fixed-wing uav. *IEEE Transactions on Control Systems Technology*, 26(5):1884–1891, 2018.

[32] A. R. Gerlach, S. G. Manyam, and D. B. Doman. Precision airdrop transition altitude optimization via the one-in-a-set traveling salesman problem. In *American Control Conference (ACC)*, 2016.

[33] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[34] K. Granstrom, C. Lundquist, and O. Orguner. Extended target tracking using a gaussian-mixture phd filter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3268–3286, 2012. doi: 10.1109/TAES.2012.6324703.

[35] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Attitude estimation using biased gyro and vector measurements with time-varying reference vectors. *IEEE Transactions on Automatic Control*, 57(5):1332—1338, 2012. doi: 10.1109/TAC.2011.2173415.

[36] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. Nonlinear observer for gnss-aided inertial navigation with quaternion-based attitude estimation. *American Control Conference*, pages 272–279., June 2013. doi: 10.1109/ACC.2013.6579849.

[37] K. Gryte, T. H. Bryne, S. M. Albreksteb, and T. A. Johansen. Field test results of gnss-denied inertial navigation aided by phased-array radio systems for uavs. *The International Conference on Unmanned Aircraft Systems*, 2019.

[38] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu. Advanced deep-learning techniques for salient and category-specific object detection: A survey. *IEEE Signal Processing Magazine*, 35(1):84–100, Jan 2018. doi: 10.1109/MSP.2017.2749125.

[39] Hardkernel. Odroid-xu4, 2019. URL `https://www.hardkernel.com/`. Last accessed 2019-08-28.

[40] W. Hartmann, S. Tilch, H. Eisenbeiss, and K. Schindler. Determination of the uav position by automatic processing of thermal images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39:B6, 2012. doi: 10.5194/isprsarchives-XXXIX-B6-111-2012.

[41] S. Harwin and A. Lucieer. Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (uav) imagery. *Remote Sensing*, 4(6):1573–1599, 2012. doi: 10.3390/rs4061573.

[42] H. H. Helgesen, F. S. Leira, T. A. Johansen, and T. I. Fossen. Tracking of marine surface objects from unmanned aerial vehicles with a pan/tilt unit using a thermal camera and optical flow. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 107–117, June 2016. doi: 10.1109/ICUAS.2016.7502581.

[43] H. H. Helgesen, F. S. Leira, T. A. Johansen, and T. I. Fossen. *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, chapter Detection and Tracking of Floating Objects Using a UAV with Thermal Camera, pages 289–316. Springer International Publishing, 2017. doi: 10.1007/978-3-319-55372-6_14.

[44] H. H. Helgesen, F. S. Leira, T. I. Fossen, and T. A. Johansen. Tracking of ocean surface objects from unmanned aerial vehicles with a pan/tilt unit using a thermal camera. *Journal of Intelligent and Robotic Systems*, 91(3): 775–793, Sep 2018. doi: 10.1007/s10846-017-0722-3.

[45] H. H. Helgesen, T. H. Bryne, E. F. Wilthil, and T. A. Johansen. Tracking of floating objects using fixed-wing uavs with compensation of navigation uncertainty. *IEEE Transactions on Aerospace and Electronic Systems*, 2019. Submitted.

[46] H. H. Helgesen, F. S. Leira, T. H. Bryne, S. M. Albrektsen, and T. A. Johansen. Real-time georeferencing of thermal images using small fixed-wing uavs in maritime environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 154:84 – 97, 2019. doi: https://doi.org/10.1016/j.isprsjprs.2019.05.009.

[47] H. H. Helgesen, F. S. Leira, and T. A. Johansen. Colored-noise tracking of floating objects using uavs with thermal cameras. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019.

[48] Ø. K. Helgesen, E. F. Brekke, H. H. Helgesen, and Ø. Engelhardtsen. Sensor combinations in hetereogeneous multi-sensor fusion for maritime target tracking. In *International Conference on Information Fusion (FUSION)*, 2019.

[49] E. M. Hemerly. Automatic georeferencing of images acquired by uav's. *International Journal of Automation and Computing*, 11(4):347–352, 2014. doi: 10.1007/s11633-014-0799-0.

[50] A. Hiba, T. Zsedrovits, P. Bauer, and A. Zarandy. Fast horizon detection for airborne visual systems. In *Proc. of the International Conference on Unmanned Aircraft Systems*, pages 886–891, 2016. doi: 10.1109/ICUAS.2016.7502661.

[51] J. Hosen, H. H. Helgesen, L. Fusini, T. I. Fossen, and T. A. Johansen. Vision-aided nonlinear observer for fixed-wing unmanned aerial vehicle navigation. *Journal of Guidance, Control, and Dynamics*, 39(8):1777–1789, 2016. doi: 10.2514/1.G000281.

[52] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[53] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, Aug 2004. doi: 10.1109/TSMCC.2004.829274.

[54] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz. Stereo vision–based navigation for autonomous surface vessels. *Journal of Field Robotics*, 28(1):3–18, 2011.

[55] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996. doi: 10.1109/70.538972.

[56] T. A. Johansen and T. Perez. Unmanned aerial surveillance system for hazard collision avoidance in autonomous shipping. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1056–1065, June 2016. doi: 10.1109/ICUAS.2016.7502542.

[57] A. Kadyrov, H. Yu, and H. Liu. Ship detection and segmentation using image correlation. In *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 3119–3126, Oct 2013.

[58] M. K. Kalandros, L. Trailovic, L. Y. Pao, and Y. Bar-Shalom. Tutorial on multisensor management and fusion algorithms for target tracking. In *Proceedings of the 2004 American Control Conference*, volume 5, pages 4734–4748, June 2004. doi: 10.23919/ACC.2004.1384060.

[59] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[60] T. Kirubarajan, Y. Bar-Shalom, W. Blair, and G. Watson. Immpdaf for radar management and tracking benchmark with ecm. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1115–1134, 1998.

[61] P. Konstantinova, A. Udvarev, and T. Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'03)*, pages 290–295, 2003.

[62] G. Koretsky, J. Nicoll, and M. Taylor. A tutorial on electro-optical/infrared (eo/ir) theory and systems. Technical report, Institute for defense analyses Alexandria VA, 2013.

[63] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, H. Tao, Y. Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, and P. Burt. Aerial video surveillance and exploitation. *Proceedings of the IEEE*, 89(10):1518–1539, Oct 2001. doi: 10.1109/5.959344.

[64] 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 *Proc. of the IEEE Aerospace Conference, Big Sky, US*, 2015.

[65] 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. *The International Conference on Unmanned Aircraft Systems*, pages 485–494, 2015. doi: 10.1109/ICUAS.2015.7152327.

[66] F. S. Leira, T. A. Johansen, and T. I. Fossen. A uav ice tracking framework for autonomous sea ice management. *The International Conference on Unmanned Aircraft Systems*, pages 581–590, June 2017. doi: 10.1109/ICUAS.2017.7991435.

[67] F. S. Leira, H. H. Helgesen, T. A. Johansen, and T. I. Fossen. Object detection, recognition and tracking from uavs using a thermal camera. *Journal of Field Robotics*, 2019. Submitted.

[68] Y. Li, S. Wang, Q. Tian, and X. Ding. A survey of recent advances in visual feature detection. *Neurocomputing*, 149:736 – 751, 2015. doi: https://doi.org/10.1016/j.neucom.2014.08.003.

[69] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11 – 26, 2017. doi: https://doi.org/10.1016/j.neucom.2016.12.038.

[70] D. Lowe. Object recognition from local scale-invariant features. *Proc. of the International Conference on Computer Vision*, pages 1150–1157, 1999.

[71] R. P. Mahler. *Statistical multisource-multitarget information fusion*, volume 685. Artech House Norwood, MA, 2007.

[72] R. P. S. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, Oct 2003. doi: 10.1109/TAES.2003.1261119.

[73] R. Mahony, T. Hamel, and J. M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, June 2008.

[74] D. B. Malkoff. Evaluation of the jonker-volgenant-castanon (jvc) assignment algorithm for track association. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 228–239. International Society for Optics and Photonics, 1997.

[75] M. Mallick and B. La Scala. Comparison of single-point and two-point difference track initiation algorithms using position measurements. *Acta Automatica Sinica*, 34(3):258–265, 2008.

[76] M. Mammarella, G. Campa, M. L. Fravolini, and M. R. Napolitano. Comparing optical flow algorithms using 6-dof motion of real-world rigid objects. *IEEE Transactions on systems, Man, and Cybernetics*, 42(6):1752–1762, 2012.

[77] F. L. Markley. Attitude error representations for kalman filtering. *Journal of Guidance, Control, and Dynamics*, 26(2):311–317, 2003. doi: 10.2514/2.5048.

[78] C. Martínez, I. F. Mondragón, P. Campoy, J. L. Sánchez-López, and M. A. Olivares-Méndez. A hierarchical tracking strategy for vision-based applications on-board uavs. *Journal of Intelligent and Robotic Systems*, 72(3): 517–539, 2013.

[79] S. G. Mathisen, F. S. Leira, H. H. Helgesen, K. Gryte, and T. A. Johansen. Autonomous ballistic airdrop of objects from a small fixed-wing unmanned aerial vehicle. *Autonomous Robots*, 2019. Conditionally accepted.

[80] P. McGill, K. Reisenbichler, S. Etchemendy, T. Dawe, and B. Hobson. Aerial surveys and tagging of free-drifting icebergs using an unmanned aerial vehicle (uav). *Deep Sea Research Part II: Topical Studies in Oceanography*, 58, 2011.

[81] M. Muja and D. G. Lowe. Flann, fast library for approximate nearest neighbors. In *Proc. of the International Conference on Computer Vision Theory and Applications*, 2009.

[82] D. Musicki and R. Evans. Joint integrated probabilistic data association: Jipda. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, July 2004. doi: 10.1109/TAES.2004.1337482.

[83] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, June 1994. doi: 10.1109/9.293185.

[84] P. C. Niedfeldt and R. W. Beard. Multiple target tracking using recursive ransac. *2014 American Control Conference*, pages 3393–3398, June 2014. doi: 10.1109/ACC.2014.6859273.

[85] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart. A uav system for inspection of industrial facilities. *Proceedings of the 2013 IEEE Aerospace Conference*, pages 1–8, March 2013. doi: 10.3929/ethz-a-010061810.

[86] R. Y. Novoselov, S. M. Herman, S. M. Gadaleta, and A. B. Poore. Mitigating the effects of residual biases with schmidt-kalman filtering. In *2005 7th International Conference on Information Fusion*, volume 1, July 2005. doi: 10.1109/ICIF.2005.1591877.

[87] S. Ojha and S. Sakhare. Image processing techniques for object tracking in video surveillance- a survey. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–6, Jan 2015. doi: 10.1109/PERVASIVE.2015.7087180.

[88] OptiTrack Motion Capture Systems, 2018. URL `http://optitrack.com`. Last accessed 2018-08-14.

[89] G. Pajares. Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs). *Photogrammetric Engineering and Remote Sensing*, 81(4):281–329, 2015. doi: 10.14358/PERS.81.4.281.

[90] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):1993–2016, Aug 2017. doi: 10.1109/TITS.2016.2634580.

[91] A. Qadir, J. Neubert, and W. Semke. On-board visual tracking with unmanned aircraft system (uas). In *AIAA Infotech at Aerospace Conference and Exhibit 2011*, 2011.

[92] M. Radovic, O. Adarkwa, and Q. Wang. Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, 3(2):21, Jun 2017. doi: 10.3390/jimaging3020021.

[93] M. Rehak, R. Mabillard, and J. Skaloud. A micro-uav with the capability of direct georeferencing. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*, pages 317–323, 2013. doi: 10.5194/isprsarchives-XL-1-W2-317-2013.

[94] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979. doi: 10.1109/TAC.1979.1102177.

[95] M. R.-B.-. retractable pan and tilt gimbal, 2019. URL `http://microuav.com/BallTurretCamera/BTC-88`. Last accessed 2019-08-15.

[96] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina. A tutorial on bernoulli filters: theory, implementation and applications. *IEEE Transactions on Signal Processing*, 61(13):3406–3430, 2013.

[97] 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)*, pages 1–8, July 2018. doi: 10.1109/IJCNN.2018.8489465.

[98] X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct 2003. doi: 10.1109/TAES.2003.1261132.

[99] P. Rudol and P. Doherty. Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery. In *Proc. of the IEEE Aerospace Conference*, 2008.

[100] J. Y. Sakamaki, R. W. Beard, and M. Rice. Tracking multiple ground objects using a team of unmanned air vehicles. In T. I. Fossen, K. Y. Pettersen, and H. Nijmeijer, editors, *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, pages 249–268. Springer International Publishing, 2017. doi: 10.1007/978-3-319-55372-6_12.

[101] Sensonor. Stim 300 inertial measurement unit, 2019. URL `https://www.sensonor.com/products/inertial-measurement-units/stim300/`. Last accessed 2019-08-28.

[102] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, and C. Silvestre. Landing of a quadrotor on a moving target using dynamic image-based visual servo control. *IEEE Transactions on Robotics*, 32(6):1524–1535, Dec 2016. doi: 10.1109/TRO.2016.2604495.

[103] J. Shi and C. Tomasi. Good features to track. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[104] 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, June 2015. doi: 10.1109/ICUAS.2015.7152377.

[105] P. Soille. *Morphological Image Analysis: Principles and Applications.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition, 2003.

[106] J. Sola. Quaternion kinematics for the error-state kf, 2017. URL `http://www.iri.upc.edu/people/jsola/JoanSola/objectes/notes/kinematics.pdf`. Last accessed 2018-02-02.

[107] M. L. Sollie, T. H. Bryne, and T. A. Johansen. Pose estimation of uavs based on ins aided by two independent low-cost gnss receivers. *The International Conference on Unmanned Aircraft Systems*, 2019.

[108] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985. doi: http://dx.doi.org/10.1016/0734-189X(85)90016-7.

[109] V. Sze, Y. Chen, T. Yang, and J. S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, Dec 2017. doi: 10.1109/JPROC.2017.2761740.

[110] C. Teuliere, L. Eck, and E. Marchand. Chasing a moving target from a flying uav. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4929–4934, Sept 2011. doi: 10.1109/IROS.2011.6094404.

[111] ThermalCapture Grabber USB 640 - TEAX Technology, 2018. URL `http://thermalcapture.com/thermalcapture-grabber-usb-640/`. Last accessed 2019-08-28.

[112] C. Toth and G. Jóźków. Remote sensing platforms and sensors: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:22 – 36, 2016. doi: 10.1016/j.isprsjprs.2015.10.004.

[113] ublox. Neo-m8t gnss receiver, 2019. URL `https://www.u-blox.com/en/product/neolea-m8t-series`. Last accessed 2019-08-28.

[114] H. Vermeille. Computing geodetic coordinates from geocentric coordinates. *Journal of Geodesy*, 78(1):94—95, 2004. doi: 10.1007/s00190-004-0375-4.

[115] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[116] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007. doi: 10.1177/0278364907081229.

[117] G. Welch and G. Bishop. An introduction to the kalman filter, 1995. URL `http://www4.cs.umanitoba.ca/~jacky/Robotics/Papers/kalman-filter.pdf`. Last accessed 2019-09-12.

[118] E. F. Wilthil and E. F. Brekke. Compensation of navigation uncertainty for target tracking on a moving platform. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1616–1621, July 2016.

[119] E. F. Wilthil, A. L. Flåten, and E. F. Brekke. *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, chapter A Target Tracking System for ASV Collision Avoidance Based on

the PDAF, pages 269–288. Springer International Publishing, 2017. doi: 10.1007/978-3-319-55372-6_13.

[120] M. T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger. 360-degree visual detection and target tracking on an autonomous surface vehicle. *Journal of Field Robotics*, 27(6): 819–833, 2010.

[121] H. Xiang and L. Tian. Method for automatic georeferencing aerial remote sensing (rs) images from an unmanned aerial vehicle (uav) platform. *Biosystems Engineering*, 108(2):104 – 113, 2011. doi: 10.1016/j.biosystemseng.2010.11.003.

[122] S. Yahyanejad and B. Rinner. A fast and mobile system for registration of low-altitude visual and thermal aerial images using multiple small-scale uavs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:189 – 202, 2015. doi: 10.1016/j.isprsjprs.2014.07.015.

[123] S. Yahyanejad, J. Misiorny, and B. Rinner. Lens distortion correction for thermal cameras to improve aerial imaging with small-scale uavs. *IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pages 231–236, Sept 2011. doi: 10.1109/ROSE.2011.6058528.

[124] C. Yang, E. Blasch, and P. Douville. Design of schmidt-kalman filter for target tracking with navigation errors. In *IEEE Aerospace Conference*, pages 1–12, March 2010. doi: 10.1109/AERO.2010.5446689.

[125] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823 – 3831, 2011. doi: http://dx.doi.org/10.1016/j.neucom.2011.07.024.

[126] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006.

[127] X. Yu and Y. Zhang. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences*, 74:152 – 166, April 2015.

[128] C. Yuan, Y. Zhang, and Z. Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian Journal of Forest Research*, 45(7): 783–792, 2015. doi: 10.1139/cjfr-2014-0347.

[129] N. M. Zaitoun and M. J. Aqel. Survey on image segmentation techniques. *Proceedings of the International Conference on Communications, management, and Information technology (ICCMIT)*, pages 797 – 806, 2015. doi: https://doi.org/10.1016/j.procs.2015.09.027.

[130] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000. URL `https://www.microsoft.com/en-us/research/publication/a-flexible-new-technique-for-camera-calibration/`.

[131] S. Zhao, F. Lin, K. Peng, X. Dong, B. M. Chen, and T. H. Lee. Vision-aided estimation of attitude, velocity, and inertial measurement bias for uav stabilization. *Journal of Intelligent and Robotic Systems*, 81(3):531–549, 2016.

[132] G. Zhou, C. Li, and P. Cheng. Unmanned aerial vehicle (uav) real-time video registration for forest fire monitoring. *The IEEE International Geoscience and Remote Sensing Symposium.*, 3:1803–1806, July 2005. doi: 10.1109/IGARSS.2005.1526355.

Doctoral Thesis

Håkon Hagen Helgesen

ISBN 978-82-326-4344-8 (printed version)
ISBN 978-82-326-4345-5 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2019:373

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

NTNU

Doctoral theses at NTNU, 2019:373

Håkon Hagen Helgesen

Detection and Tracking of Floating Objects using UAVs with Optical Sensors

NTNU
Norwegian University of
Science and Technology