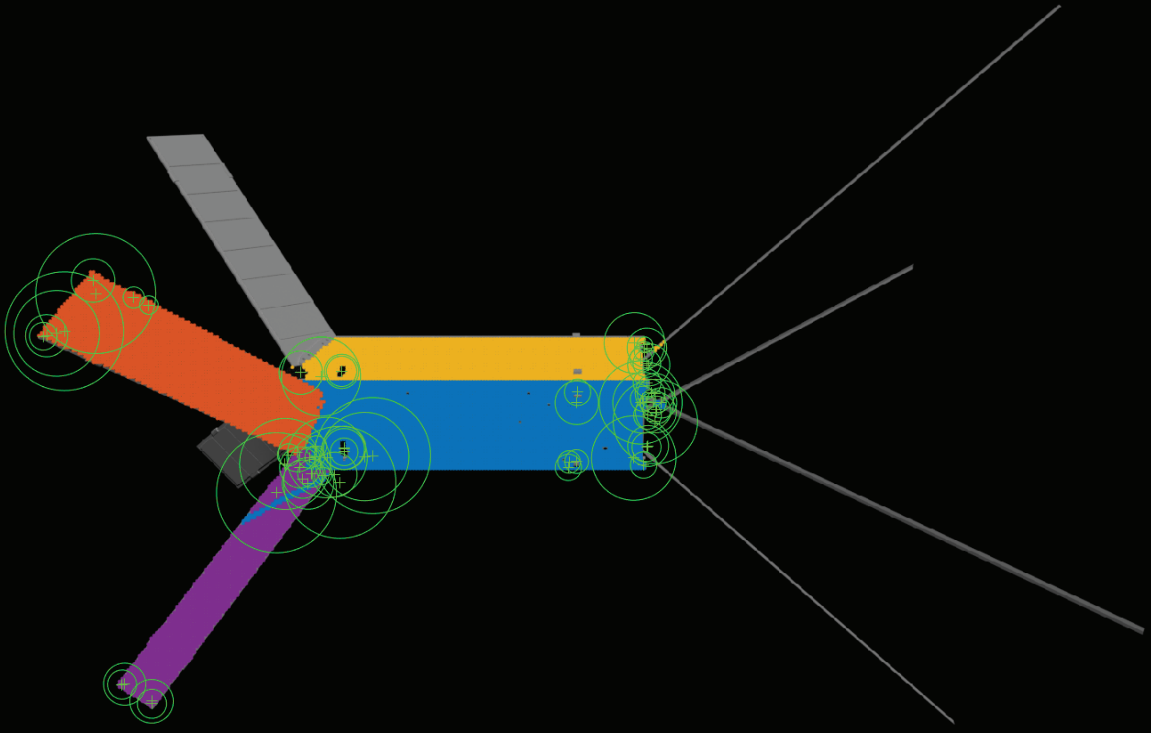
MSc. Thesis in Aerospace Engineering

Visual-Lidar Feature Detection for Relative Pose Estimation of an Unknown Spacecraft

Toon Stolk
2022



TU Delft

MSc thesis in Aerospace Engineering

Visual-Lidar Feature Detection for Relative Pose Estimation of an Unknown Spacecraft

Toon Stolk

October 2022

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering

Toon Stolk: *Visual-Lidar Feature Detection for Relative Pose Estimation of an Unknown Spacecraft* (2022)

To be defended publicly on Thursday October 20, 2022 at 09:30 AM.

Student name:	Toon Carolus Hermanus Stolk	
Student number:	4488458	
Project duration:	July 2021 - October 2022	
Thesis committee:	Prof. dr. E.K.A. Gill	TU Delft, Committee chair
	Dr. ir. J. Guo	TU Delft, Thesis supervisor
	Dr. ir. E. Mooij	TU Delft, External examiner

An electronic version of this thesis is available at: http://repository.tudelft.nl/.

Abstract

There is no denying that the ever-increasing demand in space endeavours calls for more sustainable exploitation of the space environment. Approaches to mitigate space debris, such as Active Debris Removal (ADR), are frequently suggested. Most of these proposals suggest a single sensor approach to provide accurate and continuous shape and pose estimation of the target. However, a more robust system can be developed by relying on input from multiple sensors with different modalities.

Compared to earlier literature, several alternative multimodal methods using visual-Lidar data have been researched during this thesis, of which the most promising method has been investigated in more detail. The method suggests to determine the 3D location of 2D features by projecting these on detected 3D planes, thereby fusing the visual-Lidar data at feature level. The visual-Lidar data has been acquired through both simulation in Blender using Blensor and through experimentation using an visual camera, a scanning Lidar and a robotic arm.

To verify and validate the proposed multimodal feature detection method, the detected 3D features are compared to the ground truth directly. Next to that, the method is also verified through analysis of the end-to-end process to estimate the relative pose of an unknown target. Where the resulting 3D features serve as input into an particle filter, combined with an Extended Kalman Filter (EKF), based on the FastSLAM algorithm. The proposed method showed promising results, encouraging further research to determine the pose based on plane normal vectors for the relative pose estimation to operate during adverse illumination conditions.

Keywords: Spacecraft Relative Navigation, Multisensor Data Fusion, Relative Pose Estimation, Machine Vision, Visual-Lidar

Preface

Reflecting on this thesis I can honestly conclude that this project has been the greatest challenge in my life, so far. The thesis has been a great learning experience, introducing me to previously unfamiliar topics, ranging from computer vision to robot motion planning. The latter would not have been possible without the dedication and assistance from Joshua Spaander. I greatly appreciated and enjoyed our cooperation on the Franka robotic arm and the interesting discussions during and thereafter.

Likewise, I would like to express my gratitude towards my supervisor Dr. Jian Guo. Granting me the opportunity to work on this ambitious project and guiding me along the way, supporting me to be decisive and critical on my work. Furthermore, I would like to thank Giovanni Franzese from Cognitive Robotics at 3ME. For educating me in robotics and for allowing my experiments to continue, especially when it seemed infeasible at certain times.

I am sincere when I say that it is due to my friends, family, peers but especially due to Maartje that I made it through this project. Supporting me after setbacks and boosting my self-confidence, giving me the strength to continue and to take on the next challenge. Without you I would still be at a stage far from ready for my defence and graduation, thank you!

Finally, I will always appreciate that I was able to study and pursue my ambitions here in Delft, both on- and off- campus. Therefore Delft will always be a special place for me. I can say that the hard work payed dividends to my personal and professional growth, which now culminates in the completion of my masters degree at Aerospace Engineering. I am looking forward to the privilege to call myself an engineer and to contribute to upcoming exciting projects!

Toon Stolk
The Hague, October 2022

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

ADC	Attitude Determination & Control 7
ADR	Active Debris Removal v, 1, 2, 4–7, 73, 75
AMCW	Amplitude Modulated Continuous Wave 82, 83
CCD	Charge-Coupled Device 9, 10, 80
CMOS	Complementary Metal-Oxide Semiconductors 46, 80
COTS	Commercially-Of-The-Shelf 2, 38
DIPD	Depth Image-based Plane Detection 23
DOF	Degrees Of Freedom 7, 28, 32, 42, 44, 54, 61, 62, 69, 79
EKF	Extended Kalman Filter v, 10, 16, 22, 27–30, 64, 69, 74
EKPF	Extended Kalman Particle Filter 10
EO	Electro-Optical 12, 47, 77, 79
ESA	European Space Agency 1
EVA	Extravehicular Activity 7
FMCW	Frequency Modulated Continuous Wave 82, 83
FOV	Field Of View 12, 20, 45, 46, 83
GNC	Guidance, Navigation & Control xi, 4–6, 11, 12, 14, 15, 77, 79
GNSS	Global Navigation Satellite System 7
IADC	Inter-Agency Space Debris Coordination Committee 1
ICP	Iterative Closest Point 10
IFOV	Instantaneous Field Of View 55
IR	InfraRed 45, 75, 79, 84
LED	Light-Emitting Diode 78
MATLAB	MATrix LABoratory xi, 17–19, 21, 44, 51, 59, 85
MRF	Markov Random Field 2, 3, 10, 17, 73, 74
MSAC	M-estimator SAmple Consensus 19
OOP	Object-Oriented Programming 71
OOS	On-Orbit Servicing 4, 7
ORB	Oriented FAST and rotated BRIEF 10
PF	Particle Filter 10
PMD	Photonic Mixer Device 9, 10
RANSAC	RANdom SAmple Consensus 10, 19, 23, 56, 70, 73
RGB	Red-Green-Blue 10, 17, 18
ROI	Region Of Interest xi, 51
RPO	Rendezvous & Proximity Operations xi, 4–7, 11, 12, 14

SAD	Sum of Absolute Difference 10
SAR	Synthetic Aperture Radar 7
SFM	Structure From Motion 9
SIFT	Scale-Invariant Feature Transform 2, 19
SLAM	Simultaneous Localization And Mapping 9, 10
SLR	Satellite Laser Ranging 7, 79
SNR	Signal-to-Noise Ratio 12, 39, 47, 48, 84
SURF	Speeded Up Robust Features 10, 19, 53, 59, 61, 70, 74
TOF	Time-Of-Flight 9, 41, 82, 83
UKF	Unscented Kalman Filter 10
UV	UltraViolet 79

1. Introduction & Overview

Considering the continuous and expanding exploitation of the space environment near the Earth, it is apparent that it is not the question if, but when this environment will become cluttered completely. The fact that the exploitation of the space environment goes hand in hand with the creation of space debris cannot be understated. It is acknowledged that the removal of space debris, or Active Debris Removal (ADR), is required to keep the near-Earth environment sustainable for future exploitation. ADR is a complex activity which is actively researched throughout the astronautical community. The aim of this thesis is to contribute knowledge on the relative pose estimation of an uncooperative target with no a-priori information. Specifically, to research multimodal feature detection for relative pose estimation of an uncooperative unknown spacecraft using visual-Lidar data.

In this chapter the research topic and its context will be introduced. The first section will provide more background information on the problem of ADR and the associated pose estimation. Afterwards, in Section 1.2, the research is formalised with a research proposal and aims to inform the reader on the motivation of the proposed multimodal approach. The research is structured by formulating research questions, which are presented in Section 1.3. The next section, Section 1.4, will provide a clear demarcation of the scope of this thesis. The last section, Section 1.5, will provide an overview of this document and the subjects covered in this thesis.

1.1. Background & Motivation

The risk of collision between a spacecraft and another satellite was already suggested before man-made spacecraft reached orbit around Earth [1]. These collisions can occur with any kind of object at various velocities with potential dire consequences. Unfortunately, these consequences are not only restricted to the directly involved objects. Collisions in space create new objects, which again can cause collisions, creating a cascading effect known as the Kessler Syndrome [2]. Consequently, entire regions around Earth could become inaccessible, hindering the advantageous exploitation of the space environment, examples of which are telecommunications and climate research from space.

Every launch introduces new space debris to the space environment, ranging from flecks of paint to entire defunct satellites, such as the Envisat operated by European Space Agency (ESA). Space agencies have made agreements to mitigate space debris creation as much as possible. Although, according to the Inter-Agency Space Debris Coordination Committee (IADC), even in almost perfect scenarios with 90% mitigation adherence and no collisions, the space debris will increase nonetheless[1]. Therefore, research has shown that at least 5 pieces of space debris should be removed from orbit each year to sustain the current accessibility of the space environment [3]. It is proposed to remove pieces of space debris in so-called Active Debris Removal missions, currently researched and designed by the astronautical community. The implementation of ADR is becoming increasingly relevant, especially considering the commercialisation of the space sector and the arrival of mega-constellations, such as Starlink.

The current state-of-the-art ADR concepts propose that a dedicated spacecraft, referred to as the servicer or the chaser, is launched to consecutively rendezvous, inspect, capture and de-orbit the debris, referred to as the client or the target [4]. The capture phase relies on accurate estimation of the relative position and attitude, referred to as the pose, of the target. All the while accounting for rapidly changing illumination conditions, such as eclipses and Sun rises. This estimation problem becomes increasingly complex if no a-priori information, such as the features, of the target is available, i.e. the target is unknown. In addition, ADR missions require a high level of autonomy, due to signal delay and lack of coverage [5]. These challenges have to be overcome to enable the success of an ADR mission, to clear space debris and to sustain the current space environment.

[1] Retrieved from: `https://www.esa.int/Safety_Security/Space_Debris/Active_debris_removal`, accessed at: 09-05-2022.

1.2. Research Proposal

As suggested in the section above, the accurate relative pose estimation is crucial in an ADR mission. In addition, it is apparent that the pose estimation without a-priori information requires the most research [6]. In addition, methods developed for unknown targets might accommodate targets that have suffered damage, since the presumed spacecraft model is no longer correct. This, in turn, could lead to a wrong pose estimation when using a method developed specifically for a known target. Next to this absence of knowledge on the target, the pose estimation of the target is considered using a multimodal approach in this thesis. This multimodal pose estimation makes use of multisensor data fusion with sensors that sense different modalities. Specifically, it is proposed to research the application of multisensor data fusion from a visual camera and Lidar for relative pose estimation without a-priori information.

In several other fields, when multisensor data fusion was implemented correctly, it provided the following benefits compared to single sensor approaches: improvements in quality, availability and reliability [7]. Furthermore, there is steady increase in high quality Commercially-Of-The-Shelf (COTS) sensors from the automotive and industrial sector. In addition to the lower price, the newly developed sensors are less bulky, cheaper and require less power. Despite the advances in the previously characteristics, the multimodal approach will always be more demanding compared to a single sensor approach. However, if the advantages prove to be worthwhile, the sensor budgets may be relaxed to accommodate the extra sensor, especially on a spacecraft designed specifically for an ADR mission.

These potential benefits have not gone unnoticed in the astronautical community, multimodal relative pose estimation methods have been suggested in literature before. However, the research using a multimodal approach for uncooperative and unknown targets specifically, is limited compared to cooperative and known targets. The research which specifically takes uncooperative unknown targets into consideration often rely on assumptions which do not generalise well to reality, such as the assumption that "the features are perfectly tracked and matched between frames" [8, p. 5]. One notable exception is the work by Hao et al. [9], where a relative pose is accurately determined using a Lidar and visual camera, without strict assumptions. It is important to note that in this work the feature locations are estimated by combining Scale-Invariant Feature Transform (SIFT) with a Markov Random Field (MRF) model.

For research to add to true state-of-the-art knowledge, knowledge gaps within the context of this topic were identified before the start of the research. Which are the following:

1. Multisensor data fusion in context of spacecraft relative navigation
2. Capability of real-time computation for relative pose estimation algorithms
3. Algorithm sensitivity on sensor performance under conditions in a space environment
4. Scalability of sensors from ground experiments to space environment

These knowledge gaps formed a good basis for the research after the literature study. However, as the research progressed, it became apparent that the main knowledge gap was more related to multimodal feature detection rather than the relative pose estimation. Therefore, the main knowledge gap which will be handled is the first one: Multisensor data fusion in context of spacecraft relative navigation, with a special note towards multimodal feature detection. When applicable the other knowledge gaps will be taken into account as much as possible, such as the scalability of the experiment.

To conclude, the aforementioned potential benefits, current sensor development trends and apparent knowledge gaps in the field are the main causes to research multimodal relative pose estimation. Which is done specifically in the context of spacecraft relative pose estimation with an uncooperative unknown target. This research is represented by the following research objective:

> "To develop methods for, and gain insights into, the relative pose estimation of an unknown uncooperative target using multisensor data fusion by means of computational simulation, analysis and experimentation."

The ultimate goal would be that, as a result from this research, an approach would be produced which provides the benefits of improved quality, availability and reliability, which is validated through real-world experimentation. This objective and approach is quite broad, therefore a more detailed approach to this research is established in Chapter 3, after relevant literature has been presented in Chapter 2.

1.3. Research Questions

To guide the thesis towards academic value, this thesis will investigate several research questions which will cover the relevant knowledge gaps as much as possible. It is important to note that the research questions should be manageable within the constraints of the MSc thesis, with respect to both time and resources. Each research question is accompanied by its associated motivation for that question[2].

RQ-1 *How can visual-Lidar data be used for multimodal feature detection?*

In other pose estimation approaches utilizing visual-Lidar data, the depth information from the Lidar is primarily used to determine the scale of the target. It would be interesting to investigate if other methods can be implemented to perform 3D feature detection.

RQ-1.a *Next to the Markov Random Field (MRF), are there other ways to fuse visual-Lidar data to benefit multimodal feature detection?*

In the context of spacecraft relative pose estimation with an unknown uncooperative target, the only presented methods involve 3D feature detection through the use of an MRF. What other methods can be used, or developed, to detect features using multisensor data fusion?

RQ-1.b *How can the proposed multimodal feature detection method be verified?*

If other methods show to be promising, how can these methods be verified? Oftentimes an end-to-end process is carried out, which relies on the proposed computer vision method, thereby verifying that the method is indeed implemented correctly. However, can this be done using another method? Which will provide insights into, and verify, the proposed method?

RQ-2 *How can the fused visual-Lidar data be used for relative pose estimation?*

For other relative pose estimation algorithms, it is not directly apparent how the input data relates to the estimation method. Therefore it should be investigated how the features, detected through multimodal feature detection, should be used for the relative pose estimation process.

RQ-2.a *How should the 3D feature derived from visual-Lidar fusion be used for relative pose estimation?*

If possible, can the detected 3D features be used for spacecraft relative pose estimation? What specific changes to the filter are required to account for the different features, since most probably changes are required for the measurement model or equations of motion.

RQ-2.b *What are the benefits of visual-Lidar multisensor data fusion on relative pose estimation?*

Because a method is valid, does not mean it should be used, the advantages should be weighed against the disadvantages. Is the application of the proposed method actually beneficial and why?

1.4. Research Scope

A clear demarcation of the research scope will assist to focus the specified topic to a certain depth. This focus allows for an actual contribution to the state-of-the-art knowledge rather than a high level analysis of the topic. In addition, the scope helps other researchers to get involved or continue the research, since the limitations of the work are clearly documented.

As proposed in the sections above, the topic of this thesis is multimodal relative pose estimation of an uncooperative unknown target. From this title alone, certain scope demarcations, namely three, can readily be distinguished. Firstly, this work will only focus on a multimodal method for relative pose estimation, thus no single sensor approaches will be investigated. Furthermore, only combinations of cameras in the visual spectrum and Lidar are considered, combinations with other sensors, such as infrared cameras, are not considered. Secondly, only relative pose estimation will be carried out, no

[2] In the case that some terms and definitions are currently unfamiliar to the reader, the most important relevant literature will be reviewed in Chapter 2.

control aspects will be covered in this thesis. Thirdly, the work will focus on relative pose estimation of a single uncooperative unknown target. Although, most probably the investigated methods can be benefit relative navigation with cooperative or known targets as well.

This thesis will place the research in the context of ADR, however other applications of pose estimation are possible, such as On-Orbit Servicing (OOS). Additionally, this work will cover some additional required topics for multimodal relative pose estimation, such as computer vision and multisensor data fusion. Lastly, the data required for the research will originate from simulations and lab-based experiments. These different aspects which lie either within the scope (green in blue) or just out of scope (red) are visualised in the figure below:

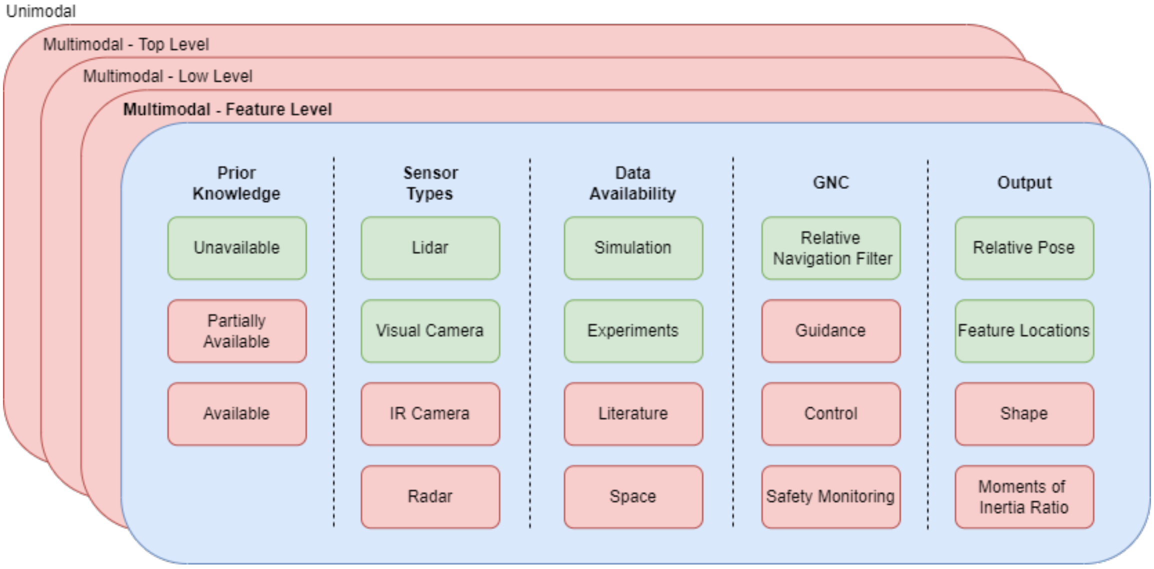


Figure 1.1.: Visualisation of the scope of this thesis, featuring the rough building blocks of the GNC-loop of an RPO mission. Green and blue indicate what is considered in the scope, red indicates what is considered out of scope for this thesis.

1.5. Thesis Overview

In this chapter, the problem of pose estimation of an uncooperative unknown target has been introduced. The main aspects of this problem is the absence of a feature map and the need for autonomy. As a potential improvement on existing methods, a multimodal approach using visual-Lidar fusion was suggested to determine the pose of the target. In this context the research was focused on multimodal feature detection, within the scope presented in Section 1.4.

The remainder of this thesis will cover the work that has been done to carry out the research, developing the multimodal feature estimation method and implementation of the accompanying pose estimation algorithm. The next chapter will provide an overview of the relevant literature for this research, which is supported by the essentials on sensors, covered in Appendix A. After the relevant literature has been discussed, Chapter 3 will be dedicated to the assumptions and requirements for the different elements of this thesis. But most importantly, the research approach will be presented along with some high-level design decisions.

After the research approach has been covered, the development on multimodal feature detection methods is presented in Chapter 4. Afterwards, using a promising multimodal feature detection method, the implementation of the pose estimation algorithm is presented in Chapter 5. The data which is used for the development and analysis is derived computationally through simulation or from experiments, which is described in Chapter 6. Afterwards, the results, verification and validation are discussed in Chapter 7. Finally, the conclusion and recommendations for this work are presented in Chapter 8.

2. Overview of Relevant Literature

In this chapter the state-of-the-art on the disciplines related to the thesis topic will be covered in more detail. This chapter will start off in a more general sense, getting more specific for the thesis topic as the chapter progresses. This will cover topics ranging from spacecraft relative navigation to spacecraft relative pose estimation using multisensor data fusion. For the sake of conciseness, the required theoretical basis for the topic related to sensors is covered in Appendix A. Which might prove useful especially for the readers that are not that familiar with the topic and the disciplines related to spacecraft relative pose estimation.

Firstly in this chapter, the general topic of spacecraft relative navigation will be covered in Section 2.1, which will provide a clear overview of its distinct aspects and applications. Secondly, the literature on spacecraft relative pose estimation will be discussed in Section 2.2, which will cover spacecraft relative pose estimation methods in general. Lastly, the topic of multisensor data fusion is covered in Section 2.3, which also covers the literature on multimodal spacecraft relative pose estimation methods.

It should be noted that the literature covered in this chapter and Appendix A have been mostly covered in the literature study at the start of the thesis. Where some aspects have changed to better reflect the actual work of the thesis and some have not, finding their way in this overview of relevant literature again.

2.1. Spacecraft Relative Navigation

Spacecraft relative navigation can be classified as a Guidance, Navigation & Control (GNC) process. It consists of determining the relative pose and manoeuvring with respect to another satellite, which can be artificial or natural, singular or multiple. As mentioned in the ADR example in the Introduction, the dedicated spacecraft is referred to as the servicer or the chaser, whereas the other satellite, or satellites, is referred to as the client or the target. Activities that make use of spacecraft relative navigation are often referred to as Rendezvous & Proximity Operations (RPO).

Relative pose estimation is only a small part of these activities, furthermore, in some cases it might not even be required at all. However, when the target and chaser are in close proximity of each other, relative pose estimation is always required to prevent a potential collision. In Figure 2.1, the separate rendezvous phases for an RPO mission are visualised.

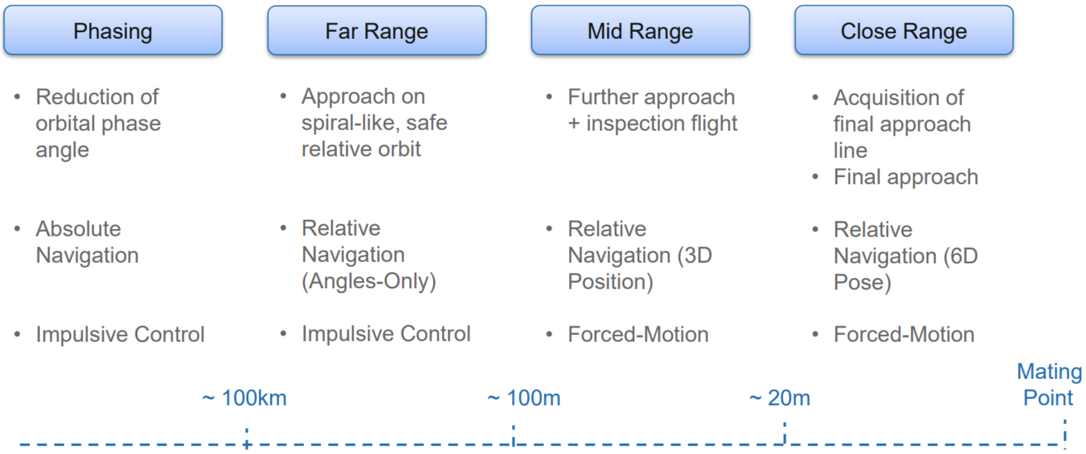

Phasing
Far Range
Mid Range
Close Range
Reduction of orbital phase angle
Approach on spiral-like, safe relative orbit
Further approach + inspection flight
Acquisition of final approach line
Final approach
Absolute Navigation
Relative Navigation (Angles-Only)
Relative Navigation (3D Position)
Relative Navigation (6D Pose)
Impulsive Control
Impulsive Control
Forced-Motion
Forced-Motion
~ 100km
~ 100m
~ 20m
Mating Point

Figure 2.1.: Infographic showing the different phases of an RPO mission [10].

From the overview above, Figure 2.1, the requirement for relative navigation and motion control can be seen, for phases indicated by rough distance demarcations. Most notably, the close range phase shows that the full pose, position and attitude, of the target needs to be known. As previously discussed in Section 1.4, this thesis will focus on feature detection methods for close range relative navigation. When taking a closer look at the close range phase, the aspects of the GNC can be distinguished, as visualised in Figure 2.2.

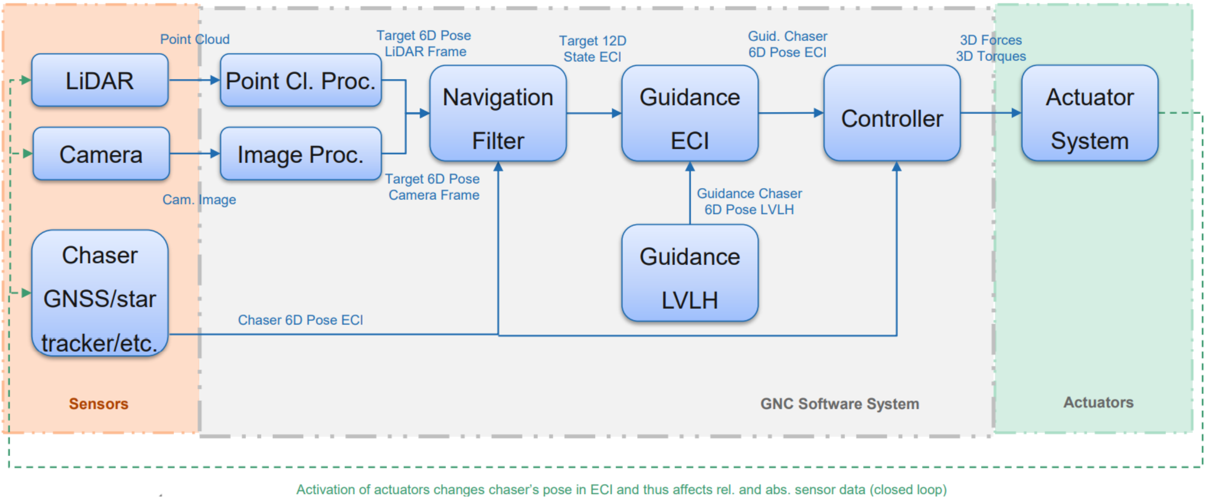


Figure 2.2.: GNC control loop for a close range RPO mission [10].

In line with the thesis scope, the topics depicted in the top left are those relevant for this particular thesis topic, which are the following: Lidar, visual camera, point cloud processing, image processing and navigation filters. In addition to these relevant topics, multisensor data fusion and associated methods will be discussed in the remainder of this chapter and Appendix A. But first, some additional applications of spacecraft relative navigation will be presented, to provide a clearer overview of the relevance of this subject in general.

Applications of Spacecraft Relative Navigation

In Chapter 1, the problem of relative pose and shape estimation was introduced in the context of ADR. However, the process of relative pose estimation is applied in several different missions which have aspects of Rendezvous & Proximity Operations (RPO). Some of these are currently only proposed or are still in development, while others have already been done before. To sketch how relative pose estimation plays a role in these different type of activities, two activities other than ADR will be presented. In particular, the aspect of relative pose estimation will by highlighted.

Formation Flying

Similar to the RPO activities, formation flying can be considered to be a more general term for relative navigation, since a formation can be flown close by, or even several hundreds of kilometers apart. Therefore, an argument can be made that each controlled close proximity mission is a formation flying mission.

As stated before, the focus of this work shall be more towards the relative navigation in case of close proximity flying. The formations that fly further apart have different mission goals, such as Synthetic Aperture Radar (SAR) interferometry, magnetospheric observation and gravimetry. Even though the relative distance is several magnitudes larger, the required accuracy for relative positioning can be in order of millimeters [11]. Although, these relative navigation techniques often involve offline methods, using measurements from Global Navigation Satellite System (GNSS), Radio Frequency-Based Relative Navigation, and Satellite Laser Ranging (SLR). However, these techniques are generally not suited for close range relative pose and shape estimation [4].

On orbit Servicing

On-Orbit Servicing (OOS) is not a new concept. In fact, in 1973 the first OOS mission was executed to service the Skylab module. In that particular case, astronauts fitted the module with a parasol to return the spacecraft to an acceptable thermal configuration which saved the mission [12]. And various other OOS missions have been performed since then. Although, these missions were successful at the time, they were carried during manned Extravehicular Activity (EVA).

Nowadays, OOS is even more relevant, since in the past, launch failures were the most common cause of failure for satellites. Although, due to increasing reliability of these launches, the on-orbit failures of satellites are now the dominant form of satellite failure [13]. It is even shown that a high-reliability approach, making use of high-reliability components and subsystems, and redundant systems, is inadequate for long term satellite platform lifetimes. Therefore, the robotic OOS is one of the keys to further development of the space infrastructure [14].

Since the servicing of satellites using manned spaceflight is more expensive and risky, autonomous concepts are considered for OOS missions. Most simple OOS missions, such as refuelling, orbit modification and perhaps simple repairs, can be performed by robotic servicing missions [12]. OOS missions are commonly performed with known spacecraft, and in addition they can be cooperative as well. It is interesting to note that the first activity during an OOS mission can be to visually inspect a target satellite, which is conveniently possible to combine with vision based relative navigation.

2.2. Spacecraft Relative Pose Estimation

This section will cover relative pose estimation methods applied in the context of spacecraft relative navigation. The adjective 'relative' is specifically used to indicate pose estimation of another object relative to a specified reference frame, usually the chasers' camera reference frame. This is opposed to own-, or ego-, pose estimation in the robotics field or Attitude Determination & Control (ADC) in the astronautical field.

The relative pose consists of the six Degrees Of Freedom (DOF) parameters, these are three translational DOF, i.e. position, and three rotational DOF, i.e. attitude. In addition, the first derivative of these parameters can be determined as well, i.e. translational and rotational velocity, to a total of 12 DOF. This section will provide some insight into proposed methods to determine the relative pose of a target in space, but first, an overview is given how different types of spacecraft relative pose estimation can be distinguished.

Types of Pose Estimation

Naturally, the approach towards automation of relative navigation is heavily influenced by the cooperative nature between the chaser and the target. In light of this cooperative nature, four different types of relative pose estimation can be identified [4]:

- Actively cooperative
- Passively cooperative
- Uncooperative known
- Uncooperative unknown

A target is considered cooperative if it is built to provide information which is useful for the pose estimation. Even then, a distinction can be made between targets that actively provide information to the chaser for instance using radio frequency communication, or those that only passively interact with the target, for instance using light patterns. It is worth noting that it is even possible for both satellites to actively cooperate and manoeuvre to perform relative navigation, that both spacecraft take on the role as both a target and a chaser.

On the other hand, uncooperative targets do not contribute to the process of relative navigation. However, a distinction can be made if any a-priori knowledge is available on the geometrical properties of the target, making the target known, such as another defunct but intact satellite, or unknown, for instance a piece of space debris or a natural object.

It should be noted that for cooperative targets no distinction is made between known and unknown appearance, since it is assumed that detailed knowledge is available for every cooperating target. Although, there is a distinction to be made for actively cooperating or passively cooperating, which in contrast, can not be made for uncooperative spacecraft.

In addition to the distinction which can be made on the cooperative nature of the target, another distinction can be made on the physical properties and docking interface of the target [15]. Physical examples of these different types if space debris are given, in decreasing orders of the physical knowledge and docking interface:

A. For instance, known targets which feature a docking interface can be dysfunctional satellites.
B. Targets with known physical properties and without a docking interface can be rocket stages.
C. Unknown targets which feature a docking interface can be foreign or damaged satellites.
D. Targets of unknown physical properties and without a docking interface can be fragmented space debris or natural satellites.

It should be noted that the availability of a docking interface does not directly affect the relative pose estimation process itself, but it can influence subsequent processes to dock or interact with the target.

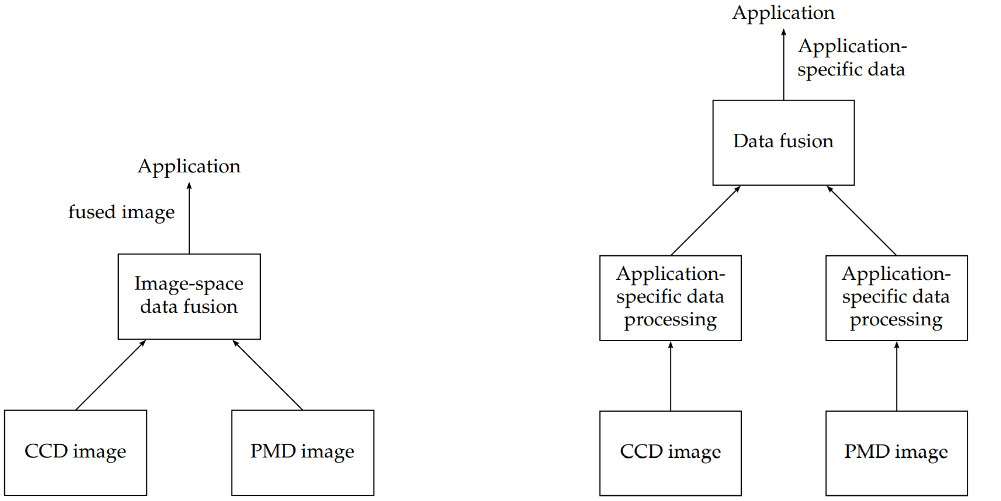
2.3. Spacecraft Relative Pose Estimation using Multisensor Data Fusion

Considering that by now the general relative pose estimation process, the sensors and feature detection methods have been covered, a closer look can be given towards multisensor data fusion. This is considered to be the literature which is most tightly related to the topic of this thesis. Even though this thesis will only focus on unknown targets, the literature for both known and unknown targets is included, since the availability of the literature on this topic is limited.

Multisensor Data Fusion Levels

Since the introduction of multisensor data fusion as an engineering disciple, many different models have been suggested to describe the process as close as possible. These process models have been described in the literature study, but further along the thesis they found no further practical use. However, a similarity and key takeaway from all these process models is the abstraction of the fusion at different levels.

The different levels range from a lower level at the image level, to a mid level at the feature level and a higher level at the application level. The image level and application level are visualised in Figure 2.3. It is the general consensus that the lower the multisensor data fusion the better [16]. However, when considering multisensor data fusion, it is important to understand the sensor modalities of sensor, which modality complements which and at what level can they be fused [17].



(a) Data fusion on the image level.

(b) Data fusion on the application level.

Figure 2.3.: Different levels of multisensor data fusion [18].

Relative Pose Estimation Methods using Multisensor Data Fusion

Several methods have been suggested in literature for multimodal relative pose estimation, which will be covered in this section. These methods apply multisensor data fusion to an end-to-end process, which is relative pose estimation, which starts from sensor input to spacecraft relative pose as output. From other research disciplines, such as robotic and machine vision, these methods are often referred to as Simultaneous Localization And Mapping (SLAM), Structure From Motion (SFM) & photogrammetry. These methods mainly rely on Bayesian methods, reducing the assumed Gaussian error of a process. In the remainder of this section, such methods for relative pose estimation using multisensor data fusion will be presented. The first two presented methods are specifically proposed for known targets whereas the latter ones for unknown targets.

In 2015 a pose estimation algorithm for uncooperative known targets is suggested by Tzschichholz et al [19]. The suggested algorithm makes use of a passive Charge-Coupled Device (CCD) sensor and an active Photonic Mixer Device (PMD) sensor, i.e. a Time-Of-Flight (TOF) Lidar. These different sensors will produce different data, namely high resolution grayscale images and range images, respectively. The authors identify that the data fusion will enable the algorithm to operate to the strengths of the different sensors. These are depicted in Table 2.1. It should be noted that the algorithm does not fuse the data at an image level, rather it fuses the data at a higher level combining partial pose information to provide the full pose, as shown in Figure 2.3b.

Table 2.1.: This table shows the reasoning of multisensor approach, using the strong capabilities of specific sensors for specific information [19].

Pose component	Best representing quantity	Sensor
Pitch angle	Object surface normal vector	PMD
Yaw angle	Object surface normal vector	PMD
Roll angle	Outer object edges	CCD
X position	Outer object edges	CCD
Y position	Outer object edges	CCD
Z position	Object surface normal vector	PMD

In 2019, an algorithm was proposed by Peng et al. that fused measurements from a Lidar sensor and stereo-vision system [20]. Even though both sensors create a 3D point cloud, the authors identify that the stereo-vision system has a higher spatial resolution compared to the Lidar sensor, while the Lidar is less susceptible to adverse illumination conditions. It is interesting to note that the authors showed a comparison in computation speeds of stereo-vision processing algorithms, for which the reader is referred to the original paper. The independent 3D points are fused together to create a full 3D point cloud. The points are fused using the RANdom SAmple Consensus (RANSAC) algorithm [21], based on correlations provided by the Sum of Absolute Difference (SAD) algorithm [22]. In the case when the 3D points provided by the stereo-vision system are inaccurate, i.e. if the target is $> 5\ m$ away from the sensors system, only the 3D points by the Lidar are processed. The pose is determined by extracting features out of the point cloud data and processing these features in an Extended Kalman Filter (EKF) algorithm using Iterative Closest Point (ICP) matching.

Capuano et al. propose an algorithm to determine the relative pose and reconstruct the model of an unknown uncooperative target [23]. It is dubbed the Simultaneous Estimation of Pose and Shape, or SEPS, algorithm. The algorithm relies on a monocular camera and a single beam Lidar measurement. The depth measurement is needed to recover the scale of the reconstructed model. The monocular camera provides the image for feature extraction and selection. These features are then used in an EKF, although, the authors mention that any nonlinear filter might be used.

A similar approach making use of a monocular camera and a single range measurement is suggested by Jin et al. [24]. The algorithm makes use of both an EKF and an Extended Kalman Particle Filter (EKPF). For feature detection, the algorithm makes use of the Harris corner detection.

The algorithm proposed by Conway et al. relies on the measurements from an Red-Green-Blue (RGB)-Depth camera [25]. The algorithm makes use of feature based SLAM methods based on both landmarks from both 2D color images and 3D point clouds. The feature detection and matching is implemented using the Oriented FAST and rotated BRIEF (ORB) descriptor [26]. It is interesting to note that the map, or model, of the target is updated at a reduced rate compared to the state estimation of the target. The algorithm is both tested numerically and experimentally, using an Microsoft Kinect as the RGB-Depth sensor.

Volpe et al. proposed an algorithm in 2017 to determine the pose of an unknown target [27]. The method relies on a monocular camera and a distance sensor. The algorithm track significant features using the Speeded Up Robust Features (SURF) algorithm. These features are combined with the targets dynamic model in an Unscented Kalman Filter (UKF). The data from the multiple sensors are fused on the feature level, 2D detected features are converted to a 3D feature on the object and combined with the 3D features from the range sensor.

Hao et al. proposed in 2015 a multimodal scale-unambiguous relative pose estimation of space uncooperative targets [9]. They proposed to fuse the visual-Lidar data based on a modified range-intensity MRF model. Using the high resolution RGB camera to increase the resolution of the depth images. The resulting 3D features serve as input for a EKF-UKF-PF navigation filter which can perform the required SLAM functionality. In a separate step, the scale is retrieved to provide the scale-unambiguous relative pose.

3. Research Approach for Multimodal Feature Detection

The previous chapters presented the background and related theory on the topic of multimodal relative pose estimation. To investigate the topic and associated research questions properly, it is proposed to develop an a multimodal feature detection method and use the detected features for relative pose estimation. Additionally, it is proposed to acquire visual-Lidar dat through both computational simulation and experiments, which is required for the algorithm development, verification and validation. To encapsulate these different elements, a general approach towards this research is carefully determined and documented in this chapter.

Since the scope of this thesis is only a piece of the GNC required for an RPO mission, the research relies on assumptions which cover the other aspects of the GNC. These assumptions will be presented in the first section of this chapter, Section 3.1. Afterwards, the requirements for the different elements of this thesis are presented in Section 3.2. While taking the earlier presented scope, assumptions and requirements into account, the research approach is discussed in Section 3.3. This includes the motivation for certain high-level design aspects, such as the selection of the filtering scheme for the pose estimation.

3.1. Assumptions

The assumptions are intimately related to the scope of this research, which was presented in Section 1.4. Additionally, as stated before, the research scope is part of larger GNC system, of which the possible influences, or contracts, are covered by the assumptions as well. If the assumptions cover all uncertainties as much as possible, the research can be carried out in a structured manner such that it is done correctly and can be reproduced.

The assumptions listed according to their respective element of the research, which are the following: the target, the relative motion, the external conditions and the data. Additionally, each assumption is accompanied by a motivation, reasoning why the assumption is necessary. Lastly, each assumption will be assigned to a unique identifier, which is denoted as **ASM-&&-#**, in which the ampersand sign (&&) indicates the element to which the assumption applies and the hash sign (#) indicates the respective number of the assumption in the associated element.

Firstly, the assumptions with respect to the target:

- **ASM-TGT-1** *The target is assumed to be a rigid satellite.*
  Satellites might contain liquid sloshing or flexing appendices, which can cause the irregular motion or features moving with respect to the target. Which can make an accurate continuous pose estimation troublesome.

- **ASM-TGT-2** *The target consists of planar features.*
  The proposed feature detection method for pose estimation relies on accurate and continuous plane detection. This is Man-made satellites often include panels due to ease of manufacturing and shape factor. For an actual application it should be considered that a panel might be damaged and therefore harder or even unable to detect.

Secondly, the assumptions with respect to the relative motion of the target:

- **ASM-MOT-1** *The trajectory of the target is only considered in close range, i.e. a distance of* < 10 *m, compared to the chaser.*
  Even though both the visual camera and the Lidar can provide useful information before the target is situated close to the chaser, this thesis only covers the close range phase of the RPO. This leads to higher resolution information on the target and allows the representation of the target motion using Euclidean kinematics, such that the Clohessy Wiltshire equations are not required.

- **ASM-MOT-2** *The satellite spins around a single axis, which passes through the center of gravity of the satellite.*
  This assumption significantly simplifies both the data acquisition and analysis, since the moment of inertia, precession and nutation do not have to be considered. This assumption is also apparent in earlier literature [9].

- **ASM-MOT-3** *The accelerations of the target are minor only.*
  By making this assumption, the research is specifically focused on the feature detection and relative pose estimation only, since other aspects of GNC are not considered. And given that the target does not behave dynamically on its own.

- **ASM-MOT-4** *The target remains in full visibility of both sensors.*
  This assumption simplifies the relative pose estimation, since the required data is more continuously available, rather than falling partially or completely outside the FOV of one of the sensors.

Thirdly, the assumptions with respect to the external conditions:

- **ASM-EC-1** *Sunlight is the only source of light considered for the visual-Lidar data acquisition.*
  In the space environment, light from the Sun is the major contribution to the illumination of the target, given that the target does not actively illuminate the target (with exception of the active Lidar). Other sources of illumination are neglected, such as albedo.

- **ASM-EC-2** *For the visual camera, effects of adverse illumination conditions are considered to be negligible.*
  Even though adverse illumination conditions play an important role in EO based spacecraft relative navigation, it assumed that these effects are negligible. Including this aspect in the research as well would not be possible given the time. In earlier literature this effect is also neglected, to focus the research on the proposed relative pose estimation method [8].

- **ASM-EC-3** *For the Lidar, effects of adverse illumination conditions are limited to an increase in depth noise.*
  Compared to the visual camera, the effect of adverse illumination conditions on Lidar are generally limited. It is assumed that effect of a lower SNR expresses as an increase in depth noise. To research this effect in detail, the current state-of-the-art technology on Lidar simulation should be available.

Lastly, the assumptions with respect to the data and its availability:

- **ASM-DAT-1** *The required visual-Lidar data is assumed to be available at each time step, or time-synchronised.*
  This simplifies the algorithm structure and the research significantly, since the effect of a real time acquisition of measurements can be neglected. Additionally, it is assumed that both the visual and the Lidar data is simultaneously available, since unimodal methods are considered out of scope.

- **ASM-DAT-2** *It is assumed that both sensors are rigidly attached to the chaser and that this transformation is known for both sensors.*
  It is important that the rigid transformation between the two sensors and the chaser is known. In this way, the measurement of the visual camera and the Lidar can be correctly related to one another and, ultimately, the target.

These assumptions will be taken into account throughout the work of this thesis. When these assumptions will be referred to in the text to follow, the assumptions will be denoted by their identifier in *italics font*, e.g. *[ASM-TGT-1]* refers to the assumption: *The target is assumed to be a rigid satellite.*

3.2. Requirements

The proposed work for this thesis consists of many different parts, which all interact with one another. In order to avoid loosing track of the goal of each different part, requirements have been established. Similar to the assumptions, requirements are established according to distinct major elements of this thesis.

Simulation

- **REQ-SIM-1** The simulation tool shall be able to simulate both the Lidar and the visual camera.
- **REQ-SIM-2** The simulation shall be able to model different resolutions and focal lengths for both the Lidar and visual camera.
- **REQ-SIM-3** The simulation shall be able to model multiple relative navigation trajectories.
- **REQ-SIM-4** The Lidar simulation shall be able to mimic depth noise for the Lidar.

Algorithm

- **REQ-ALG-1** The relative pose estimation filter shall be able to determine the location of the target in at least 1.5 times the standard deviation of the range sensor ($> 1.5 \cdot \sigma$).
- **REQ-ALG-2** The relative pose estimation filter shall determine the attitude of the target with sub degree precision in rotation ($> 1^\circ$).
- **REQ-ALG-3** The proposed methods shall not be making use of learning-based methods.
- **REQ-ALG-4** The feature detection shall be able to detect false unknown data correspondences.

Experiment

- **REQ-EXP-1** The experiment shall closely emulate favorable illumination conditions, i.e. simulated sunlight located aft with respect to the sensor direction.
- **REQ-EXP-2** The experiment shall be able to emulate different trajectories.
- **REQ-EXP-3** The experiment shall be repeatable for future researchers.
- **REQ-EXP-4** The ground truth of the target pose shall be measured throughout the experiment.
- **REQ-EXP-5** The experiment shall accommodate external calibration for the Lidar and visual camera.

3.3. Research Approach

In the previous sections the assumptions and requirements for this thesis have been established, based on the reviewed literature within the scope of the subject. Even though this provides a solid foundation for the research, the exact approach for the entire research has not yet been presented, other than the research proposal and research questions presented in the introduction. In this section, the research approach for the topic of multimodal feature detection will be discussed.

In a similar fashion to the realisation that the feature detection proved to be the knowledge gap which required the most research, as discussed in Section 1.2. As the thesis progressed, it proved that the verification and validation of computer vision methods is not that straightforward. Therefore, this section will first cover the approach for verification and validation, which will impact the approach for the research. Afterwards, some high-level design aspects will be discussed in detail, such as the algorithm architecture and the navigation filter selection.

Verification & Validation

The research on multimodal feature detection is part of the discipline of computer vision. In general, processes which make use of computer vision methods are verified and validated through empirical analysis of an end-to-end task [28]. The computer vision methods are rarely verified and validated specifically. For example, a facial recognition task is only tested for the complete end-to-end task, rather than the feature detection itself. In some cases, the feature detection is even assumed to be perfect in order to research pose estimation filter specifically [8], this example was discussed in Section 1.2.

In this work both the feature level approach and the end-to-end approach for verification and validation will be taken into account. These verification and validation methods are not general, both approaches are dependent on the application considered. The verification and validation at feature level is specifically established to accommodate the correct implementation of multimodal feature detection, which will be explained in more detail in Section 7.2.

Clearly, the end-to-end task that will be considered for this work is that of relative pose estimation. Which, in turn, can be split in four different options. Which are the following, in increasing order of credibility, but also complexity[1]:

A. Through simulation and analysis
B. Through comparison with published data and methods
C. Through lab-based experiments
D. Through in situ (space-based) experiments

Already during the literature study, it became apparent that real-world visual-Lidar data of an actual space mission was not available. Probably due to the fact that multimodal approaches are currently not considered as mature as unimodal approaches. Moreover, it even appeared that visual-Lidar data from earlier publications, those reviewed in Section 2.3, was not published next to the article itself. It makes little sense to compare methods with different data sets, simply due to the influence of a multitude of factors, for instance illumination.

The two verification and validation options remaining are both A and C, which will both be carried out in this work. Which will nudge the work towards data acquisition both through simulation and experimentation, as was already proposed in the research proposal. Finally, the lack of required data for option B is an extra motivation to publish the data used in this work.

Relative Pose Estimation Architecture

Since the verification and validation will be done by analysing the relative pose estimation, it is important to determine how this algorithm architecture will look like. The earlier presented scope, based on the GNC of an RPO mission, will serve as a starting point for this architecture. Later in this report, the currently unknown types of multimodal feature detection and navigation filter will be included. In this section the building blocks will be organised in the correct order.

First of all, in line with the research objective, the required data will be provided by data acquisition from both a visual camera and a Lidar. Subsequently, the features are extracted from the preprocessed measurements which are considered the input for the computer vision element of the algorithm. Afterwards, the features will be combined in the multisensor data fusion step, which will later be used as input for the navigation filter. The navigation filter will eventually provide the relative pose of the

[1] One can argue about the order of option B and C, although if an experiment is done correctly, this can be used and verified by other researchers as well.

target for the remainder of the GNC, outside of the scope of this thesis. This architecture is visualised in the figure below[2]:

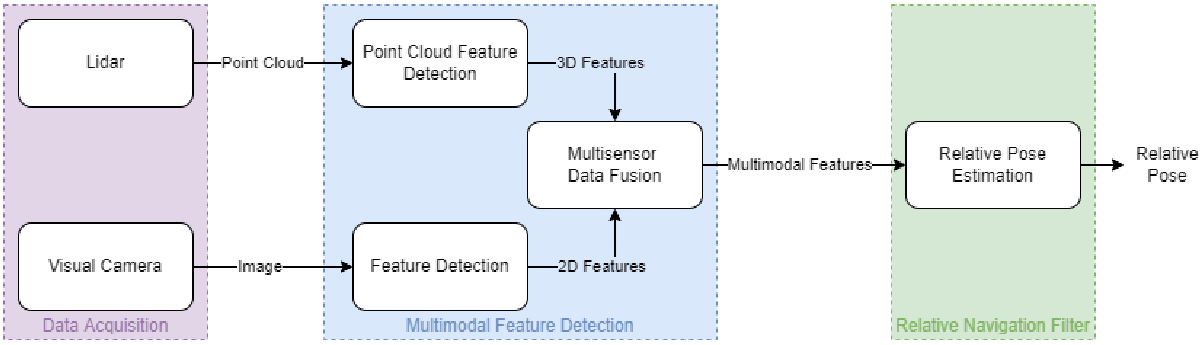


Figure 3.1.: This figure shows the general algorithm architecture, of which the building blocks will be more detailed as the thesis progresses.

Lastly, if both the 2D and planar features have been detected at exactly the same time, also referred to as time-synchronised, both features can be combined to perform multimodal feature detection. Naturally, it is not plausible that the timestamp of both data inputs align precisely, therefore it is assumed that the data from the visual camera and the Lidar are time-synchronised *[ASM-DAT-1]*. This could probably be resolved by aligning sampling times and allowing a time margin between the data inputs. Although, time-synchronising the experiment sensors and actuators was considered outside the scope of this thesis, therefore an alternative method will be suggested in Section 6.1.

Navigation Filter Selection

As required, the proposed pose estimation algorithm shall determine the pose of the target without a-priori information on the target. Therefore, the algorithm needs to relate corresponding measurements from frame to frame, enabling the algorithm to estimate the pose of the target. As hinted in the first paragraph of this section, this process can be subdivided into two distinct parts:

- Computer vision for feature detection
- Navigation Filter for pose estimation

The computer vision will detect 3D feature locations which serve as an input for the navigation filter. The navigation filter is used to combine discontinuous data for estimation and correction of both the pose and feature locations of the target. The problem requires the estimation of the following vector:

$$\chi = \left[\bar{s}_{1:t}\ ,\ \mu_{1,x}\ ,\ \mu_{1,y}\ ,\ \mu_{1,z}\ ,\ \dots\ ,\ \mu_{M,x}\ ,\ \mu_{M,y}\ ,\ \mu_{M,z}\right]^T \tag{3.1}$$

In which $\bar{s}_{1:t}$ represents the target pose for time step 1 to t. A single pose vector is expanded below in Equation 3.2. The variables $\mu_{1,x}, \mu_{1,y}, \mu_{1,z}$ represent the feature locations in the target reference frame from 1 to M.

$$\bar{s} = [\underbrace{x, y, z}_{Position}, \overbrace{\dot{x}, \dot{y}, \dot{z}}^{Velocity}, \underbrace{\phi, \theta, \psi}_{Rotation}, \overbrace{\omega_x, \omega_y, \omega_z}^{AngularVelocity}\ ]^T \tag{3.2}$$

[2] Next to the tasks of multimodal feature detection and relative pose estimation, there should be accurate bookkeeping of all relevant variables, particularly the ground truth during the data.

As is apparent from the equations, the problem requires the estimation of a high dimensional vector, a vector of size $12 + 3 \times M$ to be exact. Only for low numbers of M, this would be feasible to estimate given the computational requirements of such a high dimensional estimation problem. Since the correspondences of the features on the target are unknown, it is highly unlikely that this estimation problem would reduce to a low dimensional estimation problem. Therefore, the FastSLAM algorithm was selected for the filter.

The FastSLAM algorithm is a particle filter for navigation combined with an EKF for feature positions which can reduce the computational load to $M\, log(M)$ compared to an M^2 for the case of a regular EKF as navigation filter. The main idea of the particle filter is to use a set of samples to represent the state space, the state space is a distribution of all the possible states. Each sample is an hypothesis for both the pose and the feature location, these samples are referred to as the particles. In addition, each of these particles is assigned a weight, according to the correspondence of the hypothesis to the measurements. Using the weight of these particles, a distribution is formed to represent an overall hypothesis of the pose. In addition to the reduction in complexity, this type of representation can model arbitrary, even multimodal[3], distributions. Which is, for example, the case for an EKF which is limited to a Gaussian distribution.

[3] Please be advised, there are two definitions of multimodal. This definition portrays a multi-peaked probability density function.

4. Multimodal Feature Detection

In this chapter the research on multimodal feature detection is presented. In Section 2.3 we saw that a MRF was used for the depth estimation of features. Three different methods were considered for multimodal feature detection, of which the last one provided the main contribution to the state-of-the-art. Namely, performing multimodal fusion at feature level: 3D feature detection by combining 3D plane detection and 2D feature detection.

The required input data and preprocessing for all methods will be covered in Section 4.1. Then, the proposed 3D feature detection method is proposed in detail in Section 4.2. Afterwards, the feature matching and registration will be covered in Section 4.3. Finally, the two rejected methods will be introduced and and the rejection will be substantiated in Section 4.4.

4.1. Input Data Preprocessing

As with any computer vision algorithm, the input data must comply with the expected data format specified for the algorithm. This allows for increased data encapsulation for all processes involved with the computer vision algorithm. In this section, a closer look is taken on the data formats and how the data is preprocessed for further use. As apparent from the algorithm architecture, the measurement of both the 2D and 3D data are considered to be available at the start of the algorithm loop.

First, the preprocessing of the 2D visual data is covered, which is similar for both simulated and recorded data. Which is followed by the preprocessing of the 3D Lidar data, which shows some differences for simulated and recorded data. It should be noted that in principle, the following preprocessing methods are applied for all multimodal feature detection methods presented in this chapter, including the rejected methods discussed later in this chapter.

Preprocessing Visual Camera Data

In both the case of simulated and recorded data, the visual camera data is a series of timestamped images which have sampled the visual spectrum of light. If the spectrum was sampled at multiple wavelengths of the visible region, for instance in full color Red-Green-Blue (RGB), the visual camera data needs converted to a grayscale or monochrome format. This is done since most computer vision algorithms require this type of input format.

First however, it is important to understand the details of the image coordinate system used in MATLAB. By default, an image in MATLAB is stored in an array, which is a fundamental MATLAB object. The array can be either two-dimensional or three-dimensional, for monochrome images or color images respectively. Each index from a row and a column corresponds to a pixel in the image, rows from top to bottom and columns from left to right, as visualised in Figure 4.1. The entries at each index indicate the intensity of that color, for instance three values for RGB.

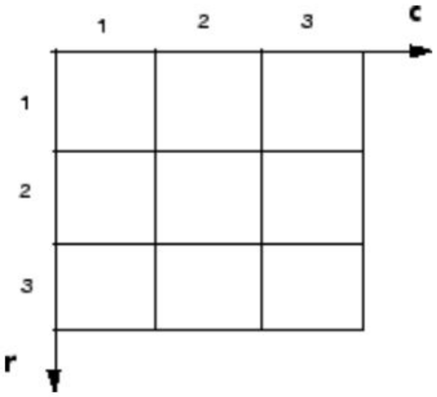


Figure 4.1.: Image coordinate system in MATLAB.

As suggested before, in the case of an RGB image, it should be converted to a grayscale format for further use. This is done by calculating the luminance with the use of the following MATLAB function: `rgb2gray`. This function takes the three Red-Green-Blue values as input for the weighted sum which describes the luminance according to ITU standards [29].

$$E'_y = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B \tag{4.1}$$

Alternatively, only a single channel of the full RGB image can be used to create a monochrome image. This is done by using one of stored channels in the first, second or third index of the pixel location [30]. This is, however, not often applied in computer vision.

Preprocessing Lidar Data

Similar to visual camera data, the Lidar data requires preprocessing as well. As suggested in Section 3.1, it is assumed that solid state Lidar recordings are available. The recordings by the Lidar in the experiment require some more thought before they can be used in the feature detection algorithm, this will be covered in more detail in Section 6.5. However, the following preprocessing steps have to be carried out for both simulated and experimental data.

First of all, the origin of the Lidar point cloud is transformed to the origin of the visual camera, except in the case of a hybrid colinear sensor. If the Lidar and visual camera have exactly the same orientation, a translation would suffice, otherwise a translation and rotation is required. This is done in order to correctly relate the 2D data with the 3D data, in the multisensor data fusion step, as described in the next section. In a simulation environment, these transformation parameters are easily determined, whereas in an real-world application, these have to be determined through extrinsic calibration of both sensors, as covered in more detail in Section 6.5.

Secondly, from experience with both simulated and experimental data, it appears that each point cloud contains invalid points. This can be for instance an infinite coordinate or a point close to the origin $(X: 0, Y: 0, Z: 0)$. This is caused by the default settings for an wrongly measured or undetected point set by the sensor. These invalid points may impact the computations on the point cloud in a negative way. Therefore, these points are removed with the use of: `removeInvalidPoints`.

Lastly, there are two optional preprocessing methods which might be beneficial, if applied correctly. If more information is available on the Lidar noise and, most importantly, if it is consistent, the points might be denoised using `pcdenoise`. However, this is not required in the case of plane fitting, since it will fit the best plane regardless of the noise. Furthermore, if processes are too slow due to a large dense point cloud, it might prove useful to downsample the pointcloud using `pcdownsample`. This decreases the number of points in a point cloud, especially useful for close range Lidar measurements.

4.2. 3D Feature Detection

In this section, the developed method for 3D feature detection by combining 3D plane detection and 2D feature detection will be explained in detail. As suggested, this method has proven to be useful for relative pose estimation with an unknown and uncooperative target. First, the methods for 2D feature detection will be discussed, followed by the 3D plane detection afterwards. Then, both the 2D features and the 3D planes are fused to determine the 3D location of the feature. Briefly, this multisensor data fusion step at feature level is done by projecting the 2D features on the 3D plane. Lastly, the feature tracking, rejection and registration will be covered in more detail. This is considered to be prudent for accurate pose estimation of the target, which requires clean data association.

2D Feature Detection

As described before, most feature extraction methods process grayscale images to establish feature descriptors and subsequently detect feature locations. Accordingly, the measured images are converted to a grayscale format as a data preprocessing step as presented in Section 4.1. Various feature detection methods are available from literature, two of which are SIFT and SURF. These methods were considered as they are invariant to both scale and rotation, particularly useful for the considered motion of the target. Of the two methods, SIFT is described as the most stable method [31]. However, it was found that the SIFT method requires significant parameter tuning compared to the SURF method, in addition, the SURF method is significantly faster compared to SIFT. For these reasons, the SURF method was selected for feature extraction.

The SURF method was executed using the `detectSURFFeatures` function from the computer vision toolbox from MATLAB. It performs SURF feature extraction on a grayscale image and returns a `SURFPoints` object which describe the feature. The `SURFPoints` object is used as input for the `extractFeatures` to determine the location and descriptor of valid SURF features. The detected feature location is a floating point number in pixel coordinates $[u, v]^T$, used by the navigation filter as described in Section 14. The extracted feature descriptor is a vector of length 64 which is used for feature tracking, as described in Section 4.3.

In addition to the SURF feature descriptor, the Harris corner detection method was used. This method was specifically applied for the verification of the 3D feature detection method, described in this section. The Harris corner detection method was chosen due to the fact that it is more deterministic to indicate a corner, rather than a region, as is the case with the SURF method. This allowed for the direct comparison of a corner on a target and a feature detected by the proposed method, this is covered in more detail in Section 7.2.

Plane Detection Using RANSAC

Recall that the proposed method requires a 3D plane to project 2D features on, such that the 3D location of the feature can be determined. According to the assumption that the target includes planar regions, these planar regions can be measured by the Lidar and subsequently determined using computer vision methods. In addition, the point cloud which describe the measurements from the Lidar are considered to be unorganized. A plane is fitted by using the function `pcfitplane` which makes use of the M-estimator SAmple Consensus (MSAC) algorithm, a variation on the RANdom SAmple Consensus (RANSAC) algorithm.

Described in words, this method iterates over the points in the point cloud to check if the points match a plane hypothesis, the plane hypothesis with the highest score is selected to be a plane, consisting of the points that are considered inliers. In addition to the indices of these inliers, the function returns the planar equation which describes the plane:

$$a \cdot x + b \cdot y + c \cdot y + d = 0 \tag{4.2}$$

Algorithm 4.1: MLESAC algorithm [32]

```
$i \leftarrow 0$
$bestFit \leftarrow 0$
$bestPlane \leftarrow [0,\ 0,\ 0,\ 0]$
for $i = 1 : M$ do
    $tempInliers \leftarrow selectRandom(data,\ n)$
    $tempPlane \leftarrow fitPlane(data,\ tempInliers)$
    $score \leftarrow getResiduals(data,\ plane)$
    if $score > bestFit$ then
        $bestFit \leftarrow score$
        $bestPlane \leftarrow tempPlane$
    end
    $i \leftarrow i + 1$
end
return $bestPlane$
```

Additionally, this type of plane description directly gives the normal vector of the plane, namely $[a, b, c]^T$. It is more than likely that the target will present more than one planar feature in the FOV, therefore the above process is iterated over 5 different initial estimates for the planes normal to find a maximum of 5 planes. It was established that 5 different planes would be sufficient for the relatively simple target under consideration, namely the Delfi-n3Xt.

Feature Position Estimation

As described in Section 3.3 the multisensor data fusion step takes place at feature level, right after the detection of both the 2D features and 3D planes, from the visual camera and Lidar measurements respectfully. As suggested briefly in the introduction of this section, this is done by projecting the 2D feature on the 3D plane. This is visualised simply in the figure below:

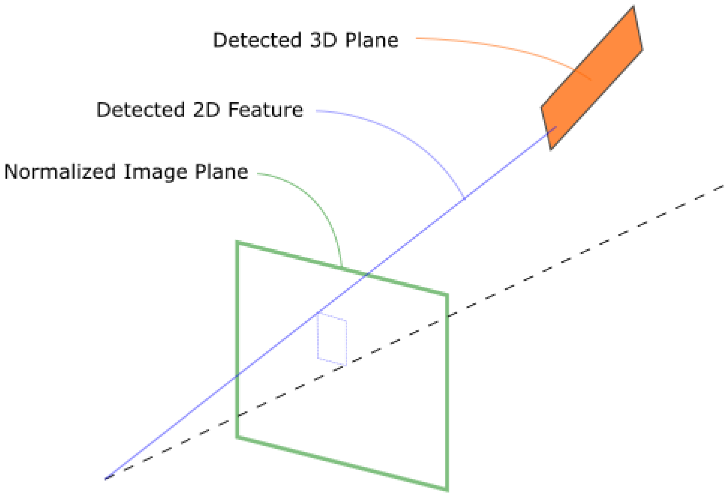


Figure 4.2.: Simple visualisation of an intersection of a 2D feature with a 3D plane.

However, before the position of the feature can be determined, several steps are required. As a first step, it should be determined which 2D feature is located on which 3D plane. To achieve this, it is analysed which feature is located within the boundary of the plane. This boundary is determined on a set of 2D points, which is the projection of the point cloud on the 2D image plane. The second step is to find the 3D feature location by determining the intersection of the feature ray and the associated plane. This can be done since the intrinsic parameters of the visual camera are known and the 3D plane was determined in the reference frame of the visual camera. Therefore, the location of the feature can be found by determining the intersection of the 3D plane and a 3D line. These steps will be explained further in the remainder of this section.

So firstly, the points which are considered inliers of a 3D plane, are projected on the 2D image plane, which means that each 3D point is projected as a 2D pixel coordinate. This projection can be achieved by using the image equation based on the pinhole camera model with central projection, which is shown below:

$$\begin{bmatrix} u \\ v \end{bmatrix} = K \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.3}$$

In which the pixel coordinates are as u, v and K is referred to as the camera intrinsics matrix, which consist of the focal lengths f_x, f_y for both the x- and y-direction and the principal point offset x_0, y_0. The experienced reader will notice that the skewness parameter s is missing, this is due to the fact that the images have been undistorted as part of the preprocessing steps. The inversion of this equation is usually not possible, since there is a loss of information when projecting a 3D point on a 2D image, unless information on the depth is available.

Afterwards, the boundary of the projected points is determined using the `boundary` function from MATLAB. Which is subsequently used as an input to determine which 2D feature lies within the boundary, using the `inpolygon` function. This allows 2D feature to be associated to the correct 3D plane. Moreover, this ensures that only features are selected which are actually located on the target and not the background, which is demonstrated in Section 7.2.

Now that it has been determined which feature lies on which plane, the location of the feature is determined by finding the intersection of the feature ray, which is colinear with the normalised image coordinate. To determine the normalised image coordinates, essentially the inverse of Equation 4.3 is calculated, realising that the Z-value will be equal to 1. This is determined using the following equation.

$$p_{norm} = K^{-1} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (u - x_0)/f_x \\ (v - y_0)/f_y \end{bmatrix} \tag{4.4}$$

Finally, the 3D feature position is provided by the intersection of the normalized image coordinate and the associated plane. This is determined using the following equations.

$$\begin{bmatrix} x_n^C \\ y_n^C \\ z_n^C \end{bmatrix} = \begin{bmatrix} x_n^i \cdot \frac{-d}{a \cdot x_n^i + b \cdot y_n^i + c} \\ y_n^i \cdot \frac{-d}{a \cdot x_n^i + b \cdot y_n^i + c} \\ \frac{-d}{a \cdot x_n^i + b \cdot y_n^i + c} \end{bmatrix} \tag{4.5}$$

The above method is repeated for each 2D feature which is detected within a 3D plane, resulting in the 3D coordinates of all the detected features which are within the boundary of the detected planes. These 3D coordinates, as shown below, will serve as an input to the pose estimation filter.

$$\bar{y} = \begin{bmatrix} x_1 & y_1 & z_1 & \dots & x_n & y_n & z_n \end{bmatrix}^T \tag{4.6}$$

4.3. Feature Matching and Registration

In the sections above, the feature detection method has been presented which take place during each frame. Although, for every pose estimation filter to work correctly, correct data associations should be made between frames. This data association allows for tracking of specific features, such that these features can be matched to a pose and feature posterior. To achieve this, first the features from the previous frame are matched to the features of the current frame. This is done using the `matchFeatures` function and makes feature associations based on the feature descriptor.

However, not all features which are found matching are actually a match, since some feature descriptors describe a non-unique feature, such as a corner of a featureless planar surface, an example of this is shown in Figure 4.3. Therefore, it is checked whether the feature 3D measurement does not exceed the expected location by too much. This rejection is based on the distance update the EKF suggests, which is explained in more detail in Section 5.2.

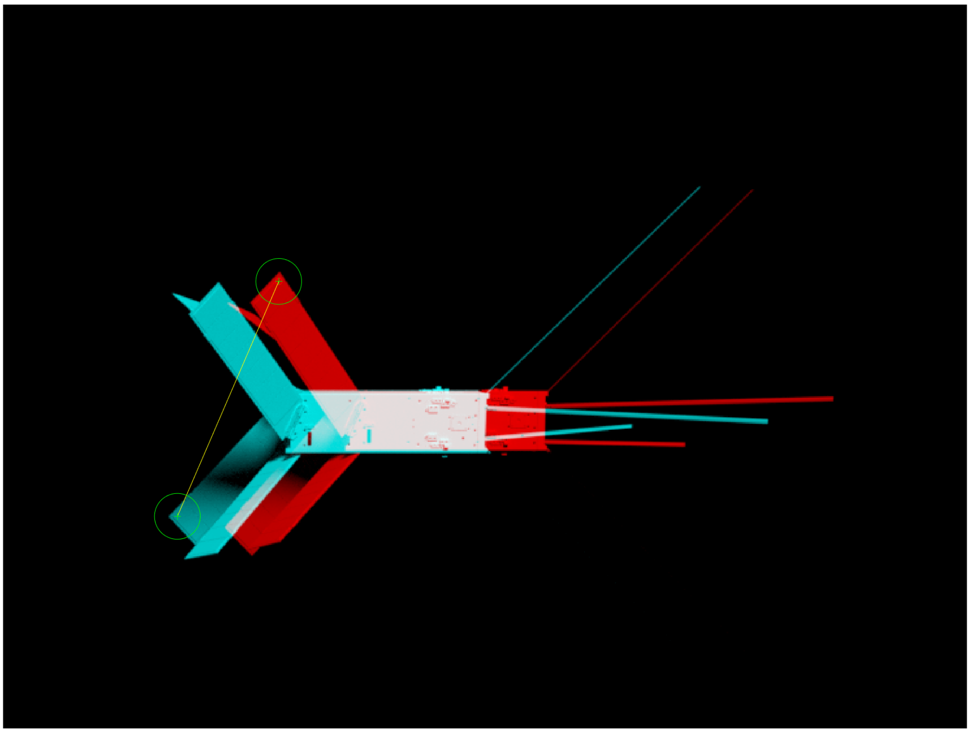


Figure 4.3.: A rejected feature in a false color image of 2 separate frames.

Since multiple features are detected at each time step, with some being matched and some being rejected, a concise bookkeeping system for the features should be adopted. Even though each particle in the FastSLAM algorithm has its own feature location posterior, this feature registration done as much as possible independently of the particles to reduce the computational load. This can be done since it is recognised that the particle only stores a 3D feature location and the feature detection and matching is independent of the particle posterior. All that remains to be done for each particle is the feature rejection scheme which is applied in each EKF update.

Thus, recall that in each frame features are detected[1]. The features are newly found features are either matching or nonmatching with the previous features, while the matching features are either rejected or accepted by the EKF update. Only the accepted features are used for the particle weighting, which is explained in more detail in Section 5.3. The newly detected features are added to the register, whereas the rejected feature updates are removed from the register and added as newly detected features. This feature registration process is visualised in Figure 4.4.

[1] Or no features are detected, in which case the filter propagates to the next frame.

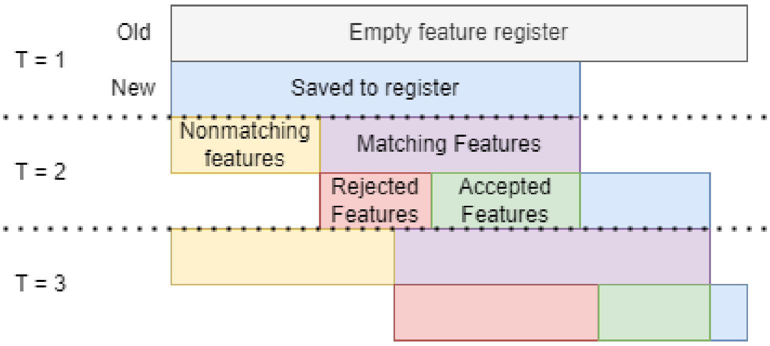


Figure 4.4.: A visualisation of the feature registration process.

4.4. Rejected Multimodal Feature Detection Methods

As is apparent from the method described above, for correct multisensor data fusion, information from both modalities have to be extracted correctly and related to one another. Recall that the goal of this thesis was to investigate if there were other multimodal methods useful for relative pose estimation, although, not all attempted methods proved to be promising or reliable. Next to the successful method described above, two other multimodal feature detection methods were considered. However, both of which were rejected due to the ineffective unimodal process, already before fusion. Even though these methods were not yet found to be promising, it was interesting to pursue them, which finally lead to the 3D feature detection method described earlier in this chapter.

Other attempts for plane detection

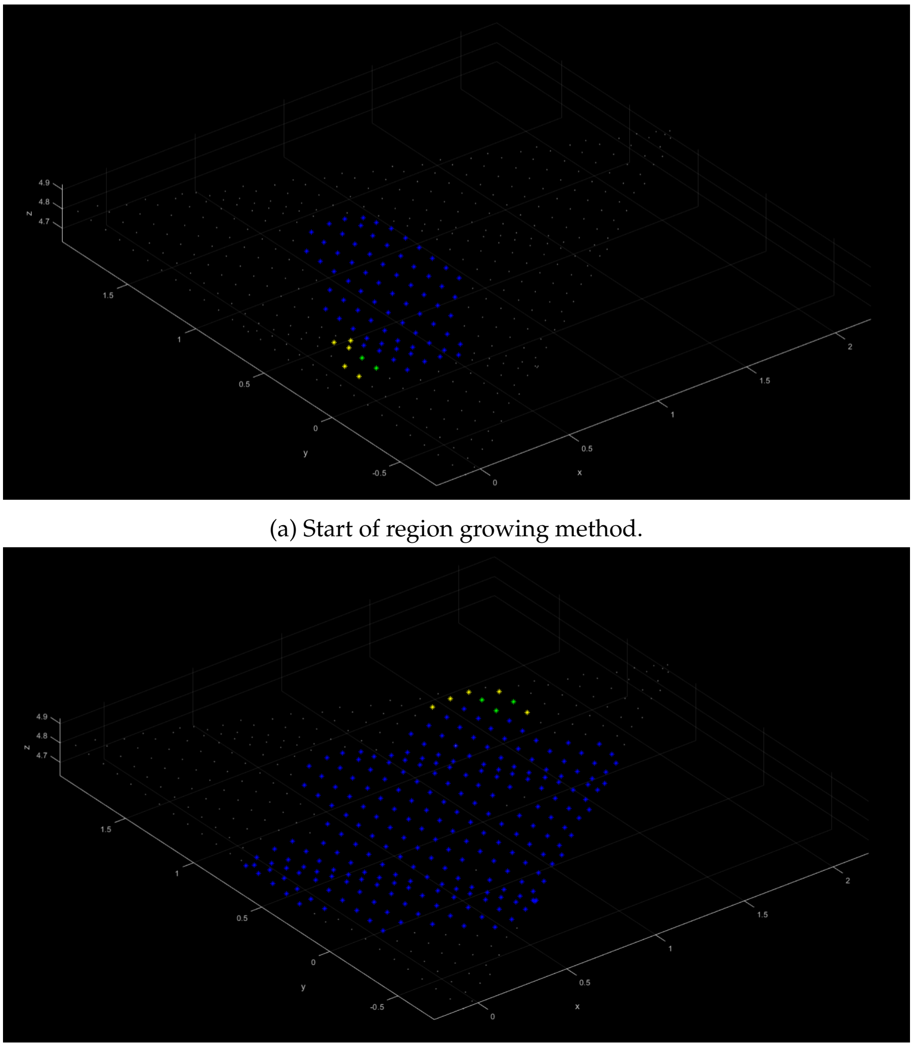
In order to detect planar surfaces and particularly planar edges more accurately, an attempt was made to detect planar regions trough a different method. The method made use of a region growing method based on RANSAC to determine both the plane and the edges, introduced as Depth Image-based Plane Detection (DIPD) [33]. The fusion step would enable the refining of the detected 3D edge, through fitting the 3D edge with a higher resolution 2D edge using Canny edge detection. Below you can see 2 images showing the region growing process:

Eventually, this method was considered to be too costly in terms of computation time to be useful for spacecraft relative navigation. This was due to the fact that multiple hypotheses needed to be fitted through the region growing process for each frame. In addition each region growing process required a computationally expensive nearest neighbour search for large three-dimensional data. Therefore, this method was discontinued even before the fusion step was implemented.

Edge-based detection

In the second pursued method the edge detection strategy from above was approached from the visual data perspective. The previously mentioned Canny edge detection was used to find edges on the visual image of the target. Subsequently, the intersections of these edges would be used to refine the corners found using the Harris corner detection method. Afterwards, the fusion of the planar equation would provide the depth to both the edges and corners [34]. Finally, the edges and corners would be related to one another using graphing theory.

Although, some practical problems were encountered with the Canny edge detection method. In some cases, the target was orientated such that the light scattered evenly on two different surfaces, this caused that the local gradient of the image was to low to detect an edge. In addition, the method proved to be largely unstable when analysing more complex targets with multiple or smaller planar regions. Lastly, there were two other challenges encountered with the rejected methods. Namely that in existing

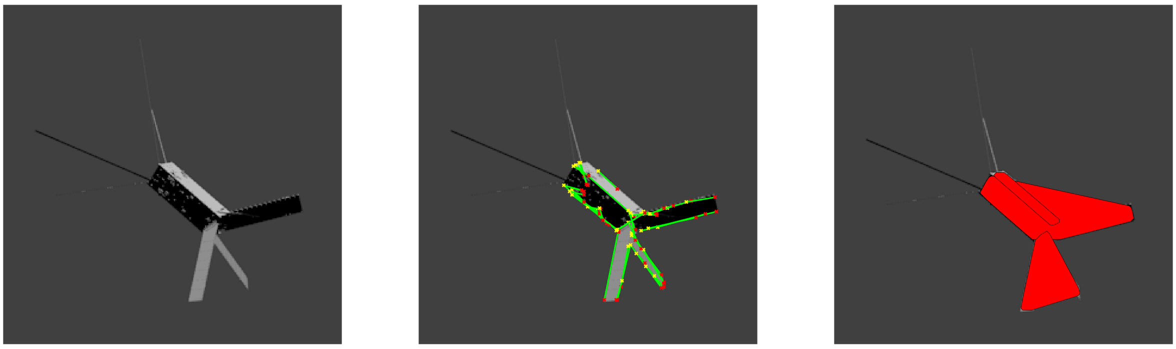


(a) Start of region growing method.

(b) Further progressed region growing method.

Figure 4.5.: Two figures showing the region growing progress at different stages. The blue points are those who are fitted on the plane, the green points indicate a newly detected point in the plane and the yellow points indicate the new candidates for the region growing method.

literature no method has been found that can relate the detected edges to the pose of an unknown uncooperative target. Furthermore, the dependence on edges might result in ambiguous results for symmetric targets. For the reasons stated above, this method was discontinued as well.



(a) The original image.

(b) The detected Canny edges.

(c) The detected planes.

Figure 4.6.: Demonstration of Canny edge detection on Delfi-n3Xt, shown step by step.

5. Spacecraft Relative Pose Estimation using FastSLAM

As suggested in Chapter 3, feature detection methods are often verified by evaluating their performance through end-to-end computer vision processes. In the scope of this particular thesis, the end-to-end process is the relative pose estimation process of an uncooperative unknown target. Additionally, this implementation allows for research on this aspect of spacecraft relative navigation as well. In this chapter, the implementation and changes required for a relative pose estimation method using FastSLAM will be presented. Most notably, the observation equations from earlier literature using FastSLAM for relative pose estimation required adjustments to operate using the new 3D feature detection method.

Firstly, a detailed overview of the FastSLAM algorithm will be provided in Section 5.1. This is followed by the implementation of the feature location EKF in Section 5.2. Then, the weight update step and filter resampling step will be described in Section 5.3 and Section 5.4, respectively. Finally, the initialisation of the filter will be described in Section 5.5. Most of the methods covered in this chapter are closely related to the works from Augenstein and Rock [35] and Hao et al. [9], since in both articles, the FastSLAM algorithm is suggested for relative pose estimation.

5.1. FastSLAM Overview

As discussed in the introduction, the eventual goal of the relative pose estimation filter is to provide accurate relative pose estimation of an uncooperative unknown target. The general algorithm architecture was already displayed in Figure 3.1, which can be expanded now that the multimodal feature detection method has been made clear in the previous chapter, Chapter 4. This more detailed algorithm architecture is shown in Figure 5.1. The implementation of the relative navigation filter will be based specifically on FastSLAM while using the 3D features as input.

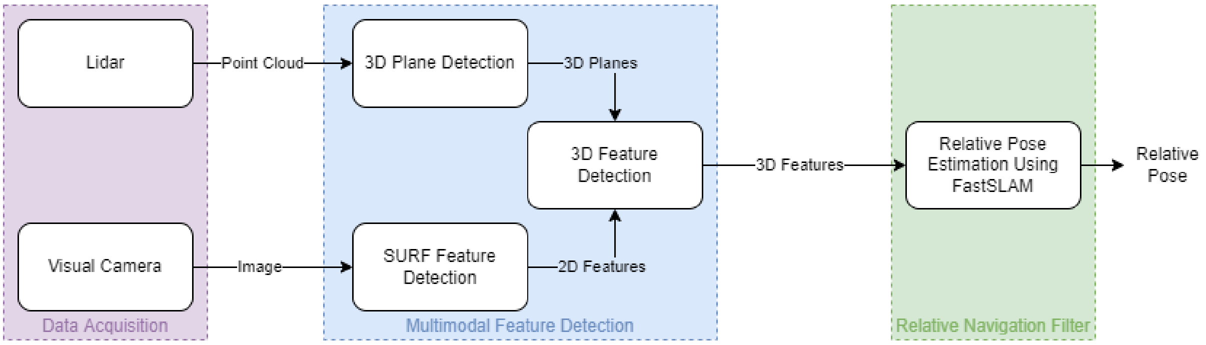


Figure 5.1.: Detailed overview of the filter architecture.

Based on the architecture displayed above, it is clear that the FastSLAM algorithm will process the 3D features from the multimodal feature detection, resulting in the pose of the target. It should be noted that only the matching features are used as input for the filter, as the features without a match can not be updated from frame to frame. Thus, the matched features are the input for the FastSLAM algorithm.

The relevant assumptions for this part of the following: *[ASM-MOT-2]*, *[ASM-MOT-3]* & *[ASM-DAT-1]*. Which result in a target which spins around a single axis with minor accelerations and with data available at each time step.

Considering the required input and the relevant assumptions, the FastSLAM algorithm can be covered in more detail. In this thesis, FastSLAM is the algorithm of choice to estimate the relative motion of the target, as suggested in Section 3.3. Recall that the FastSLAM employs a particle filter for the relative pose and an EKF for the feature positions. The particle filter represents the relative pose as a distribution with N particles, with assigned weights to each particle.

Finally, there are two distinct outputs of the FastSLAM algorithm, which is the relative pose of the target in the reference frame of the chaser and the feature positions in the reference frame of the body. Both outputs are determined at each frame, where the relative pose will likely change over time and the feature positions will remain largely constant. The output of the relative pose is given by the weighted average of the particles and the output of the feature positions is determined by the EKF. These combined outputs are presented mathematically in the following equation:

$$\chi = \left(\bar{s}_{0:t}, \mu_{1,x}, \mu_{1,y}, \mu_{1,z}, \ldots, \mu_{M,x}, \mu_{M,y}, \mu_{M,z}\right)^T \tag{5.1}$$

In which the first entry $(\bar{s}_{1:t})$ represents the 12 DOF relative pose, as shown in Equation 3.2, over time from 0 to T. The remainder of the entries represent the location $(\mu_x, \mu_y$ & $\mu_y)$ of the 3D features in the reference frame of the target. The general structure of this FastSLAM implementation is depicted in Algorithm 5.1, shown below:

Algorithm 5.1: FastSLAM Algorithm for Spacecraft Relative Pose Estimation using 3D Features

```
$\bar{s}_0 \leftarrow$ initial state estimate            //Initialise Algorithm as shown in Section 5.5
$\sigma^2_{\bar{s}_0} \leftarrow$ initial noise estimate
$w_0^{s[j]} \leftarrow 1/N$
$\mu_0 \leftarrow$ map initial features                   //Using measurement model (Equation 5.3)
for $t = 0 : T$ do
    //Loop over the measurements
    $\bar{s}_t \leftarrow$ propagate particles
    for $i = 1 : N$ do
        //Loop over the particles
        for $j = 1 : M$ do
            //Loop over the matched features
            $\mu_j \leftarrow$ feature location update        //Feature EKF (Algorithm 5.2)
            $w_j^{[i]} \leftarrow$ feature weighting          //Feature weighting (Equation 5.5)
        end
        $w^{[i]} \leftarrow$ particle weighting
    end
    $\bar{s}_t \leftarrow$ resample                           //Resampling step (Algorithm 5.3)
end
```

As can be seen in Algorithm 5.1 above, several processes are still to be clarified. For instance, for each feature for each particle there will be a feature state update, updating the estimated location of the feature in the reference frame of the target. But first, the particle state update will be covered, which represents the motion of the target.

state update equations

The state update equations are provided by the relatively simple forward propagation of the current state. This is possible, due to the Rao Blackwellisation for the posterior of the feature locations and the relative pose. Every particle is an estimate where it is assumed that the pose is correct, therefore the mapping step is trivial. Therefore, the feature state estimates do not affect the particle state propagation.

$$\bar{s}_{t+1} = \bar{s}_t + \Delta t \cdot \frac{\partial \bar{s}_t}{\partial t} \tag{5.2}$$

The propagation shown in Equation 5.2 is repeated for each particle. However, the initial state estimate is wrong, and without resampling of the state estimate, the particle feature would not be able to converge to the true value. The particles are weighed according to the feature state update, making use of an EKF, as is shown in the next section.

5.2. Extended Kalman Filter for Feature Location

Since the pose estimation is performed without a-priori knowledge on the target, both the feature locations and the feature correspondences are unknown. The feature locations are distributed across the target and should be determined by the pose estimation filter as well. Whereas the feature correspondences are partly determined by the feature matching as explained in Section 4.3, the feature rejection is done in the EKF update. This section will cover the feature location update and associated feature rejection.

Measurement Model

As the theory of the FastSLAM suggests, the feature location in the reference frame of the body is considered to be decoupled from the estimated pose. However, the 3D features locations are measured in the camera reference frame. Therefore a measurement model should related the measurements to the feature position in the different reference frames. There are two options available for the measurement model, one with respect to the camera reference frame and one with respect to the chasers reference frame. However, the first option can be used for two reasons. First, the connection between the sensors and the chaser is assumed to be rigid and the transformation is known. Second, the work in this thesis only concerns with the relative pose estimation, thus no ego-pose estimation or uncertainties are treated, which usually is done in the reference frame of the chaser.

So, the measurements of the detected 3D features are dependent on the target relative pose in the reference frame of the camera and the feature location in the reference frame of the target. This results in the following measurement model $g(state, featurelocation)$, displayed for an arbitrary detected feature:

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= g(state, feature\ location) + noise = g(\bar{s}, \mu) + r^2 \\ &= R_T^C \cdot \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \cdot \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T + \begin{bmatrix} X_s & Y_s & Z_s \end{bmatrix}^T + \mathcal{N}(0, r^2) \end{aligned} \tag{5.3}$$

In which R_T^C indicates the rotation matrix from the chaser to the target, the subscripts i and s indicate the feature location and target, respectfully. The symbol r represents the measurement noise of the 3D feature detection method.

Measurement Update

The measurement update of the EKF changes the estimate of the feature state in the reference frame of the target. This is done according to the mapping step of FastSLAM and the measurement model shown in Equation 5.3. The measurement model is the center of the EKF for this FastSLAM implementation. The implementation of the EKF is depicted in Algorithm 5.2, shown below:

Algorithm 5.2: Extended Kalman Filter [9]

```
$\bar{s}_{t|t-1} \leftarrow g(\bar{s}_{t|t-1}, \mu_{t-1})$    //Measurement model given particle pose t-1 (Equation 5.3)
$G_s \leftarrow$ Jacobian measurement model    //Equation 5.4
$Q_{j,t} \leftarrow G_s \Sigma^i_{j,t} (G_s)^{\mathrm{T}} + r^2 I$
$K_{j,t} \leftarrow \Sigma^i_{j,t|t-1} (G_s)^{\mathrm{T}} Q^{-1}_{j,t}$
$\mu^i_{j,t} \leftarrow \mu^i_{j,t|t-1} + K_{j,t} \left(\bar{s}_{t|t} - \bar{s}_{t|t-1}\right)$
$\Sigma^i_{j,t} \leftarrow (I - K_{j,t} G_s) \Sigma^i_{j,t|t-1}$
return $\mu^i_{j,t}, \Sigma^i_{j,t}, Q_{j,t}$
```

The Jacobian of the measurement model should be clarified in more detail, it is relatively straightforward and therefore the derivation is not treated in most literature. This is shown in Equation 5.4, which can be seen below:

$$\begin{aligned} G_s &= \frac{\partial g(\bar{s}, \mu)}{\partial \bar{s}_i} \\ &= \frac{\partial R^C_T \cdot \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T \cdot \begin{bmatrix}X_i & Y_i & Z_i\end{bmatrix}^T + \begin{bmatrix}X_s & Y_s & Z_s\end{bmatrix}^T}{\partial \bar{s}_i} \\ &= \frac{\partial R^C_T}{\partial \bar{s}_i} \end{aligned} \tag{5.4}$$

In words, this is the partial derivative of the rotation matrix from the camera reference frame to the target reference frame, with respect to the current attitude angles.

Feature Rejection

As explained in Section 4.3, the EKF is only executed for features which are detected in the current and the previous time step. However, these correspondences between the matched feature are not always correct, as demonstrated in Figure 4.3. Therefore, a feature is rejected if the state update for that particular feature, is considered to be too large.

Currently too large is ambiguous, fortunately, the EKF computes the (co)variance of the feature positions. In line with the Gaussian distribution related to the EKF, the distance is set to everything within ± 2σ. Such that 95 % of a Gaussian distribution is considered to be an inlier. The distance is currently being computed by the cityblock distance, which could be improved with a spherical distance or Mahalanobis distance. It should be noted that it for this particular part of the filter that the depth information provides more information compared to relative navigation filter based on 2D features only.

5.3. Weight update

As each particle estimate is propagated throughout time and the feature locations are updated, it is clear that some particles diverge from the true state. Therefore the particles are weighted to indicate their likelihood that the particle is matching the measurement. To determine the importance weights of each of the particles three distinct steps are required.

Firstly, each feature state estimate is weighted according to their correspondence to the measurements. This is done using the Gaussian likelihood method, as it is done in the FastSLAM algorithm. The weight of a single particular feature state estimate can be determined using the following equation:

$$w_j^{[i]} = \frac{1}{\sqrt{2\pi \cdot |Q|}} \, e^{-\frac{1}{2}\left(\bar{z}_j - \hat{z}_j^{[i]}\right)^T Q^{-1}\left(\bar{z}_j - \hat{z}_j^{[i]}\right)} \tag{5.5}$$

This is done for each feature which has been both matched and accepted, as explained in Section 4.3 and Section 5.2. Afterwards, the number of accepted features is determined for each particle. The particles that have at least 3 less accepted features compared to the particle with the most accepted features is weighed in the following step.

$$\mu_{acc}^{[j]} > max\left(\mu_{acc}^{[j]}\right) - 3 \tag{5.6}$$

In which μ_{acc} indicates the number of accepted features for particle i. This is done to remove any particles with severely diverging estimates. Next, the particles which satisfy the above requirement are assigned an importance weight using the following equation:

$$w^{[i]} = \prod_j w_j^{[i]} \tag{5.7}$$

All particles which did not satisfy the requirement in Equation 5.6, are assigned a weight of 0. Lastly, all the weights of the particles are normalised using the following equation:

$$w^{[i]} = \frac{w^{[i]}}{\sum_{i=1}^{N} w^{[i]}} \tag{5.8}$$

It should be noted that the weight of different number of features, can lead to a difference in importance weight of several orders of magnitude.

5.4. Filter Resampling

As the filter progresses, particles which show to diverge are resampled based on particles which provide a better estimate. The resampling is performed based on the distribution of importance weights, which has been described in the section above, Section 5.3. This allows the filter to concentrate the particles to the state space with higher probability.

Not all particles are resampled, the particles which were assigned a weight which exceed the number of particles divided by 20 ($w^{[i]} > N/20$) will persist throughout the resampling step and will be used to form the distribution for resampling. This means that the particles which did not satisfy Equation 5.6 will always be resampled since their weight is set to 0. Additionally, particles are not resampled at every time step, rather it is dependent on the amount of particles which provide a good estimate compared to those that do not. This is done using the following equation:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}\left(w^{[i]}\right)^2} \tag{5.9}$$

In which N_{eff} represents the number of particles which have sufficient weight, which is set to $0.5 \cdot N$. When this condition is met, the particles which satisfy the conditions above are resampled using low variance resampling, as described in Algorithm 5.3.

Algorithm 5.3: Low Variance Resampling [36]

```
$\bar{s}_t \leftarrow \emptyset$                                   //Empty state space/distribution
$r \leftarrow rand(0, M^{-1}$
$c \leftarrow w_t^{[1]}$
$i \leftarrow 1$
for $m = 1 : M$ do
    $u \leftarrow r + (m-1) \cdot M^{-1}$
    while $u > c$ do
        $i \leftarrow i + 1$
        $c \leftarrow c + w_t^{[i]}$
    end
    add $\bar{s}_t^{[i]}$ to $\bar{s}_t$
end
return $\bar{s}_t$
```

Finally, after the resampling step, all the particle weights are reset to $1/N$ and the filter can be propagated to the next time step.

5.5. Filter Initialisation

Now that all the distinct elements of the FastSLAM filter have been explained, all that remains is to cover the initialisation of all these different elements. The elements which require initialisation are the following: the initial particle states, the feature locations and their associated noises, the particle noise and the feature location noise, respectively.

First and foremost, the particles itself will be initialised by generating N different particles with equal importance weight of $1/N$. The estimated pose of the particle will be drawn from a Gaussian distribution $N(\bar{s}_0, \sigma_{\bar{s}_0}^2)$, in which $\bar{s}_0$ is the initial pose estimate and $\sigma_{\bar{s}_0}^2$ is the initial particle process noise. Exactly like the estimated pose, the initial pose estimate consists of the 12 DOFs as depicted in Equation 3.2.

Since the initial particles will be drawn from a Gaussian distribution, the initial estimate will be approximately equal to the mean ($\bar{s}_0$), given that sufficient particles are initialised. As indicated before, the state estimate consists of the position (a), the translational velocity (b), attitude (c) and rotational velocity (d). It is assumed that the mid-range rendezvous phase will provide a decent quality first estimate for the relative pose, this will serve as a starting point such that the estimation converges properly. The initial mean of these elements are initialised as stated below, based on the previously assumed available data:

(a) The initial estimate for the position of the target is an equally weighted average of the points in the first point cloud. Given a roughly equal distribution of the density of the target, this estimate will be around the center of mass, with a bias towards the chaser, since the points are located in between the target and the chaser.

(b) Whereas the translational velocity is assumed to be known to lie within the standard deviation of the particle noise of the translational velocity, which will be explained in further detail in the list below.

(c) Since no target model is available at all, the initial estimate for the attitude of the target is ambiguous. Since there is no reference frame available to compare it to. Therefore, all attitude angles will be initialised at the ground truth of the target motion. This will be demonstrated in further detail in Section 7.3.

(d) With respect to the rotational velocity, the initial estimate is assumed to be within the standard deviation of the true rotational velocity as well. The standard deviation will be shown in the list below.

Next to the mean of the particle state, the noise ($\sigma^2_{\bar{s}_0}$) has to be initialised as well. The initial estimates for the noise have been determined through a sensitivity analysis, resulting in the following initialisation approach for each of the standard deviation ($\sigma_{\bar{s}_0}$):

(a) The standard deviation of the position is initialised as 0.1 times the smallest cross section of the target.

(b) The standard deviation of the velocity is less than 5 centimeter per second.

(c) For the attitude the standard deviation is less than 0.1 degrees, a small value is suited since the initial state estimate for the attitude is ambiguous.

(d) The standard deviation for the rotational velocity is initialised at most 0.5 degrees per second.

With respect to the initial feature state estimate, the initialisation is done according to the principle of FastSLAM: the feature location is assumed to be exactly corresponding to the measurement assuming a correct pose. Therefore, the combination of the particle's pose and the first measurement of each feature will directly result in the initial feature state estimate, as indicated in Equation 5.3.

Whereas the feature measurement covariance is initialised according to the following equation:

$$\sum_{j,0}^{[i]} = r^2 \cdot G_\mu \tag{5.10}$$

In which G indicates the covariance matrix of the feature state estimation, as presented in Algorithm 5.2, and r indicates the measurement noise, which is primarily dominated by the measurement noise introduces by the Lidar, which is shown in Section 7.2.

6. Visual-Lidar Data Acquisition

As part of the research, visual-Lidar data is required to enable multisensor data fusion. In this thesis, the required visual-Lidar data finds its origin from two different sources. The first source is from computational simulation, which is mainly used to develop and analyse the developed algorithm. The second source is from experiments, which finds its primary use to be for verification and validation of the proposed methods. This chapter will give an overview of the acquisition of the visual-Lidar data from these different sources and form a clear understanding of their features.

Firstly, an overview of the visual-Lidar data acquisition is provided in Section 6.1. Next to some aspects of data acquisition in general, it discusses the coupling between data acquisition and verification and validation. Secondly, the implementation of the computation simulation in Blender is presented in 6.2. Afterwards, relative motion by the Panda robotic arm, the sensors and illumination will be discussed in Section 6.3 and 6.4, respectively. Lastly, the actual planning and execution of the different scenarios in the experiment will be presented in Section 6.5. At the end of the thesis, the acquired data will be made publicly available through Github, adding a visual-Lidar data set for use by the community.

6.1. Data Acquisition Overview

The multimodal approach requires a sequence of images and point clouds as input for feature detection and subsequent pose estimation, as suggested in Section 4.1. This data was used to perform the research to the topic discussed in this thesis. Although, the required data can find its origin in different sources, considering a variety of computational tools, different kind of experiments or different sensors in general. This section will provide a clear overview of the objectives of this data and its associated sources. Afterwards, some general aspects which apply to both simulation and experiments will be covered, such as the considered targets, the relative motion, the sensors and the illumination conditions.

Data Acquisition Objectives

Already in Section 3.3 the observation was made that for this particular research the only source of data originates from this thesis itself, due to the fact that no visual-Lidar data of spacecraft is publicly available. For this reason, the origin of the visual-Lidar data is coupled to the different verification and validation methods, both analysis through simulation and verification and validation using data from lab-based experiments.

The first source is through computational simulation. Several different tools were available to simulate either a visual camera or a Lidar. However, it is required *[REQ-SIM-1]* that the simulation tool can simulate both a visual camera and a Lidar. This resulted in the choice for Blender which is often used for visual camera simulation. In combination with the Blensor plugin, which enabled the simulation of a Lidar sensor [37]. The implementation of this source of data acquisition is covered in more detail in Section 6.2.

With respect to the experiments, it was required that the experiment should represent the real-world scenario as closely as possible. Such that a data set can be provided which is realistic, accurate and repeatable. This data set will mainly be used for verification and validation through the analysis of an end-to-end process. The successful creation of this data set appeared not to be straightforward, due to the many different available sensors and the required relative motion.

Over the course of the thesis, three different experiments were performed with the ultimate goal to create a realistic data set. The first couple of these experiments were not specifically necessary for validation of the algorithm, but rather to test the sensors suitability for use in space and to generate data sets to aid development. In turn, each of these experiments did provide useful insights into spacecraft relative navigation and sensor performance.

The first experiment, designated as the 'Cyber Zoo Experiment' was performed in the Cyber Zoo at the faculty of Aerospace at Delft University of Technology. For this experiment the relative motion was achieved by making use of a turntable and a trolley, which proved to be too irregular for accurate research. In addition, it was concluded that the Intel Realsense L515 did not provide the required performance for sunlit measurements. As a result from this experiment, it was decided to procure both an industrial visual camera and a Lidar, applied in both automotive and aerospace research. Furthermore, it was proposed to use a Panda robotic arm to carry out the required relative motion of the target.

The second experiment, designated as the 'Office Experiment', was performed in an office at the faculty of Aerospace. The goal of the experiment was to gain insight into the correct functioning of the newly acquired Livox Mid-70 Lidar. Therefore, no motion was required to perform this experiment. In addition, it was confirmed if the Lidar provided the promised outdoor performance. This was proved by fitting a plane on point cloud by measuring the sunlit floor outside the window office, which closely matched reality. Furthermore, a Lidar recording was made with the Delfi-n3Xt as a target to provide a point cloud to test plane detection on a space target.

The final experiment, designated as the '3ME Experiment', was the culmination of the preceding experiments. It made use of the newly acquired sensors and the Panda robotic arm to carry out the relative motion. Compared to the first two experiments, it was the most complex in nature and produced the most notable results, therefore this experiment and its setup will be described in more detail in the last three sections of this chapter, Section 6.3 - 6.5.

To summarize, each of the different sources of data served a different purpose, which aided in the research to multimodal spacecraft relative navigation. An overview of the different sources of the data sets and their respective objectives is provided in the table below:

Table 6.1.: The different sources of data to research multimodal feature detection.

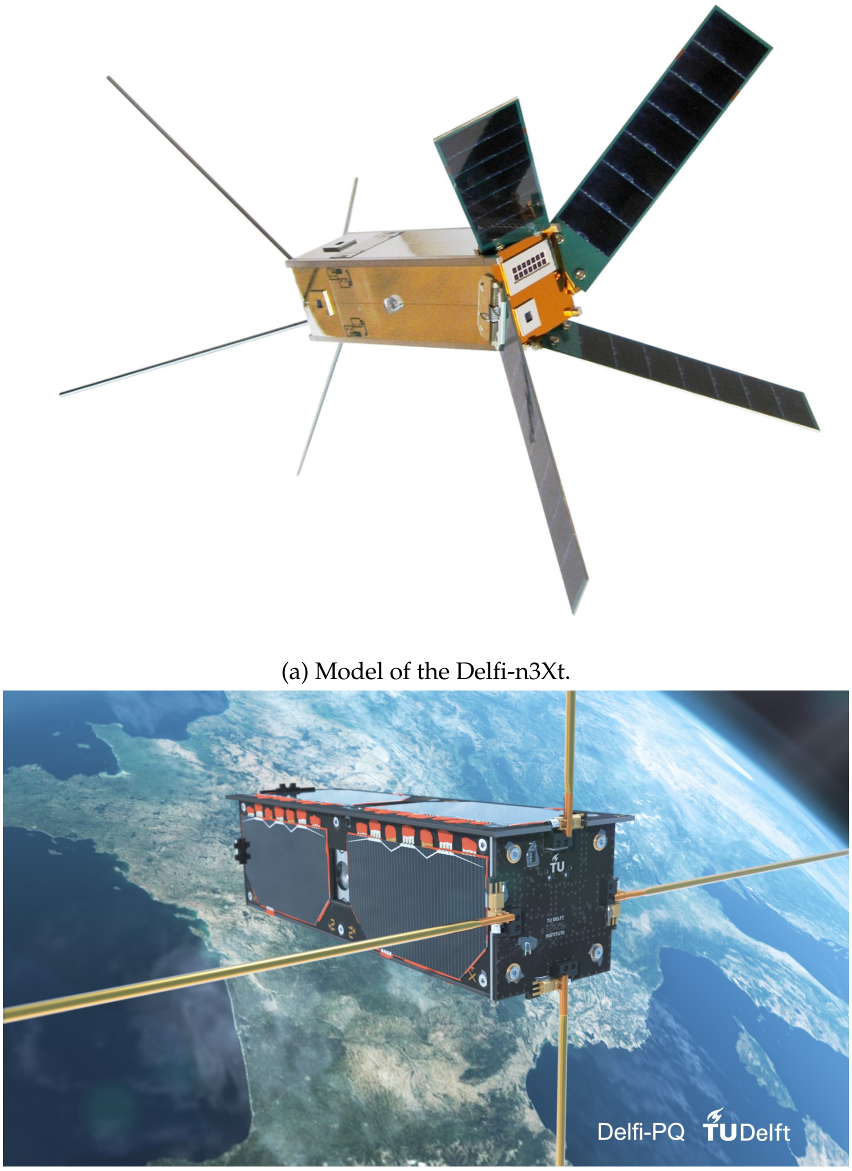
Activity	Date	Objective
Blender Simulation	Continuous	Algorithm Development & Analysis
Cyber Zoo Experiment	30/09/2021	Suitability Intel L515 Hybrid Sensor
Office Experiment	13/12/2021	Suitability Outdoor Performance Livox Mid 70
3ME Experiment	21/04/2022	Create Data Set Pose Estimation

Nevertheless, the source of the data does not determine the scenario which describes the details of the relative navigation, such as the target, relative motion, sensors and the illumination conditions. Therefore, to create a better understanding on these aspects, an overview of the different possible scenarios is provided in the remainder of this section.

Targets

Ideally, a relative pose estimation algorithm would be independent of the target, such as size or features. In this way, it would be able to deal with any unknown target. However, the multimodal feature detection method proposed in this thesis relies on accurate plane detection. Therefore, the availability of planar features is the single most important requirement for all considered targets with this proposed method. In addition, the target should be a realistic representation of a satellite, being either considered as (partially) intact or debris.

For the reasons stated above, the Delfi-n3Xt and the Delfi-PQ are considered to be satisfactory targets. Both these targets present clear planar features, given they are mostly intact. Furthermore, for both satellites the 1:1 models are available at the faculty for experiments. It should be noted that specifically a 1:1 model is required for these experiments, since the 3D measurements are directly related to the 2D features. This is opposed to approaches which rely solely on a single camera, which can make use of down-scaled models due to the scale issue. Both satellite models can be seen in the figure, Figure 6.1, below:



(a) Model of the Delfi-n3Xt.

(b) Model of the Delfi-PQ.

Figure 6.1.: Models of both the Delfi-n3Xt and the Delfi-PQ, not shown to scale[1].

Next to the realistic targets introduced above, several development targets have been used during the development. These targets are considered to be simple with clear visual features, i.e. low poly meshes with checkerboard patterns. Examples are the cuboid model for feature detection using Harris corners, shown in Figure 6.3, and the checkerboard for camera calibration, shown in Figure 6.3.

[1] Retrieved from: https://www.tudelft.nl/lr/delfi-space/, accessed at: 05-10-2022.

Relative Motion

As discussed earlier, there is a wide variety of possible scenarios for relative navigation, even when considering only the different relative motions between the chaser and the target. It is assumed however, that the target makes a rectilinear motion with respect to the chaser and rotates around a single axis *[ASM-MOT-2]*.

One can imagine that there are no relative motion constraints in a simulated environment, if the motion is described properly, e.g. no gimbal lock. In contrast, the lab-based experiments are obviously constrained to both the lab dimensions and equipment characteristics. Even though large scale outdoor experiments were considered for this thesis, it was not regarded as feasible during this MSc thesis.

In the early phases of the experiments, the motion of the target was provided with respect to rotation only by turntable, while the rectilinear motion was provided by rolling the sensors forward and backward on a trolley. As the experiments progressed, a robotic arm became available which allowed simultaneous rectilinear motion and rotation of the target. Moreover, the robotic arm provided the capability to make the experiments repeatable and accurate.

Even though the setup described above can accommodate the simplistic relative motion which is assumed, there are other factors to take into account. For instance the effect of a relative motion on the measurement by the sensors, especially considering the rolling shutter or the integration time for visual cameras and Lidars respectively. When a motion is carried out, this will obviously affect the measurements and therefore the performance of the proposed method. Next to that, it is even assumed that solid-state Lidar point clouds are available for the feature detection method.

Therefore, a method to circumvent this problem with respect to the relative motion of the experiment was established. The method is to use a stop-motion animation rather than a continuous motion, which was inspired by animation by film makers to animate clay or miniature sets. This stop-motion can be used with set intervals, greater than the integration time of the Lidar, to capture a fully integrated recording of the target frame by frame.

Next to the benefit that scanning Lidars and rolling shutter cameras can be used for the experiment, there are other benefits associated to this approach. For instance, this approach would remove the difficult task of time synchronisation across the equipment used during the experiment. Since the motion steps can be incremented slowly and subsequently counted rather than a fully time synchronised analysis. Furthermore, by analysing the different frames separately, the experiment data can be used to verify the 3D feature detection method more specifically, rather than to use it only for the entire pose estimation.

Sensors & Illumination

As expected, the type of sensors considered for the data acquisition are visual cameras and Lidars. A wide variety of both sensors is available for data acquisition through both simulation and experimentation. For the simulation, these can be modelled freely with the use of Blender in combination with the Blensor plugin, which will be covered in more detail in the following section.

For the experiments, the physical sensors are required, therefore the options are quite limited. Even though a great range of visual cameras are readily available and the availability of COTS Lidars is increasing as well. Furthermore, sensor systems which combine visual cameras with depth estimation, so-called hybrid sensors, are currently being developed and made available on the market as well.

Throughout the research, two different sensor suites were considered for the data acquisition. Of which the first was a hybrid sensor, the Intel Realsense L515. The second sensor suite consisted of the Livox Mid-70 Lidar and the Daheng MER2 visual camera, which will both be covered in more detail in Section 6.4. Even though the Lidar is a scanning Lidar with an required integration time for a dense point cloud. Due to the proposed stop-motion method, a scanning Lidar is also a viable option, regardless of the integration time. It is, however, important that the resulting point cloud is correctly introduced as an instantaneous point cloud to the feature detection algorithm.

As described earlier in the relevant literature, the data availability and quality is intimately related to the illumination conditions during the measurements. This is mainly caused by the decreased SNR for both the visual camera and the Lidar. This can not only be caused by direct illumination of the aperture, but through reflections on the Earth and satellite as well. However, for this particular research, it was assumed that the illumination conditions are in near-ideal conditions *[ASM-EC-2 & 3]*. This can for instance be during optimal illumination conditions when the Sun is located aft, compared to the direction of the sensors.

6.2. Blender & Blensor Implementation

As mentioned earlier in the overview, Section 6.1, the tool for computational simulation was chosen to be Blender using the Blensor plugin. This allows for both analysis and development of the proposed method in this thesis, both through direct feature detection method and through an end-to-end task. Moreover, data acquisition through computational simulation allows for the creation of data sets in quick succession. Even though these aspects make it seem ideal for algorithm development and analysis, the credibility of the data should be taken into account. Which was the reason to append this method with experiment which is described in the remainder of this chapter.

The Blender tool is a scriptable Python API, this allows the automation of computational simulation and the creation of plugins. For this reason, a wide variety of scenarios can be simulated with practical ease, but only if the scripts used to automate the simulation are implemented correctly. Aspects of these simulation scenarios will be covered in this section. First the implementation of objects and their respective movement will be discussed. Afterwards the settings and characteristics of the visual camera and Lidar simulation.

Scenario Implementation

The scene which describes the scenario consists of several objects. In general the objects will consist of the target, the sensors and the illumination source. The target will typically be a model of the Delfi-n3Xt, of which the mesh model was acquired through the faculty of Aerospace at Delft University of Technology. The sensors will be two `Camera Objects` which will either represent the visual camera or the Lidar. The illumination source will be a `Light Object`, specifically a `Sun Light` object. An example of these different objects in a Blender scene can be seen in the figure below:

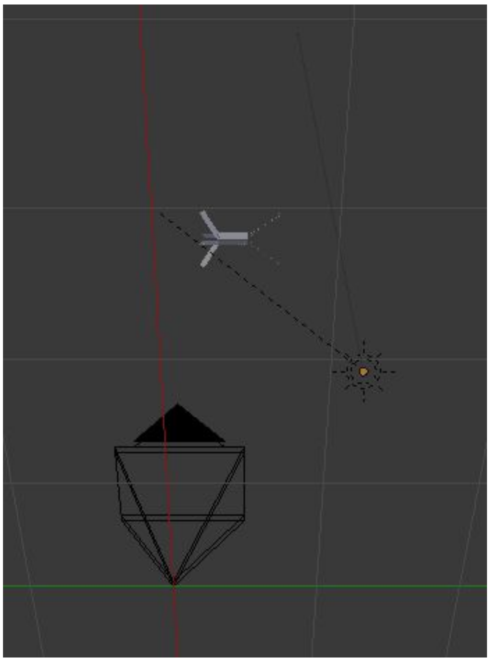


Figure 6.2.: A view of the scene in the Blender viewport. The Delfi-n3Xt model can be seen in the top, along with the Sun `Light Object` on the right and the two overlapping `Camera Objects` in the bottom.

As hinted before, the target model will typically be a mesh of the Delfi-n3Xt satellite. However, the mesh of the Delfi-PQ is also used, along with the development models. The development models is usually a clear geometric primitive such as a cube or a the smallest symmetrical shape, i.e. a tetrahedron. These are presented in the figure below:

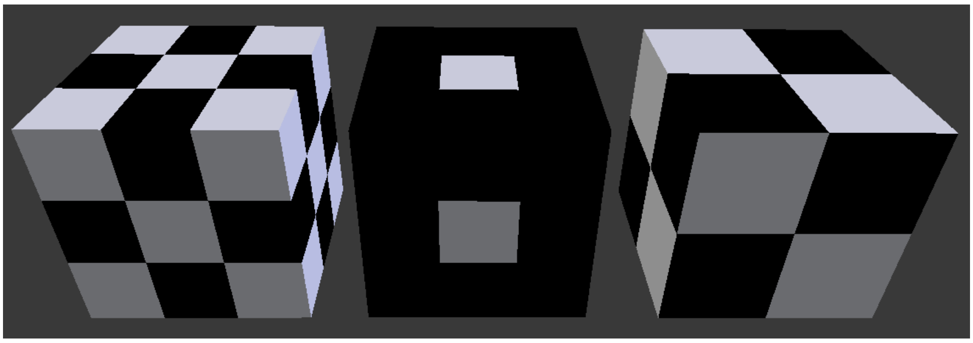


Figure 6.3.: Meshes of the development targets in the viewport of Blender.

With respect to the relative motion, it is assumed that the relative motion can be described by Cartesian movements *[ASM-MOT-1]*. In addition, no actuation of the chaser is considered *[ASM-MOT-3]*. To accommodate this, the target will move with respect to the sensor suite, of which the camera will be the origin of the reference frame of the camera. Therefore, the only motion in the scene is that of the target, which consists of both the position and orientation over time.

The properties, such as location and rotation, are adjusted throughout the animation with use of drivers. Usually, Blender controls the drivers by making use of the native keyframe system to describe a motion. The keyframes indicate the 'key' frames which are interpolated to describe the pose at each frame. However, since the motion of the target is fully deterministic, the position and orientation of the target are set explicitly on each frame. The pose for each frame is associated to a timestamp through the motion framerate, which is associated to the framerate of the individual sensors. The motion framerate is equal to the least common multiple of the sensor framerates, e.g. sensor A: 4 Hz, sensor B : 6 Hz which results in a motion framerate of 12 Hz. Once the motion framerate is known, the position and orientation is determined by forward integration of the translational and rotational velocities, starting from the initial pose.

It was stated earlier that the reference frame of the visual camera, is considered to be the origin of the relative pose estimation. This is not the case in the simulation tool however, in which the Lidar is located at the origin of the world coordinate system. This is due to the fact that otherwise a wrong point cloud is produced by the Blensor tool. The visual camera is only separated by a small baseline from the origin of the world coordinate system. Therefore, the ground truth of the pose estimation is equal to the world location of the target minus the baseline[2], provided the orientation of the sensor aligns correctly.

As stated in the assumptions, the target is illuminated by a single source, which is the Sun *[ASM-EC-1]*. Even though physically the Sun has a specific location, the location of the Sun can not be expressed in the coordinate system of Blender. However, this is not a problem since the only relevant parameter of the Sun is its direction with respect to the sensor, which was previously determined to be aft of the sensor.

Visual Camera Simulation

Visual camera simulation is on of the main native features of Blender. From the perspective of a `Camera Object`, a scene is captured while taking the objects and illumination into account. In addition to the position and orientation attributes discussed above, several other attributes of the `Camera Object` can be specified. The most important attributes are those related to the camera intrinsic parameters, such as the focal length, the resolution and the pixel size. In addition, what is special about computational simulation is that the lens type can be specified, the options are the following: perspective, orthographic or panoramic. The first one was selected, since a perspective camera represents a real-world camera.

For the visual data acquisition in Blender, the scene is rendered at each frame which aligns with the specified framerate of the visual sensor. The rendering is done using the cycles render engine, which is a physically based render engine. It makes use of backwards ray tracing, which means that the ray originates from the camera and is only registered if it encounters a light source. This is done according to the intrinsic parameters of the visual camera, which define the pixel origin and lens parameters. Repeating this procedure for each frame, produces a visual recording of the specified scene.

Lidar Simulation

Finally, to complete the full visual-Lidar data acquisition by computational simulation, the Lidar simulation remains to be discussed. As previously stated, the Lidar simulation is done using the Blensor plugin [37]. This plugin sees its origin from the active research field of robotics and autonomous driving. It aimed to provide a solid alternative to the physical sensor, such that extraordinary scenarios, such as crashes, can be modelled correctly. In addition, it allows for simple regeneration of data sets by other researchers to promote reproducible research.

The addition of the plugin to the Blender scene allowed for visual-Lidar data acquisition in a single computational tool. Although, there were some changes required for Blensor to be used for this particular research. This is mainly due to the fact that the correct sensor type was not yet available in the current release of Blensor. The available sensors were mostly representing physical Lidars which were not available for this research, such as the Ibeo LUX. Therefore as part of this work a solid state Lidar sensor type has been added, which was primarily based on the Time-Of-Flight (TOF) Lidar.

It should be noted that the Lidar simulation using the Blensor plugin is not considered to be physically based. It employs ray-tracing to determine the first encounter with a mesh. After the intersection is registered, the range of the ray is determined and subsequently altered according to the specified depth

[2] Which is part of the preprocessing steps as discussed in Section 4.1.

noise. The depth noise is modelled by a Gaussian noise: $\mathcal{N}(\mu, \sigma^2)$. In which μ is the bias and σ is the standard deviation[3]. Oftentimes, the depth characteristics of physical Lidars are described using the same Gaussian distribution. But is important to realise that other physical aspect are not considered, such as mixed-pixel errors and refraction.

6.3. Panda Robotic Arm

Over the course of the experiments, a Franka Emika Panda robotic arm became available, hereinafter referred to as the Panda. The use of the Panda enabled the consistent and accurate execution of target motions, moreover when prepared adequately the experiments can be performed in rapid succession. For the reasons mentioned above, the experiment using the Panda proved to be more valuable compared to the experiments which made use of the turntable and trolley.

The Panda is a 7 DOF robotic arm mainly used for research purposes. The two ends of the Panda are referred to as the base and the end effector, of which the latter can be a hand which holds the target. In the remainder of this document, it is assumed the target is held rigidly by the robotic arm, therefore the motion of the target is directly coupled to the motion of the end effector.

Panda Characteristics

The motion of the Panda is achieved by actuation of 7 distinct rotational joints. The independent rotation of the joints allow the Panda to move with respect to the Panda base. These rotational joints can be seen in Figure 6.4.

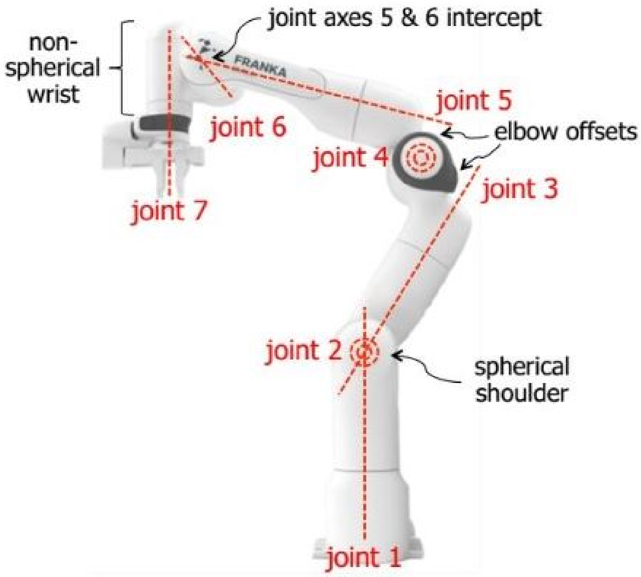


Figure 6.4.: The Panda rotational joints and reference frame.[4]

[3] It is important to note that the original author described σ as the variance, in both the article and the plugin, not the standard deviation.

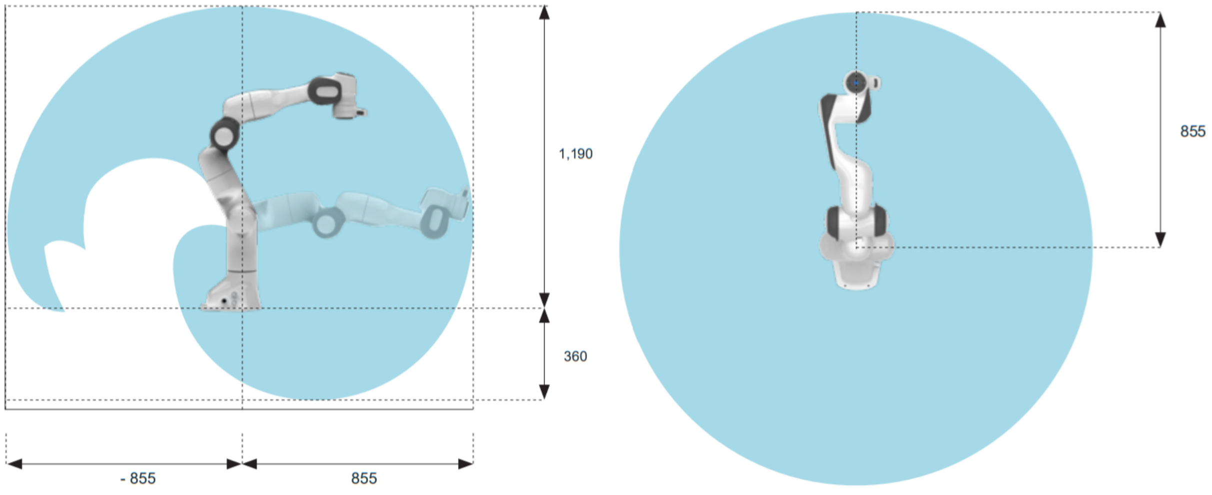
[4] Retrieved from: `https://www.chegg.com/homework-help/questions-and-answers/panda-franka-emika-shown-belowis-innovative-lightweight-robot-intended-friendly-andsafe-hu-q35002486`, accessed at: 07-09-2022.

As stated above, the motion of the Panda is limited by joints which are only able to rotate with respect to their parent[5]. Thus the motion of the panda is limited to the actuation of each of the joints, in terms of rotation, velocity, acceleration and torque. These actuation limits of the distinct joints are tabulated below:

Table 6.2.: The joint actuation limits for the Panda robotic arm[6].

Name	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Unit
q_{max}	2.8973	1.7628	2.8973	-0.0698	2.8973	3.7525	2.8973	rad
q_{min}	-2.8973	-1.7628	-2.8973	-3.0718	-2.8973	-0.0175	-2.8973	rad
$\dot{q}_{max}$	2.175	2.175	2.175	2.175	2.61	2.61	2.61	rad/s
$\ddot{q}_{max}$	15	7.5	10	12.5	15	20	20	rad/s^2
q_{max}	7500	3750	5000	6250	7500	10000	10000	rad/s^3
τ_{max}	87	87	87	87	12	12	12	Nm
$\dot{\tau}_{max}$	1000	1000	1000	1000	1000	1000	1000	Nm/s

Recognising that the Panda is the only actuator during the experiment and considering the joint limits limit the motion of the Panda, it is apparent that the motion of the target is limited to the motion of the robotic arm. In terms of the maximum motion range, the following figures depict the 2D limitations of the Panda end effector with respect to the Panda base.



(a) Side view of the motion range.

(b) Top view of motion range.

Figure 6.5.: Motion range of the Franka Emika Panda robotic arm, all dimensions are in mm.

Next to the motion range of the Panda as shown above, the limitations of the joints lead to the overall limitations of the motion of the end effector of the Panda. Namely the following[7]:

- Cartesian velocity limit of up to 2 m/s
- Pose repeatability of $< \pm 0.1\ mm$
- Path deviation of $< \pm 1.25\ mm$

Finally, there is an important characteristic to note, is the maximum payload mass of the Panda, which is a maximum of 3 kg. Apparently, the 1:1 model of the Delfi-n3Xt was over the maximum mass ($> 3\ kg$).

[5] The parent of a joint is the directly attached limb towards the base. e.g. the parent of joint 1 is the Panda base and the parent of joint 2 is the limb between joint 1 and 2.

[6] Retrieved from:https://frankaemika.github.io/docs/control_parameters.html, accessed at: 08-02-2022.

[7] Retrieved from the spec sheet, avaailable at: https://pkj-robotics.dk/wp-content/uploads/2020/09/Franka-Emika_Brochure_EN_April20_PKJ.pdf, accessed at 13-04-2022.

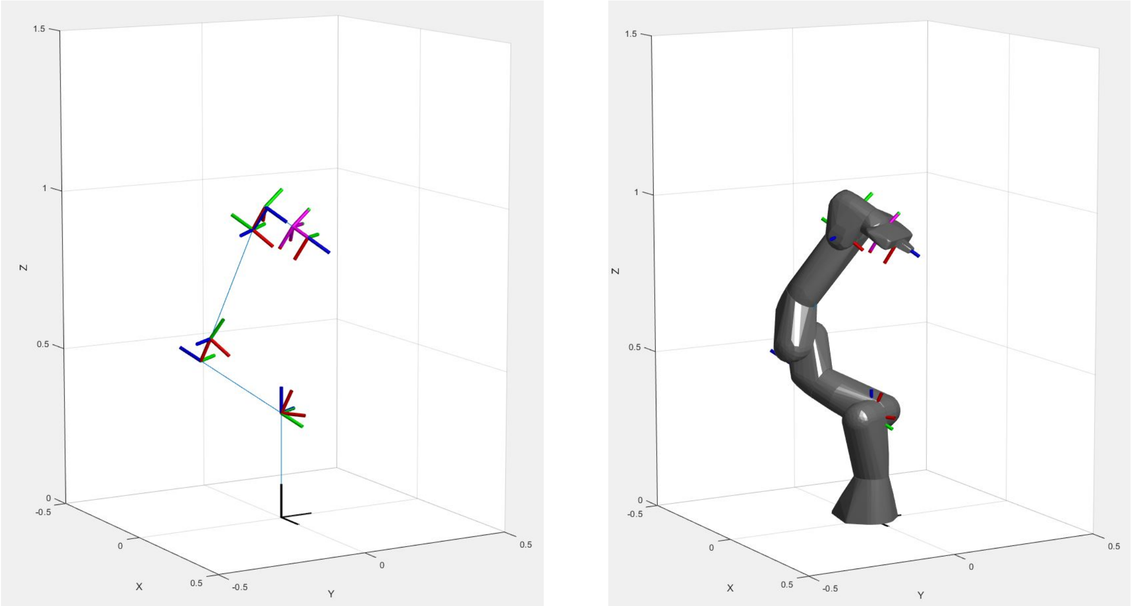
Which resulted in the fact that only the 1:1 model of the Delfi-PQ was eligible for relative motion using the Panda.

Panda Control

Now that the limits and characteristics of the Panda have been discussed, it is left to explain the control of the Panda. The control of the Panda is done through a controller developed by the Cognitive Robotics research group from Delft University of Technology[8]. The controller allows the movement of the end effector, referred to as the trajectory, with respect to the reference frame specified in Figure 6.4. The input for the controller of this trajectory is specified as a vector which describe the transform, i.e. position and orientation, of the end effector sampled at $100Hz$ over the course of the trajectory. Using this controller, a pause in the motion can be executed by sending the same transform for the duration of the pause, this allows for easy implementation of stop-motion movement.

When establishing the trajectory, several aspects are important to keep in mind. First and foremost, the trajectory should be within the physical limits of the Panda as stated in the section above. Considering the assumption that the target only rotates around a single axis *[ASM-MOT-2]*, the axis of rotation is chosen to be the rotation of joint 7. This effectively reduces the DOF of the Panda to 6, to achieve the required transform for the trajectory. In advance of the experiment, the Panda joint positions are solved for every time step through a reverse kinematics solver to verify that the Panda can achieve the specified trajectory.

Secondly, the movement should be programmed correctly to follow the specified trajectory without causing collisions between the target, the robotic arm and the surroundings. The above mentioned reverse kinematics solver verifies in addition that the robotic arm does not collide with itself. However, the target and the surroundings are not checked for collision, therefore, it should be assessed during the first run of the experiment to make sure that no collisions take place. Naturally, for experiments without lateral motion, the starting pose can be selected such that no collisions will occur, since it will only rotate around joint 7. For the set of trajectories for the '3ME Experiment', the following initial pose was selected: $[0.56, -1.12, -0.04, -1.75, 0.07, 1.65, -2.49]$. These are the joint positions in radians, which were found by positioning the Panda by hand. The initial Panda pose is shown in Figure 6.6.



(a) Panda model showing the 7 joints.

(b) Panda model showing the complete geometry.

Figure 6.6.: Panda robotic arm model in MATLAB, used for the kinematic solver.

[8] Source code available on: https://github.com/franzesegiovanni/franka_ros_TUD/tree/follow_trajectory_experiment

Thirdly, the Panda and the background should interfere as little as possible with the measurements, since the Panda will unavoidably be in the FOV of both sensors. With the starting pose presented above, the Panda will not occlude the target, it will still be in sight however behind the target. Therefore, a method has been used to filter out the false measurements of both the Panda and the background. A range filter is applied to the Lidar data to focus the Lidar measurements on the target only, which is referred to as the area of interest, which is described in more detail in Section 6.5. The remaining Lidar data will be used to detect planar features, which is used as a binary filter for the visual output of the camera, i.e. the 3D feature detection as described in Section 4.2. Therefore, only the target in the area of interest will be considered and the Panda and the background will not interfere with the measurements.

Ground Truth Estimation

For a sound academic experiment, the results should be compared to the ground truth. However, in most cases the estimation of the ground truth is subject to both measurement and process errors, just like the actual proposed estimation algorithm. The proposed validation experiment is unfortunately no exception and therefore an estimation of the ground truth should be done as well. In this section some more insights are provided on the ground truth estimation methods used in this thesis.

The first method to determine the ground truth that was considered was to make use of the Optitrack system. The Optitrack system is a modular motion tracking system, which can be constructed for various applications. The system determines the position over time by sending pulses of InfraRed (IR) light and measuring and timing the reflections. Such a system was readily available and calibrated at the Cyber Zoo at the faculty of Aerospace. However, at the time of the 'Cyber Zoo Experiment' it was observed that the wavelength of the Optitrack system interfered with the Intel Realsense L515 sensor.

The problem mentioned above was taken into account with the selection of the newly procured Lidar. Thus the newly acquired Lidar showed no interference with the system due to the spectral response to a different wavelength. However, at the time of the '3ME Experiment', the Optitrack system was unfortunately not available, due to construction and associated move of the Cognitive Robotics department of 3ME. Therefore, a different, more basic, method was chosen to determine the ground truth. Namely, the one of vector addition to a common baseline with the use of a laser distance sensor, this method is visualised in the simplified 2D figure below:

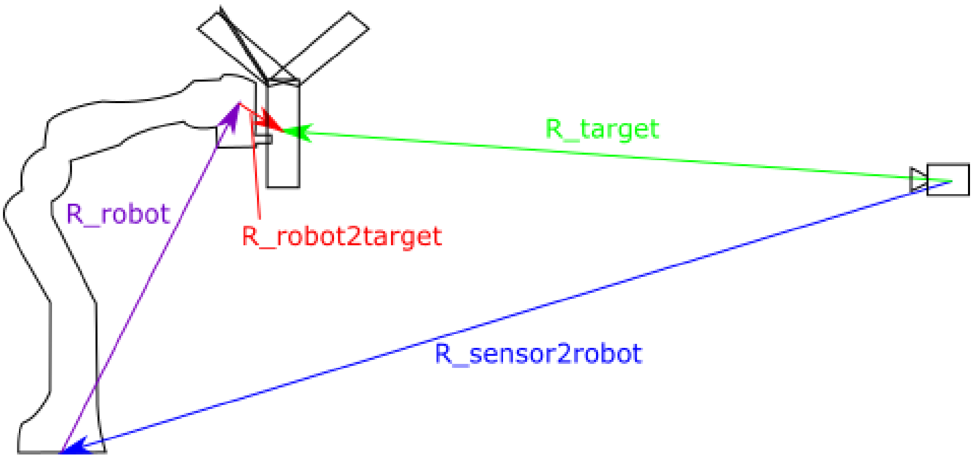


Figure 6.7.: 2D diagram of ground truth estimation.

The vector of interest is the green R_{target}, which can alternatively be determined by the vector addition of the blue $R_{sensor2robot}$, purple R_{robot} and the red $R_{robot2target}$. This can ultimately be verified by comparing both methods of determining the vector of interest. This method however, relies on three assumptions: that the measurement error of the laser is sufficiently small, the baseline for the measurements is close to perpendicular and the Panda made the correct interpretation of the own pose estimation. The full method to determine the ground truth is covered in Section 6.5.

6.4. Sensors & Illumination

As mentioned earlier in Section 6.1, three different sensors have been used during the research of this thesis. The next section will cover the sensors used in the '3ME Experiment' in more detail. Providing the most important technical specifications and a brief analysis on the suitability for the space environment. Afterwards, more information on the illumination conditions for the experiment will be provided. Which was provided by the sun simulator from an earlier project at the faculty of Aerospace.

Sensor Specifications

For the '3ME Experiment' a sensor suite was used that consisted of the Livox Mid-70 Lidar and the Daheng MER2 visual camera. This sensor suite was made to provide an alternative to the the Intel Realsense L515 sensor, which proved to be more robust to adverse illumination conditions, especially for the Lidar.

The visual camera consists of the Daheng MER2 sensor (MER2-160-227U3C) and a large aperture lens (LCM-5MP-08MM-F1.4-1.5-ND1). The visual sensor is a lightweight industrial high-resolution and high-speed camera, it features a Sony global shutter CMOS (IMX273) with a spatial resolution of 1440 x 1080 (width x height) and a pixel size of 3.45 μm x 3.45 μm. The shutter speed for this camera is adjustable in the range of: 20 $\mu s\ -\ 1\ s$. The relative spectral response for this particular visual sensor is shown in Figure 6.8.

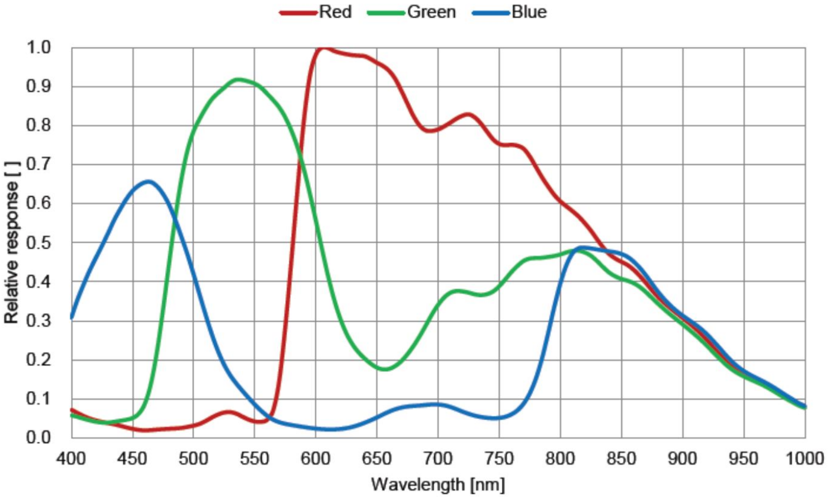


Figure 6.8.: Graph showing the relative spectral response of the MER2-160-227U3C visual sensor [38].

A significant component for the visual camera is the lens. The lens features a multi-lens system to adjust the focal length of 8 mm for the ideal working distance and an aperture of up to $f/1.4$. This large aperture allows for producing high quality images even in low illumination conditions, due to for instance low exposure times or poor lighting conditions. The fact that the lens aperture and focal length are only adjustable is unfortunate, especially considering the changing illumination conditions and working distance over time. This shortcoming was also identified by previous MSc thesis work from both David Rijlaarsdam and Marti Vilella [39, 40].

The Livox Mid-70 Lidar which has been procured is a class-1 high FOV scanning Lidar. It mainly finds its application in low-speed autonomous driving and mobile robotics. It features a 70.4° viewing angle with a non-repetitive scanning pattern illuminated by laser beams with a wavelength of 905 nm. The range error is approximately $\leq 2\ cm\ (1\ \sigma)$ at 20 m and $\leq 3\ cm\ (1\ \sigma)$ at $0.2 \sim 1\ m$. The maximum detection distance is 260 m at 80% reflectivity of the target.

As mentioned before, both sensors are not designed for use in a space environment. This paragraph will shed a little light on their respective characteristics why that is the case. The aspects of importance are the following: the rigidity to vibrations and shocks, absence of moving parts, ability to operate in a vacuum, operable temperature range and low power consumption and perhaps most importantly the robustness to adverse illumination conditions.

With respect to this rigidity, there is little known besides that they can operate on road vehicles next to mobile robotics, however this generally does not even come close to the harshness of a rocket launch. According to the manufacturer of the Livox Lidar, there are no components susceptible to outgassing, this is however unknown for the visual sensor or the lens. In contrast to the visual camera, the scanning Livox Lidar naturally included moving parts, preferably avoided on satellites due to the vibrations and mechanical wear. Finally, even though the operable temperature range of the sensor is large (-25 - 65 degrees Celsius), it is not yet sufficient for space use. For the reasons above, it can be concluded that the Lidar sensor is unusable for space applications.

However, the last aspect which has not been discussed so far, is the robustness to adverse illumination conditions of the Livox Lidar, which is discussed in the next section.

Illumination Conditions

The illumination conditions play an important role when using EO sensors for computer vision, which directly applies to the work of this thesis as well. For instance visual cameras suffer from high contrast, background noise and low SNR, especially in the space environment [41].

Even though the effect of illumination conditions will not be researched thoroughly, it is still assumed that the Sun is the source of light for the scene. To emulate the illumination conditions as close to reality as possible, use was made of a sun simulator. This sun simulator was designed, constructed and verified in an earlier project for the Microsat Engineering course at the faculty of Aerospace [42].

The spectrum of this solar simulator did closely resemble the desired spectrum of solar light in space conditions. A big drawback however, is that the sun simulator was designed to test solar panels and not to illuminate targets for relative navigation. Next to that, the intensity of the light was only 40% of the desired intensity, at a distance of 30 cm. For the experiment, the distance is in the range of 1.6 m to 2 m, reducing the intensity by at least a factor of 25. Which naturally effects the extend of realism for this experiment, as is shown in Figure 6.9.

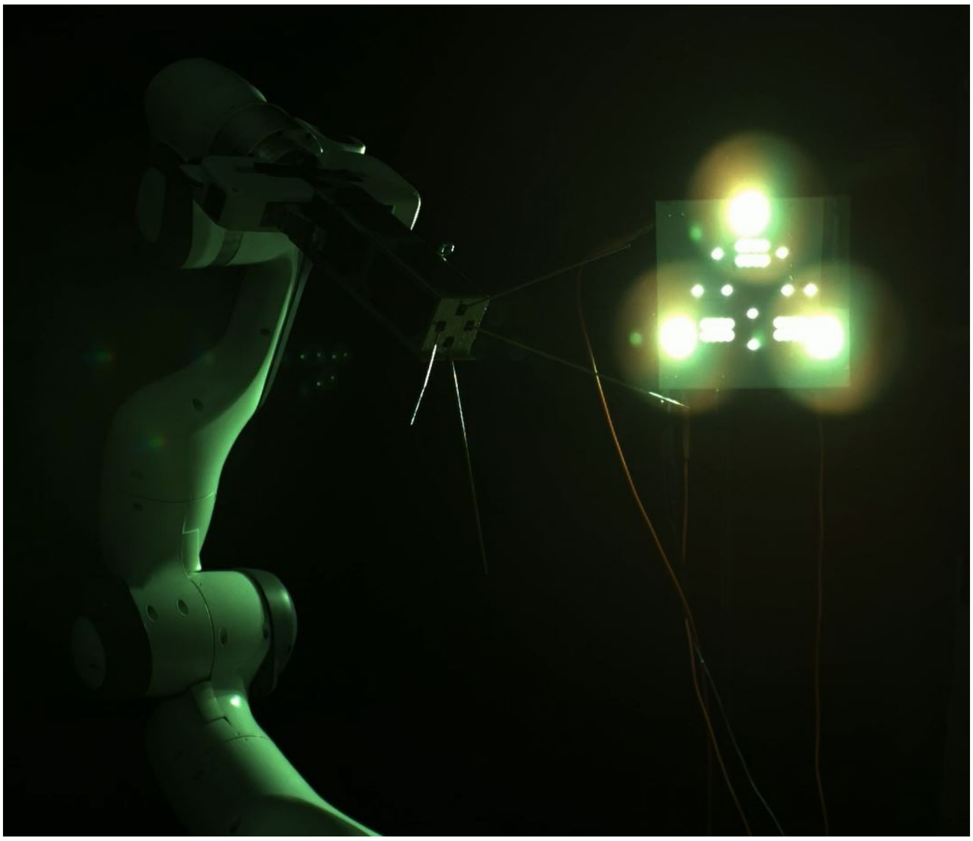


Figure 6.9.: Image displaying a frame of the visual camera in adverse illumination conditions.

The image above shows that the visual camera still performs decently, since the target is still distinguishable from its surroundings. Thus, these conditions underline that it is currently difficult to emulate the illumination conditions in a lab-environment at the faculty of Aerospace. Which emphasises the assumption to not research the effects of adverse illumination conditions more thoroughly *[ASM-EC-2 & 3]*. Although, it does add realism for the normal experiment under normal conditions compared to the office lighting, at least reducing the SNR of both the visual camera and the Lidar, reducing the quality to a certain extend. Therefore, the target will be illuminated by either the sun simulator or office lighting during the experiments.

6.5. 3ME Experiment Execution

The '3ME Experiment' is considered to be the culmination of all the experiments performed during this thesis. This is due to the fact that it made use of high quality sensors and carried out the relative motion using the panda robotic arm. In the previous two section, more information on the equipment for this experiment has been provided. In this section the actual execution of the experiment will be covered, which will present the experiment setup, trajectories and a final consideration with respect to the output data.

Experiment Setup

In the previous section, a preliminary overview of the experiment was given. As a quick recap, the only actuation will be carried out by the Panda, which is only able to lift the 1:1 model of the Delfi-PQ. The motion of the target will be recorded by the sensor suite, which consists of the Daheng MER2 visual camera and the Livox Mid 70 scanning Lidar. The experiment will take place at an office at the Cognitive

Robotics department of 3ME. The lighting of the target will be either provided by the solar simulator or the office lighting. All these elements are displayed in the image shown in Figure 6.10 below:

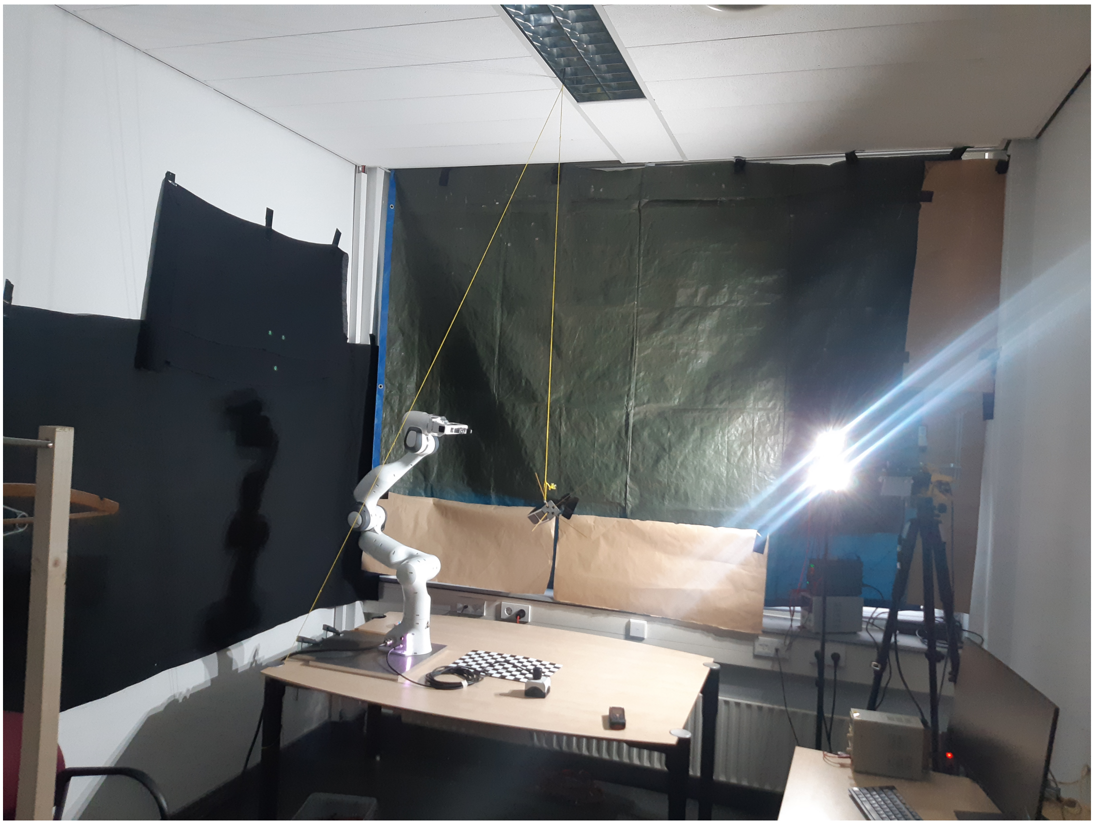


Figure 6.10.: Image displaying the 3ME experiment setup.

In the image above, three things stand out which have not been described before. Such as the fact that the windows are covered as much as possible to decrease stray light from the outdoors as much as possible. This was eventually verified by confirming that the visual camera output was indeed black, i.e. no light source was powerful enough to be detected by the visual camera.

Furthermore, the background of the target is covered with black tarp, to remove as many background features as possible. For this work it was not required due to the fact that background features are not considered since they do not align with the detected 3D planes of the target. Nevertheless, this data set might be used for a different method by another person in the astronautical community, this way the data is still useful to them.

Lastly, the attentive observer will see that in this image the Delfi-PQ hangs on yellow line suspended from the ceiling. This is to prevent damage to the model, since the gripper was not always tight enough to hold the model. In later stages during the experiment, the model was fastened to the end effector using duct tape and tie wraps.

Trajectories

The trajectories for the target are largely determined by the fact that a scanning Lidar is used. Accordingly, the integration time of the Lidar has to be considered, which is maximum 3 seconds for the Livox Mid 70 Lidar. Thus, for this experiment stop-motion is the type of movement for the trajectory. Next to the choice between continuous and stop-motion movement, several other factors can be tweaked for this experiment, which are listed below:

- Relative motion
  - Rotation and/or rectilinear motion
  - Step size
  - Different rotation vectors
- Illumination conditions
  - No lighting
  - Office lighting
  - Solar simulator
  - Incidence angle

It is important to experiment with a wide variety of scenarios to verify that the algorithm will perform correctly in all foreseen conditions. Additionally, an analysis can be done of the effect of various conditions on the algorithms performance. These different scenarios can be attributed to the different variables described above, which can be either discrete or continuous variables. Next to that, there are also sensor settings which can be varied. However, taking all these variables into account, it is clear that this domain of scenario variables is too extensive to test entirely. Therefore, a concise selection is made to provide enough data for accurate conclusions. These trajectories are stated in Table 6.3.

Table 6.3.: The trajectories carried out and recorded during the '3ME Experiment'.

Version	Trajectory	Translational Motion	Rotational Motion	Lighting	Interval	Steps
A	5	0	[5, -5] deg/s	Office	n/a	n/a
B	5	0	[5, -5] deg/s	Solar	n/a	n/a
C	5	0	[5, -5] deg/s	None	n/a	n/a
D	6	0	[15, -15] deg/step	Office	6	25
E	6	0	[15, -15] deg/step	Solar	6	25
F	6	0	[15, -15] deg/step	None	6	25
G	7	[0.01, -0.01] m/sec	[5, -5] deg/s	Office	n/a	n/a
H	7	[0.01, -0.01] m/sec	[5, -5] deg/s	Solar	n/a	n/a
I	7	[0.01, -0.01] m/sec	[5, -5] deg/s	None	n/a	n/a
J	8	[0.03, -0.03] m/step	[15, -15] deg/step	Office	6	25
K	8	[0.03, -0.03] m/step	[15, -15] deg/step	Solar	6	25
L	8	[0.03, -0.03] m/step	[15, -15] deg/step	None	6	25
M	10	[0.03, -0.03] m/step	[5, -5] deg/step	Office	6	25
N	10	[0.03, -0.03] m/step	[5, -5] deg/step	Solar	6	25
O	10	[0.03, -0.03] m/step	[5, -5] deg/step	None	6	25
P	n/a	0	0	Adverse	n/a	n/a

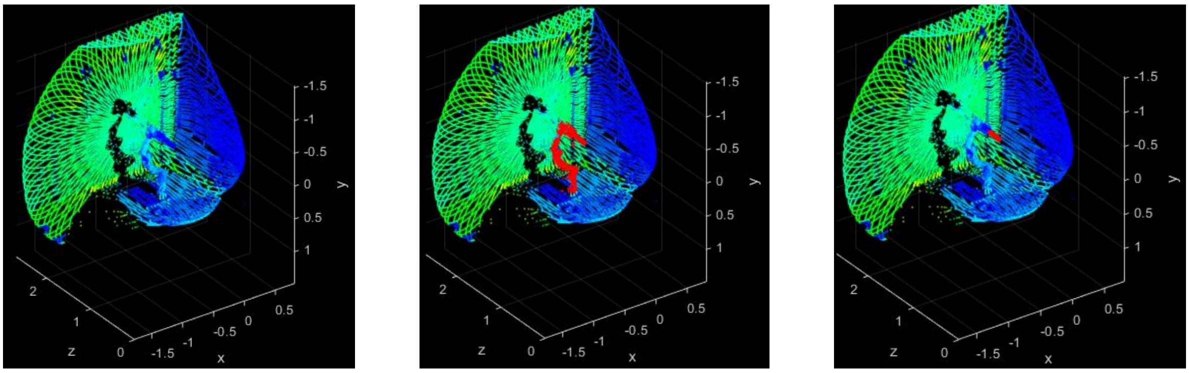
The different versions (A, B, C, etc.) of the experiment show the different trajectories (5-8 & 10) which are either continuous or stop-motion, with a specified speed or step size, respectively. As can be seen in the table, both the positive and negative speed/step size is depicted, this is due to the fact that the trajectory is back and forth, both for the translation and rotation. All experiments were carried out in office lighting, solar lighting or no lighting. With the exception of the final experiment, which was recorded under adverse illumination by the solar simulator, to get an idea of the robustness of the sensors to adverse illumination conditions.

Experiment Data

As required, during the experiment recordings were taken by both the visual camera and the Lidar. These recordings will be used as input data for the multimodal feature detection and subsequent relative pose estimation. However, these recordings can not be used as input directly. Some additional preprocessing is required, even before the preprocessing steps described in Section 4.1.

The first preprocessing step mainly relates to the stop-motion trajectory of the experiment. Only the recordings should be used that correspond to the correct step in the sequence. This is especially of concern for the scanning Livox Lidar, which requires 3 seconds for a fully integrated point cloud. When the motion is paused for at least 3 seconds, the point cloud can be selected as a whole and exported as a `.las` file. For the visual camera only a single frame should be selected during the duration it is paused. Now, the visual camera images can be used directly as input for the general preprocessing steps before the multimodal feature detection.

For the point cloud however, more preprocessing is required, due to the point measurement of objects other than the target. As was mentioned earlier, both the Panda and the background will also be included in the point cloud. Therefore, only the point cloud measurements located in the Region Of Interest (ROI) will be selected. This step needs to be done in MATLAB since no such functionality is provided with the Livox software. This is done using the function `findPointsInROI` and is set on the target with only the Panda end effector still in the ROI. Also with only the target in the ROI. This shows that only a fraction of the points in the point cloud directly represent the target.



(a) The original point cloud with no highlighted ROI.

(b) Both the Panda and the target highlighted by the red ROI.

(c) Only the target highlighted by the red ROI.

Figure 6.11.: Demonstration of ROI on Lidar measurement from experiment, highlighting different ROIs for the same point cloud.

7. Results

In the previous chapters the work has been presented to perform multimodal feature detection and subsequent relative pose estimation. The proposed algorithms enabled the processing of the data acquired by both the simulation and the experiments. Now, the results of these processes will be combined, analysed and discussed in this chapter. By doing so, the work can be discussed in a structured way, such that finally the correct conclusions can be drawn as a result of this work.

Firstly, a clear overview will be given of the origin of the different results in Section 7.1. Afterwards, the verification and validation of the different elements of this thesis work will be presented in Section 7.2. This is followed by a sensitivity analysis in Section 7.3, for both the feature detection method and the relative pose estimation method. Finally, in Section 7.4 a critical discussion is held on the research and associated results, discussing the requirements, limitations and possible improvements of this research.

7.1. Results Origin

As demonstrated throughout this thesis, the work produced several different results which can be taken into consideration for the discussion and conclusion. It is important to recognise the relevant results for this thesis and their origin. Therefore, this will be presented in detail in this section. Additionally, the representation of the results will be presented, to create a clear understanding of the results that follow.

Thus, the relevant results are provided by different parts treated during this thesis. Chronologically, it started with the simulation, followed by an initial experiment, then the feature detection methods, the relative pose estimation and the final experiment. These parts and their respective results are discussed below:

- The simulation in Blender did not only provide the sensor measurement required for the feature detection method, it provided the ground truth for the relative pose as well. Since the relative pose was determined at each frame to establish the scene. Later, it proved useful to determine the ground truth of the features as well, which was done using the Harris corner detection method as discussed in the previous chapter.
- With respect to the experiments, these did not only provide results, but also to test their suitability as discussed in Section 6.1. The first experiment did provide a data set for initial tests of the algorithm. However, the latest experiment (3ME Experiment) is considered to be useful for the results, since the ground truth could be determined as discussed in Section 6.3 & 6.5.
- The feature estimation methods resulted in different types of features, which proved to be either useful or some less so. The 3D feature detection method did provide the 2D features, the 3D planes and the 3D features. Of which, only the latter (multimodal 3D features) will be analysed in detail, as it will directly prove if the unimodal methods were implemented correctly.
- Finally the relative pose estimation provides not only the relative pose, but also the feature positions in the reference frame of the target. The relative pose can be easily analysed by comparing this to the ground truth, whereas the feature positions cannot. This is due to the fact that the feature positions are not deterministic for SURF features, there is no direct point which can be related to the target model. This is, in turn, the reason why computer vision methods are often verified through an end-to-end process, i.e. the relative pose estimation.

When comparing the 3D features to the ground truth, it might be hard to notice the subtle differences since these are on pixel-level, especially in the case without Lidar noise. Therefore, when analysing these differences, the absolute error of the 3D feature location is displayed, which can also be displayed in the three DOFs of position (X, Y & Z). An example of this is shown in the next section, in Figure 7.1.

7.2. Verification & Validation

As already stated in the research approach in Section 3.3, the verification and validation of this work is highly regarded for this work. Particularly since a new method has been proposed to be used in the field of spacecraft relative navigation. Recall that the relative pose estimation serves as an end-to-end task to verify the 3D feature detection method. In this section, the different parts of the work will be verified and validated. But first, an overview is provided in the table below:

Table 7.1.: Table showing the verification and validation activities.

Activity	Task
Verification 3D feature detection	Verify the correct flow of information throughout the algorithm.
	Verify that the 3D feature position is estimated within 1 pixel in image plane.
	Verify the above, but within the standard deviation of Lidar depth noise.
Validation 3D feature detection	Perform 3D feature detection on experiment data
Verification relative pose estimation	Verify the correct flow of information throughout the algorithm.
	Verify that the filter converges to the correct pose.
Validation relative pose estimation (end-to-end process)	Does the algorithm provide the relative pose?

The activities in the table above will be carried out in the same order as presented in the table. As can be seen in the table, the data acquired through simulation will play a role in every verification & validation activity, whereas the data acquired through experiments will only be used for validation of the proposed methods. In addition, the attentive reader will notice that the verification will always come in advance of the validation of its respective part, as is required.

3D Feature Detection

For the verification of an algorithm, one of the first steps is to verify the correct flow of information throughout the algorithm. Recall that this work only considers recorded data as input as opposed to real-time acquisition. This means that the 2D feature and 3D plane detection can be carried out sequentially[1]. For the 3D feature detection, this can be done according to the algorithm architecture shown in Figure 3.1 and the detailed explanation in Section 4.2. From that it can be concluded that the images and point clouds are (pre)processed for 2D feature and 3D plane detection. After both are done, the 3D feature location can be estimated, which proves a correct flow of the 3D feature detection method.

Now that the flow of information has been verified, the 3D feature detection method will be verified to work correctly on simulated recordings of a development target with no noise. The development target is in this case a well-lit cuboid target with clear corners for Harris corner detection, as visualised in Figure 6.3. This will be done first without the simulated depth noise on the Lidar and afterwards with simulated depth noise.

For this particular development target, the number of detectable Harris corners is equal to four, which are all located on the same plane. The result of the 3D feature detection using Harris corners is compared to the ground truth, which originates from Blender. The resulting Euclidean distances of the four points are in the order of millimeters, as can be seen below in Figure 7.1.

[1] For a real-time version of the proposed algorithm, the 2D feature detection and 3D plane detection can be carried out in parallel, as long as the 3D feature position estimation is done after these two processes have completed.

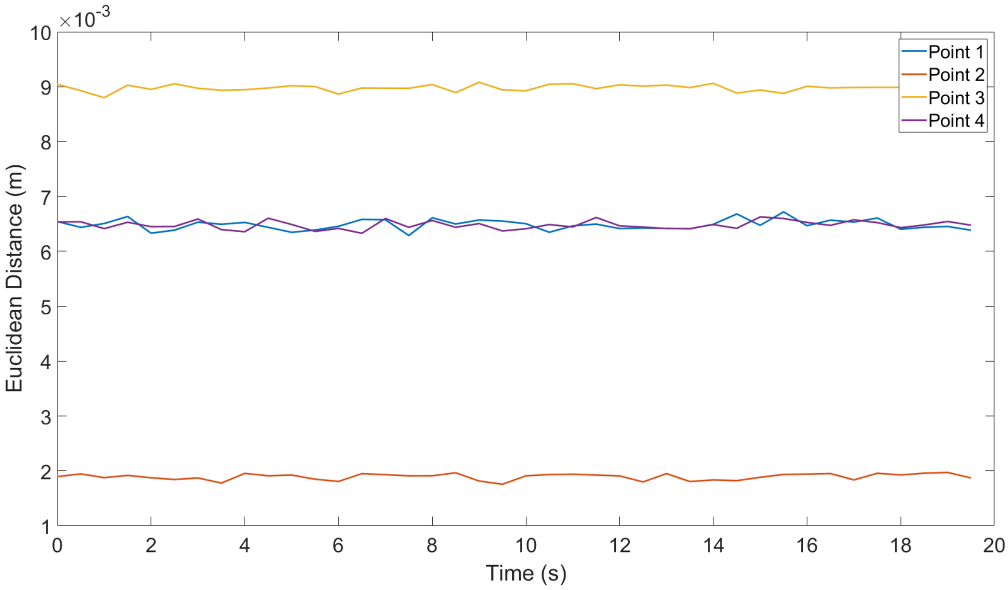


Figure 7.1.: Euclidean error of 3D feature detection using four Harris corners, for a scenario without Lidar noise.

For this particular scenario, no Lidar noise was added. However, there is both an offset from the ground truth and a random noise apparent in Figure 7.1. The offset from the ground truth can be attributed to the partitioning of the pixels, due to the spatial resolution at the specified distance. In other words, there are not enough pixels spatially available to describe certain details. The detected Harris corners are detected at 1 pixel accuracy. This leads to a theoretical maximum error of $9.8765 \cdot 10^{-04}$ m for the Y distance at the optical center, since this is the Y spatial resolution based on the Instantaneous Field Of View (IFOV):

$$\begin{aligned} A &= D \cdot IFOV \\ &= D \cdot 2 \cdot \arctan\left(\frac{d}{2 \cdot f}\right) \\ &= 9 \cdot 2 \cdot \arctan\left(\frac{0.032/1080}{2 \cdot 0.030}\right) \end{aligned} \tag{7.1}$$

Due to the angle with respect to the plane at the specified distance, the maximum error will be larger when farther away from the optical center of the image. The X and Y error for this scenario is shown in the following two figures, Figure 7.2 and Figure 7.3, respectively.

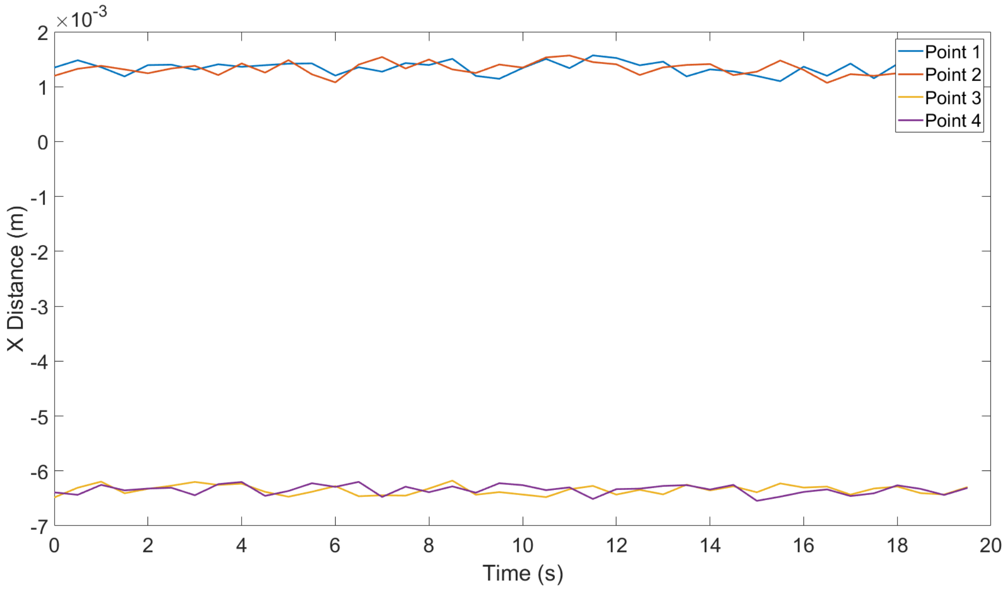


Figure 7.2.: X error of 3D feature detection using four Harris corners, for a scenario without Lidar noise.

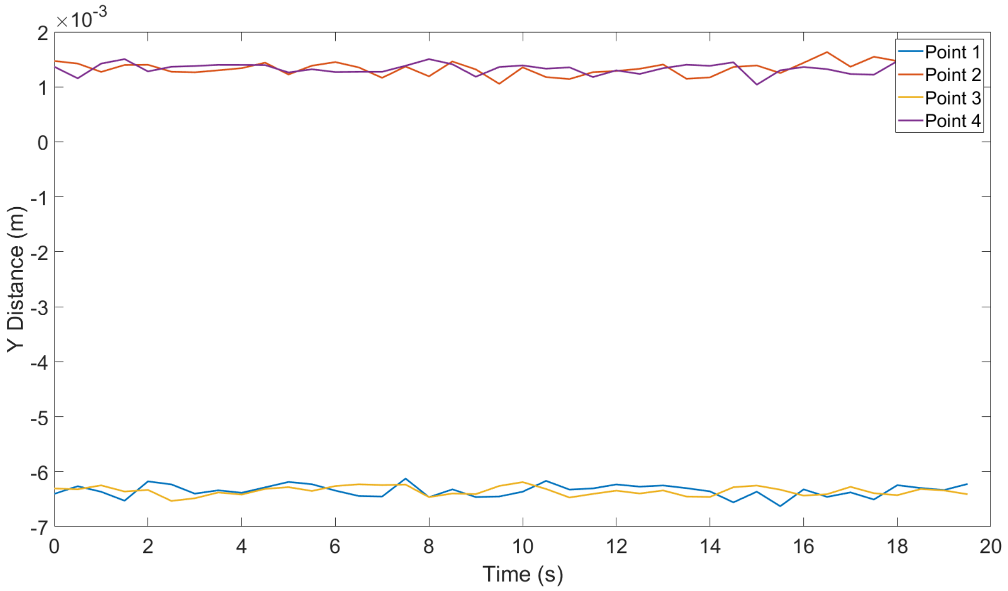


Figure 7.3.: Y error of 3D feature detection using four Harris corners, for a scenario without Lidar noise.

In the two figures it can be seen that two of each points is either higher or lower compared to the ground truth. These two points are located on the same vertical line and horizontal line for the X (Figure 7.2) and Y (Figure 7.2) distance, respectively. Which results in Point 3 (yellow line) to be the highest Euclidean distance and Point 2 (orange line) to be the lowest Euclidean distance, as can be seen in Figure 7.1.

The noise can be attributed to the fact that plane fitting using RANSAC is a stochastic process, meaning it may still deviate slightly from the ground truth. This is showed by the noise in the Z distance, which is relatively small (i.e. 1%) compared to the X and Y distances. Due to the fact that the point cloud is constructed using a perspective camera model, the noise directly leads to noise in the X and Y distances as well.

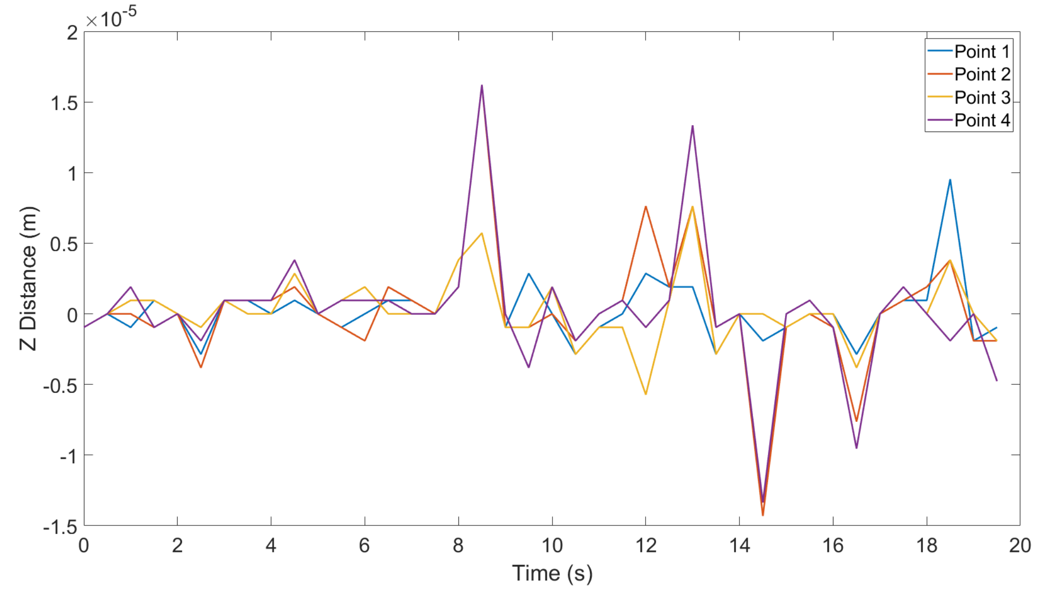


Figure 7.4.: Z error of 3D feature detection using four Harris corners, for a scenario without Lidar noise.

The previous scenario and associated results showed a that the 3D feature location can vary up to the order of millimeters, based on the spatial resolution of the target. This is of course dependent on the feature detection method, the range and the camera model. Now, the same analysis will be shown for the same scenario but with Lidar noise. The Lidar noise adds Gaussian noise ($X \sim \mathcal{N}(\mu, \sigma^2)$) to the Lidar depth measurement, in this case $\mu = 0.015$ and $\sigma^2 = 0.01$.

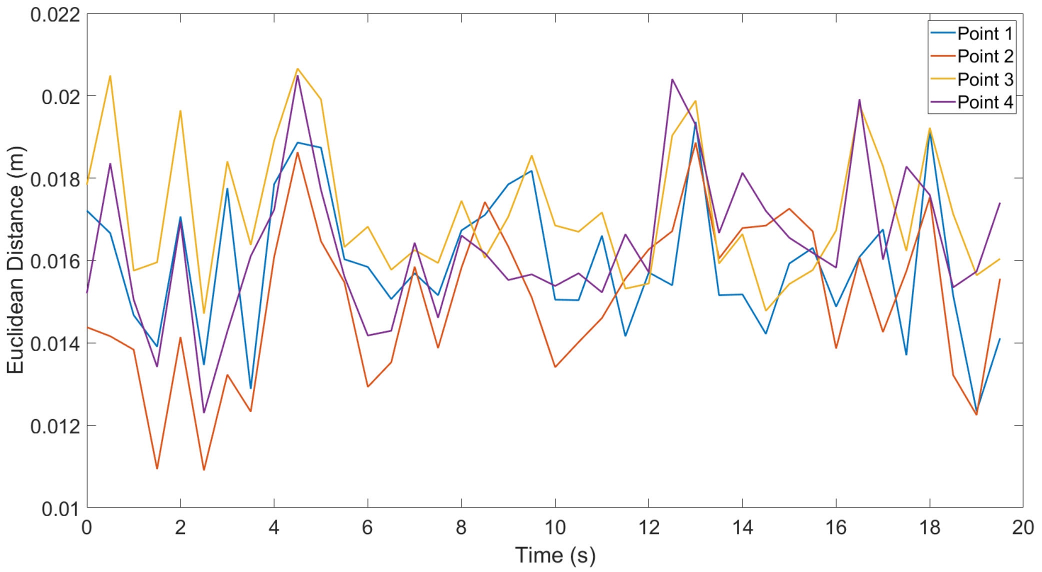


Figure 7.5.: Euclidean error of 3D feature detection using four Harris corners, for a scenario with Lidar noise: $\mathcal{N}(0.015,\ 0.01)$.

The mean of the figure above is equal to 0.0161, which is slightly higher (+ 0.011) than the specified mean of the Lidar noise. This is due to the additive nature to the Euclidean noise caused by the spatial resolution as discussed above. The error of the individual components of the position can be seen in the figures, Figure 7.6, Figure 7.7 & Figure 7.8, shown below.

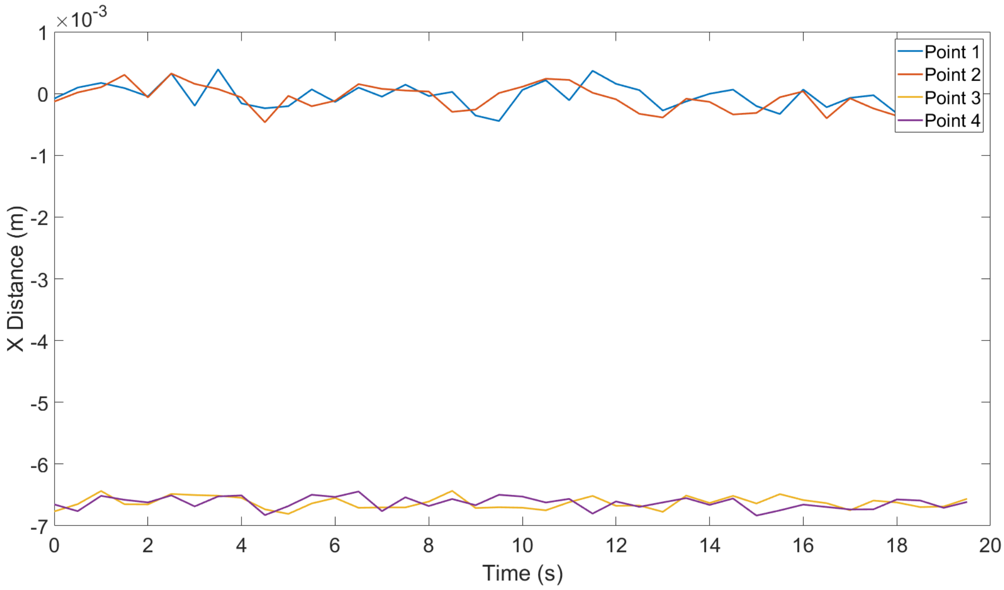


Figure 7.6.: X error of 3D feature detection using four Harris corners, for a scenario with Lidar noise: $\mathcal{N}(0.015,\ 0.01)$.

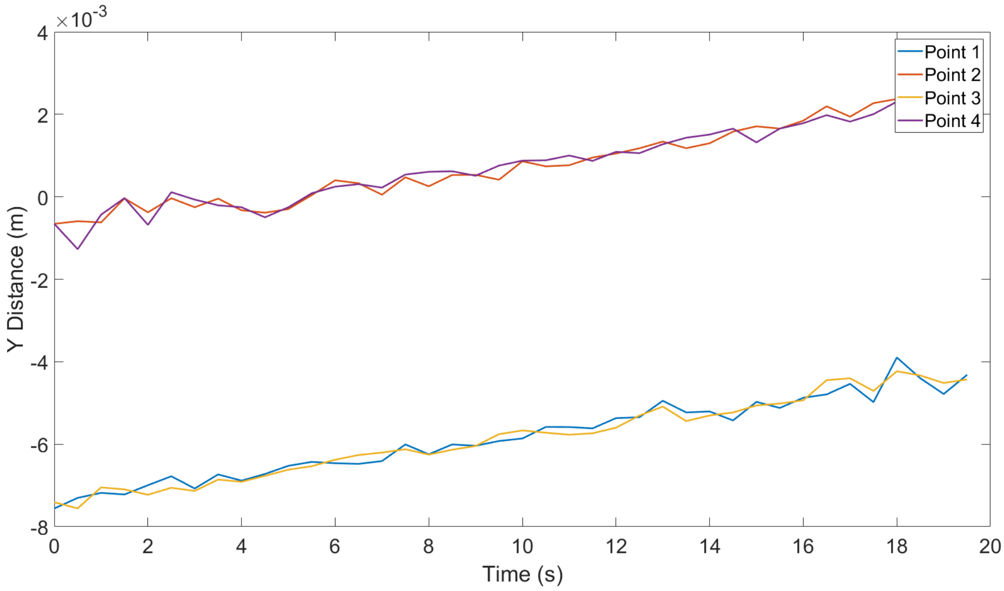


Figure 7.7.: Y error of 3D feature detection using four Harris corners, for a scenario with Lidar noise: $\mathcal{N}(0.015,\ 0.01)$.

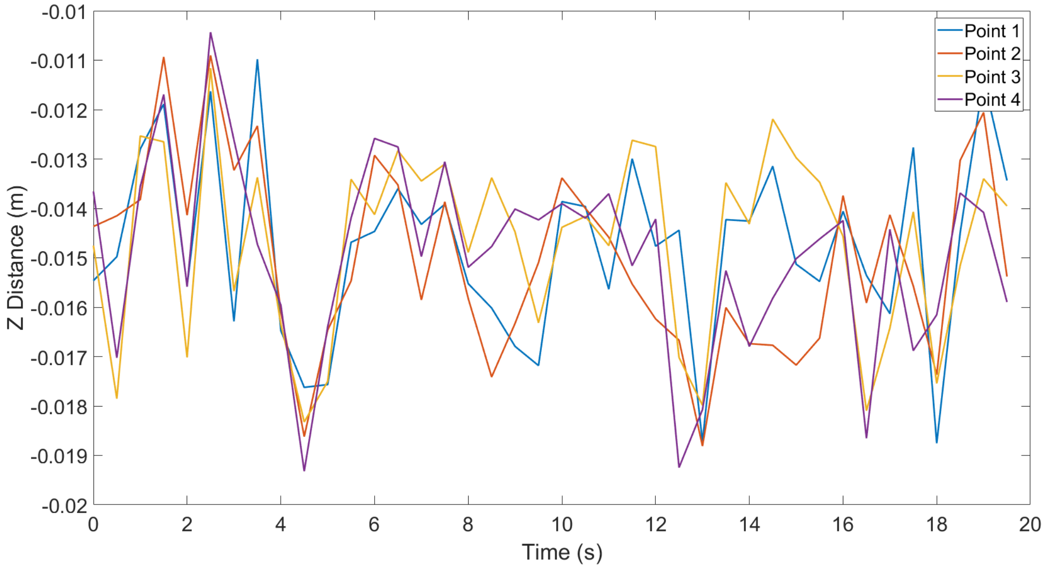


Figure 7.8.: Z error of 3D feature detection using four Harris corners, for a scenario with Lidar noise: $\mathcal{N}(0.015,\ 0.01)$.

As can be seen in the final figure, Figure 7.8, the Lidar noise is the prevalent component of the total Euclidean error for this particular scenario. From this it can be concluded that the 3D feature detection method works as expected, given that the plane detection performs to expectation and features are properly detected and matched. Thus, these two aspects are the first to follow in the verification process.

Feature Matching and Registration

As time progresses, new images and point clouds are used as input for the 3D feature detection, consequentially, the features should be matched between frames. Recall that this is done in two distinct steps, first by visual feature matching and second by feature rejection of possible wrongly associated features. Both steps require verification to provide the correct functionality to the navigation algorithm in general. However, it is assumed that the native SURF method from MATLAB is verified, and therefore direct verification of matching features is not carried out. Nevertheless, it it verified that enough features are matched over time of a recording of a Delfi-n3Xt model.

This is done by determining how many features are detected and matched as the filter progresses. This is done according to the method described in Section 4.3, of which the result is displayed identical to Figure 4.4. This is first done for a recording of a Delfi-n3Xt model without Lidar noise and afterwards with Lidar noise.

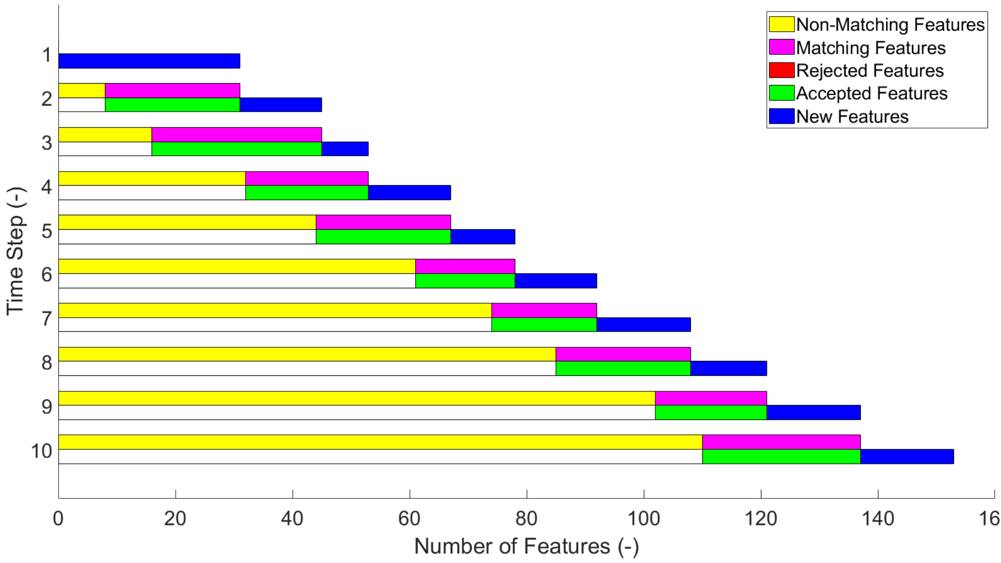


Figure 7.9.: Figure showing the number of features matched throughout the first 10 steps of a recording of the Delfi-n3Xt satellite without Lidar noise.

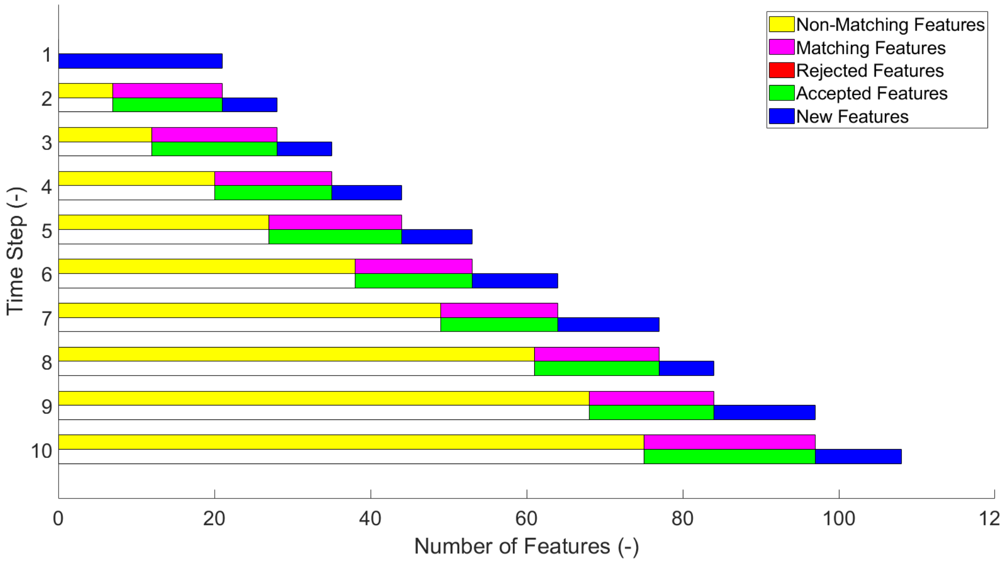


Figure 7.10.: Figure showing the number of features matched throughout the first 10 steps of a recording of the Delfi-n3Xt satellite with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

In these particular figures shown above, only the first 10 time steps are shown, for the sake of clarity. It is clear that there are less features detected in the case with Lidar noise, roughly 25% less features are available compared to the case without Lidar noise. Although for all the 40 time steps with Lidar noise, the lowest number of matched features is equal to 8, which is considered to be sufficient. It is important to note however, that in the figures above, Figure 7.9 & Figure 7.10, there are currently no rejected features apparent. This step will be verified when considering the relative pose estimation filter.

Relative Pose Estimation Filter

As stated before, the ground truth is not available for the 3D feature detection method using SURF features. Therefore, the relative pose estimation filter serves as an end-to-end process to verify the proposed 3D feature detection method. Additionally, it can be analysed if the proposed method is suitable for spacecraft relative navigation. Just like the 3D feature detection method, the first order of business is to verify the correct flow of the algorithm.

This will be done according to the algorithm layout of the FastSLAM algorithm as can be seen in Algorithm 5.1 and Algorithm 5.2. This already shows the correct flow of information, but is a rather abstract method. Therefore, a more concrete example which shows the correct flow of the algorithm will be provided in the next paragraph.

Alternatively, the flow of the algorithm can be verified by analysing the correct propagation of the particles and associated features. This can be done by initialising a single particle with the true pose and no noise. As a result, all the features should be mapped to the target model without error. As the filter propagates, more feature measurements are added, increasing the number of features on the target model.

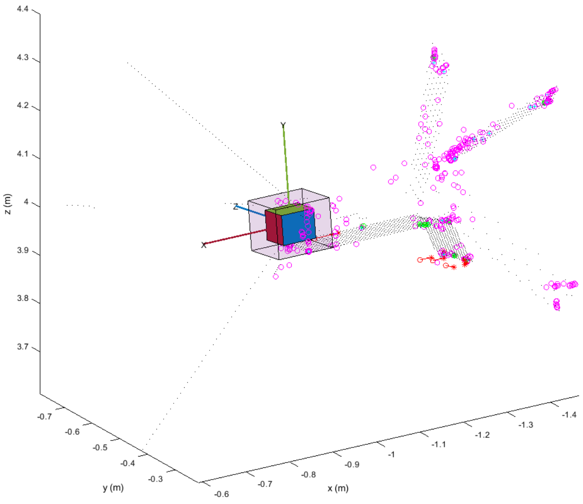


Figure 7.11.: Point cloud of the target at t_{end} with the all the feature measurements shown, which align with the point cloud. The point cloud is shown in the reference frame of the camera.

The figure, Figure 7.11, shows that the 3D feature measurements have been correctly positioned to the final point cloud. Next to the correct flow of the entire pose estimation process, this step verifies that the correct interpretation of the rotation matrices are used throughout the algorithm.

Now, the relative pose estimation filter will be tested for incrementally increasing DOFs, to verify the correct functioning of the FastSLAM method. First only the Y position will be estimated and then the full 12 DOF relative pose estimation. All relative pose estimation methods are based on a number of 50 particles, which was determined based on the sensitivity analysis on the number of particles as shown in Section 7.3, in Figures 7.18 - 7.22

For the single variable estimation the Y position was selected, since it is the only non-trivial in plane (compared to the visual image) variable. The other in plane variable is the X position, but is trivial since it aligns with the axis of rotation for this particular motion. The verification of this single variable estimation is actually shown in the next section, Section 7.3. The first sensitivity analysis is carried out for a different number of particle on the single variable estimation of the Y position. The figures 7.18 - 7.22 clearly show that the estimation converges for higher number of particles (> 25).

Now, based on the same scenario as before, the full 12 DOF relative pose estimation is verified on measurements without Lidar noise. The values are initialised according to the initialisation scheme suggested in Section 5.5.

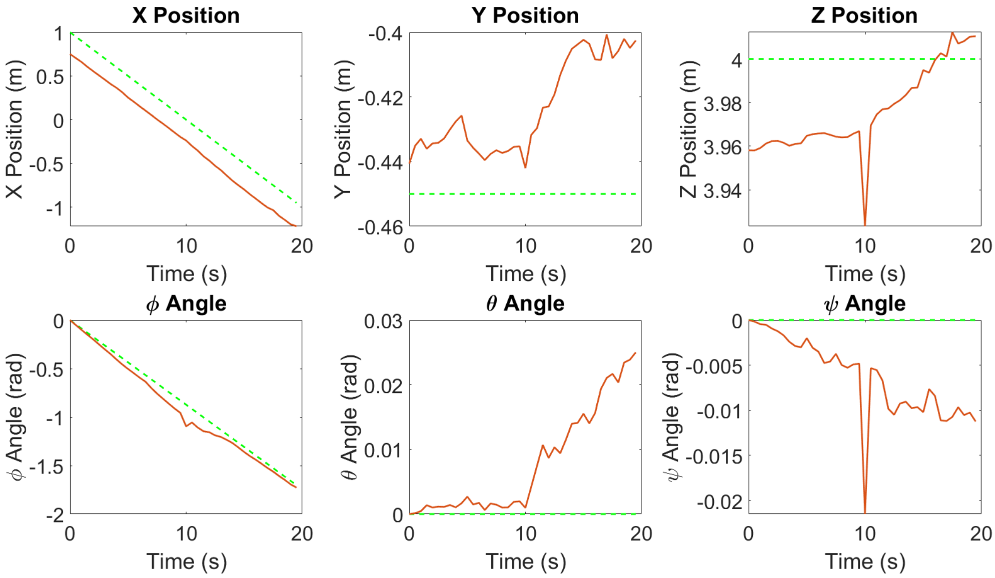


Figure 7.12.: Figure showing the resulting position of the relative pose estimation process on a simulated recording without Lidar noise.

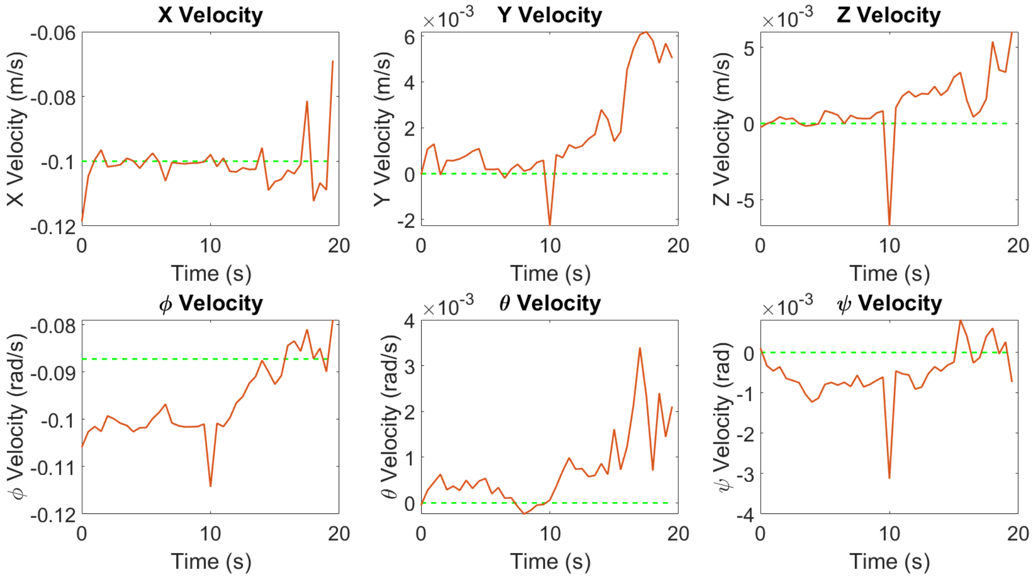


Figure 7.13.: Figure showing the resulting velocity of the relative pose estimation process on a simulated recording without Lidar noise.

The same scenario applies to the following analysis, but these are with Lidar noise. The results are shown in the following Figures 7.14 & 7.15.

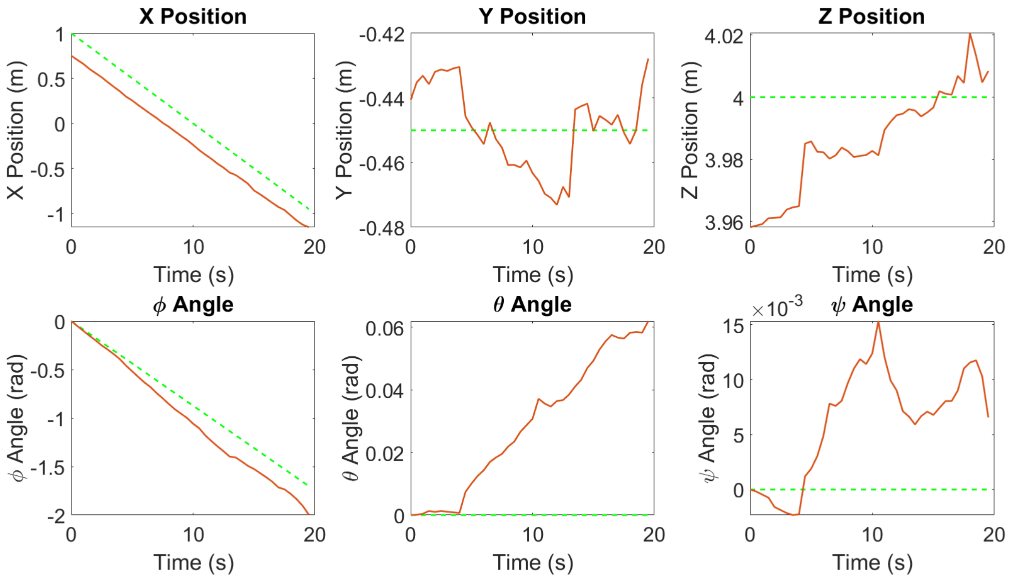


Figure 7.14.: Figure showing the resulting position of the relative pose estimation process on a simulated recording with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

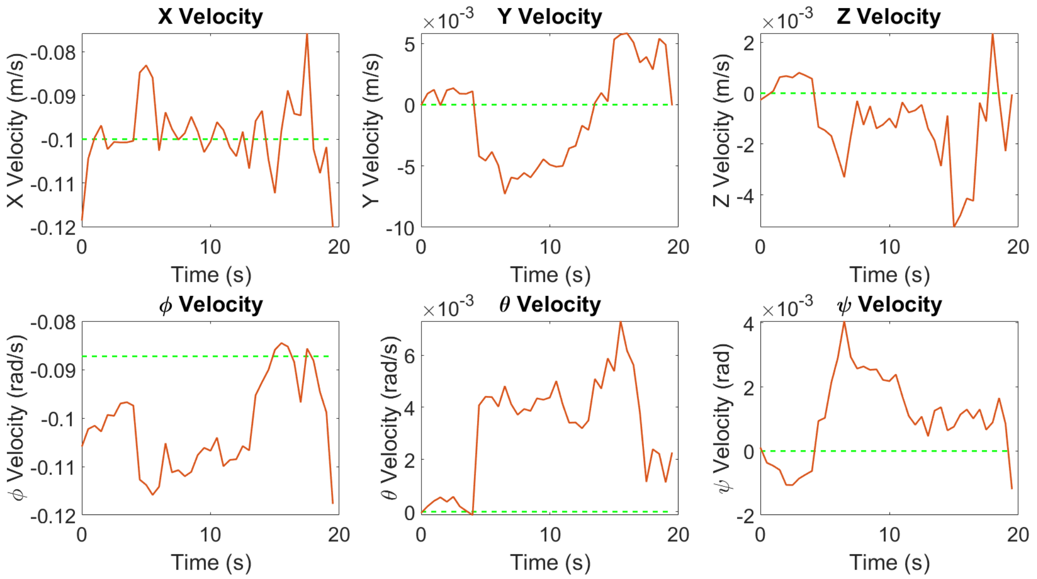


Figure 7.15.: Figure showing the resulting velocity of the relative pose estimation process on a simulated recording with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

Although both analysis, with and without Lidar noise, show significant jagged estimations of both the position and velocity, this is to be expected with a particle filter. Thus, it can be concluded that the relative pose estimation works as expected, since it converges to the true value, with the exception of the small values in the order of magnitude of 10^{-3}. This is caused by the fact that the measurements are mostly affected by noise, rather than an actual movement. This is apparent in the plots for the angles θ and ψ;

Feature Rejection

Finally, the last step to verify is the feature rejection method which is part of the EKF part of the Fast-SLAM algorithm. Using the feature rejection method as proposed in Section 5.3, the wrong data associations are detected and discarded as matches. The result of this process is visualised, identical to Figure 4.4, as described in Section 4.3. This is done first for a recording of simulated measurements with Lidar noise and afterwards without Lidar noise. It should be noted that the figures only represent the rejection of features at a single particle, since the feature rejections is based on the estimated pose of that particular particle.

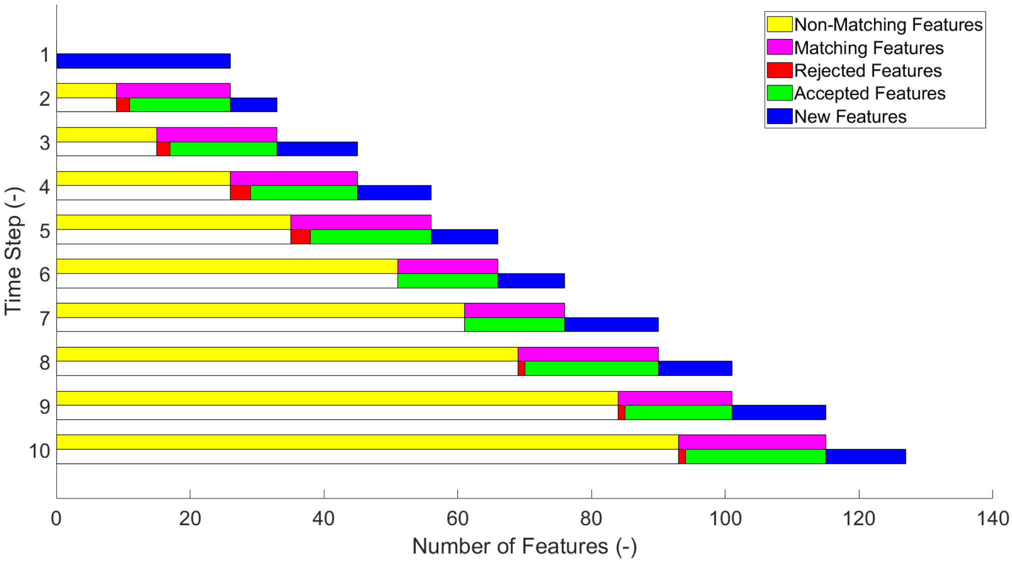


Figure 7.16.: Figure showing the number of features matched and rejected throughout the first 10 steps of a recording of the Delfi-n3Xt satellite without Lidar noise.

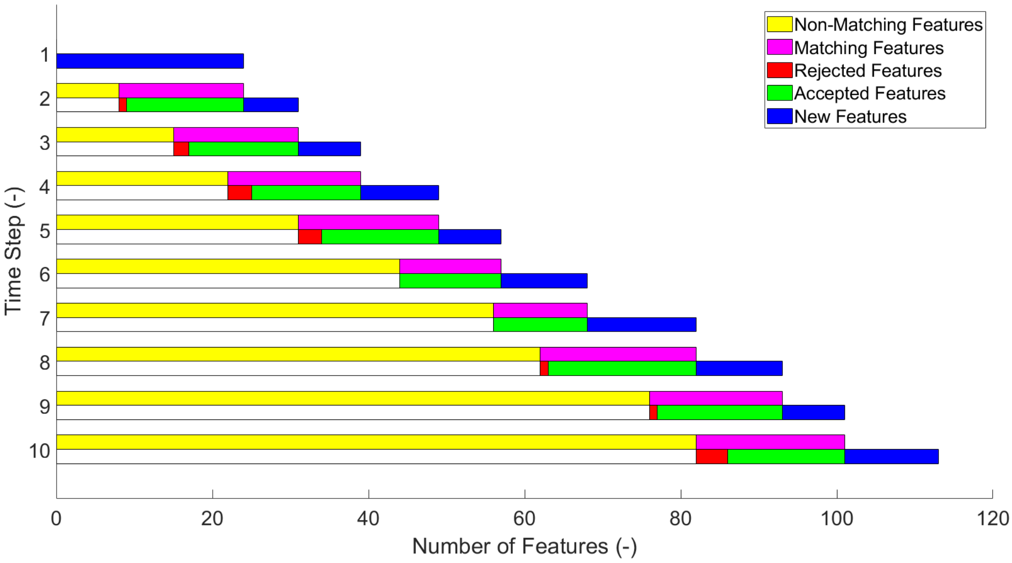


Figure 7.17.: Figure showing the number of features matched and rejected throughout the first 10 steps of a recording of the Delfi-n3Xt satellite with Lidar noise; $\mathcal{N}(-0.01,\ 0.005)$.

As opposed to earlier analysis, for this particular analysis, it is apparent that the number of rejections is not necessarily correlated to the noise of the Lidar measurement. This can be attributed to the fact that the rejection of features is only dependent on either false feature matches or wrong pose estimates. If the same features are matched, for both cases with and without noise, the same feature will be rejected. And when considering the wrong pose estimates, they do not appear to be directly correlated to Lidar noise, when comparing Figure 7.12 to Figure 7.14.

7.3. Sensitivity Analysis

For the entire proposed algorithm, there are a multitude of parameters to tune. For this particular research, the effect of the most important parameters will be analysed in more detail in this section. Firstly, the number of particles will be varied to determine the effect of this parameter on the relative pose estimation. Secondly, the starting parameters of the pose will be set to a value further from the true value, to verify for which starting parameters the algorithm still converges.

Relative Pose Estimation Filter - Number of Particles

Since the FastSLAM algorithm is a derivative of the particle filter, it is interesting to investigate the sensitivity of the relative pose estimation filter to the number of particles used in the algorithm. This sensitivity analysis is carried out for the Y estimation on a simulated recording with Lidar noise, which is the same scenario as in the previous section, Section 7.2.

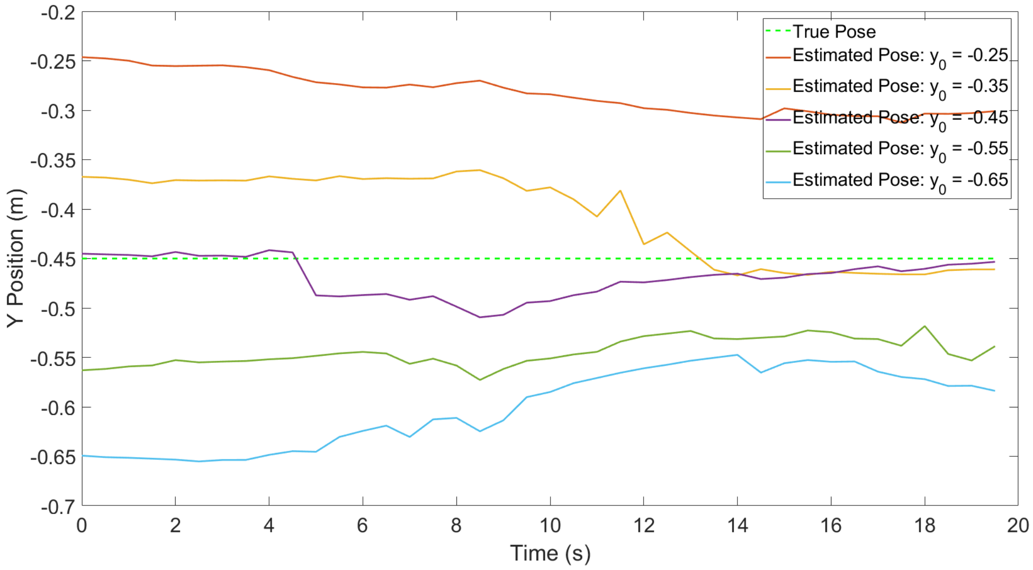


Figure 7.18.: Figure showing the Y estimation of the relative pose using 5 particles. Based on a simulated recording of Delfi-n3Xt with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

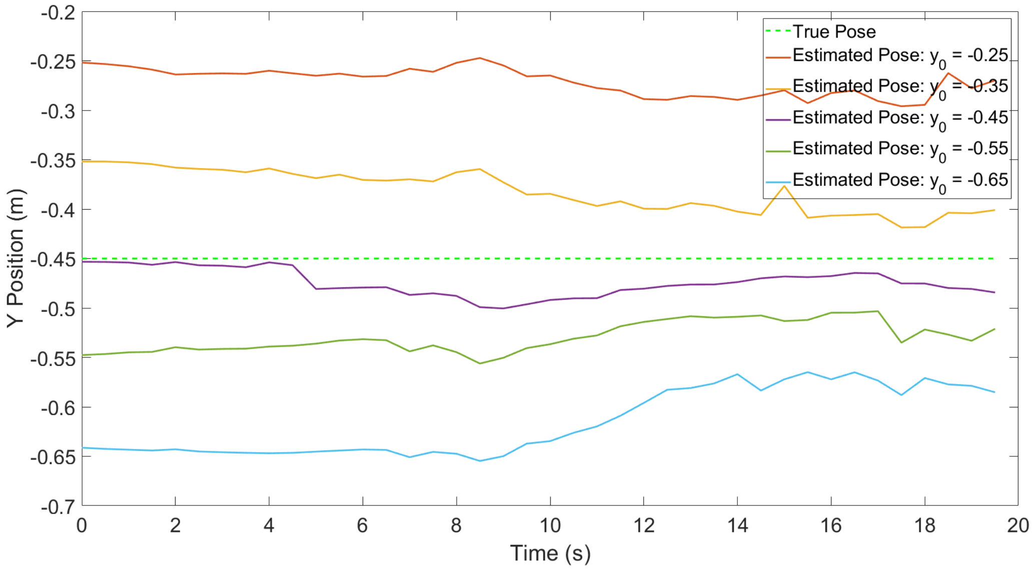


Figure 7.19.: Figure showing the Y estimation of the relative pose using 10 particles. Based on a simulated recording of Delfi-n3Xt with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

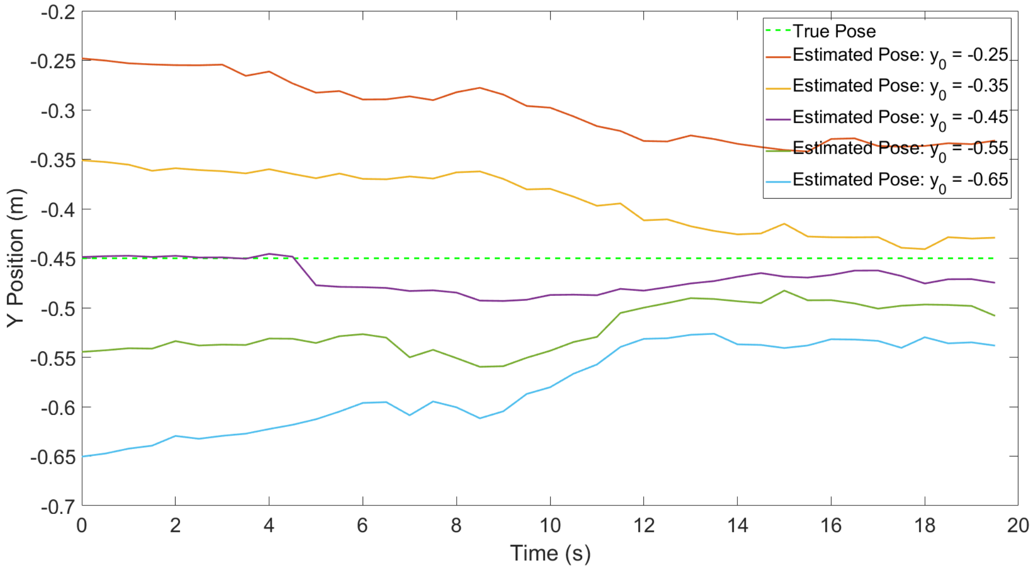


Figure 7.20.: Figure showing the Y estimation of the relative pose using 25 particles. Based on a simulated recording of Delfi-n3Xt with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

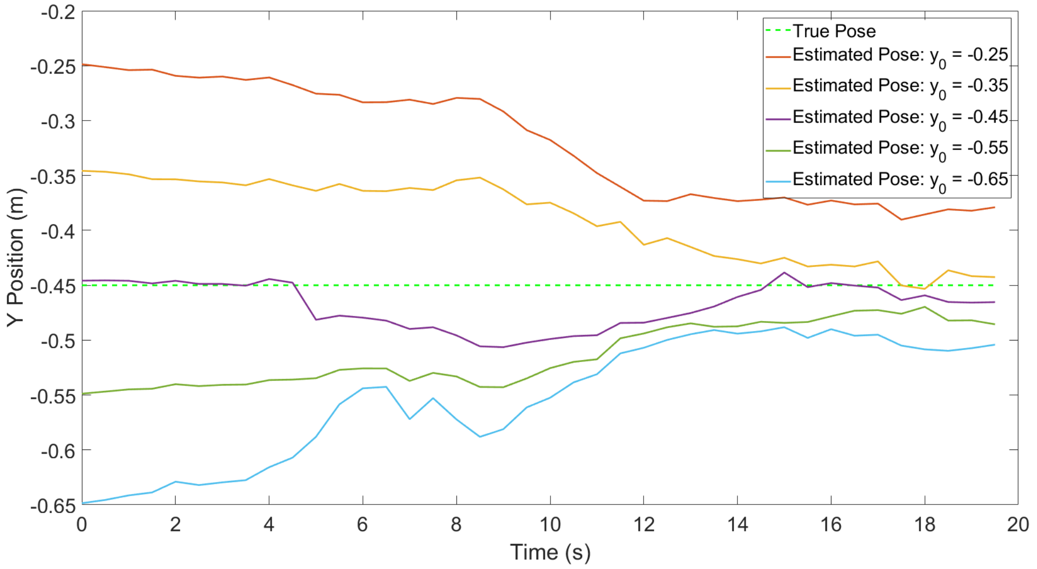


Figure 7.21.: Figure showing the Y estimation of the relative pose using 50 particles. Based on a simulated recording of Delfi-n3Xt with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

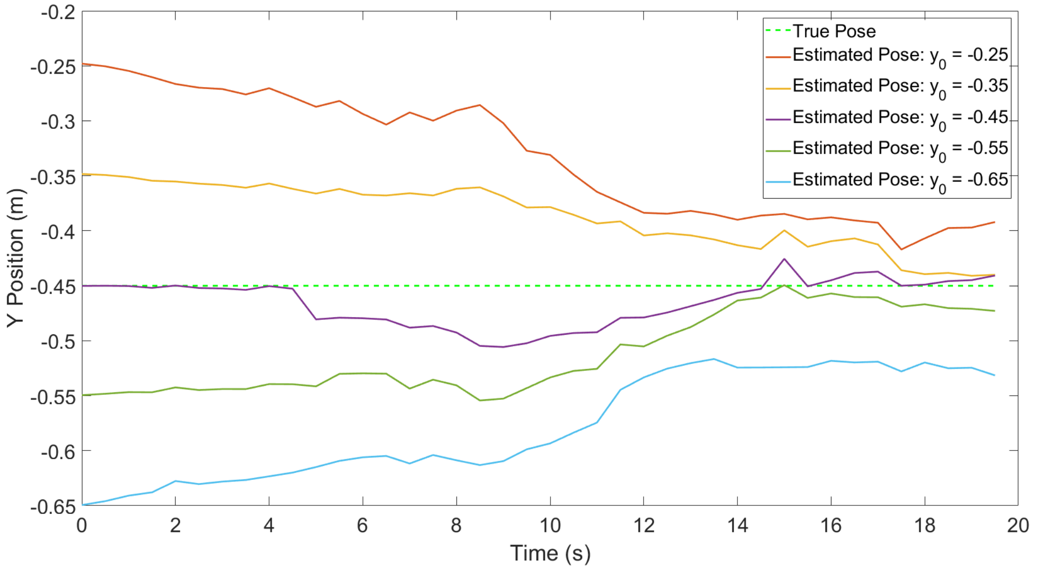


Figure 7.22.: Figure showing the Y estimation of the relative pose using 100 particles. Based on a simulated recording of Delfi-n3Xt with Lidar noise: $\mathcal{N}(-0.01,\ 0.005)$.

Relative Pose Estimation Filter - Starting Parameters

In the current state of the relative pose estimation algorithm, the initial pose estimate is based on the first measurement, as presented in Section 5.5. Moreover, as covered in Section 2.2, the close range relative navigation is preceded by the mid range relative navigation. Therefore, it is not unlikely that even better initial estimates for the pose are available, especially considering that these are dependent on the chaser control input as well. An improved initial estimate will greatly increase the initialising phase of the algorithm. However, in contrast to a more accurate estimate, the current estimate might be worse, caused by either a worse initial estimate or a slight divergence throughout time. This could be caused by adverse illumination conditions for instance.

In addition, when presenting the initialisation approach for this relative pose estimation method, it was stated that the initial angles for the attitude of the target were ambiguous. Which is proved in Figure 7.23, shown below:

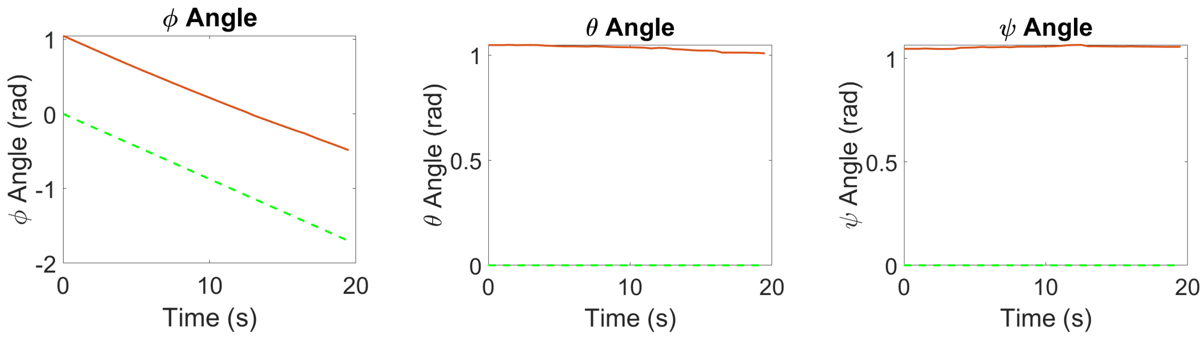


Figure 7.23.: Result showing the attitude of the relative pose estimation algorithm. The angles are initialised at a wrong value deliberately, to show that the initial angle is indeed ambiguous.

This is because the filter has no a-priori knowledge on the target, the starting value of the angles are ambiguous as expexted, since there is no available angle to compare it to. The filter executes all estimation steps with respect to the starting value, as can be seen in Figure 7.23.

7.4. Discussion

In this section, a critical look is given to the research as whole. What are the strong and weak points of this research, based on the presented work in this thesis. First the results will be discussed, followed by a list of improvements for this work.

Results

First and foremost, the proposed methods appear to be promising, but can currently not compete with earlier unimodal and multimodal relative pose estimation methods presented in earlier literature. Moreover, it appeared at the end of the thesis that the data from the experiment could not be used for its intended purpose, hopefully a researcher can make use of the data in the future.

According to the author, the fact that the proposed methods do not perform to expectations can be attributed to two main causes. Firstly, that the plane detection method requires too much tuning to work for noisy data, even then, in some cases tuning did not even enable the correct functioning of plane detection for noisy data. This is caused by wrong associations of points to the wrong plane, which is more prevalent with higher density point clouds. Thus, perhaps different Lidar resolutions can benefit the current method.

Secondly, that it appears that the particle degeneracy plays an important role in noisy and even wrong pose estimations. This is directly apparent when noting the sudden jumps in the estimation of certain pose estimates. This would be a much more fluent estimation line if the weights were more equally distributed. However, the current algorithm chooses the particles based on a Gaussian fitting function, easily reaching low or high orders of magnitude. Which directly shifts the weight towards a specific particle.

However, considering these two shortcomings, some other interesting relations were observed. Since next to the sensitivity analysis shown in Section 7.3, there were more parameters available to tune for both the 3D feature estimation and the relative pose estimation.

For the relative pose estimation, it was found that a smaller filter time step does not necessarily lead to an improved result of the pose estimation. This effect is accounted for by the fact that the EKF updates the feature location based on the measurements, a small time step will give a smaller difference in measurements between the feature state update, which could be so small, that it falls within the measurement noise of the feature detection. Therefore, it was found that the most suited time step is the one where the difference in feature measurements is greater than the measurement noise.

With respect to the feature rejection, it can be seen in Figure 7.24 that without a feature rejection method, the results are discontinuous. This is due to the fact that the weight approaches 0, due to Equation 5.5, when the wrong data association is made. When the feature weight approaches 0, the particle weight will be set to NaN due to the normalisation of the weights, as shown in Equation 5.8

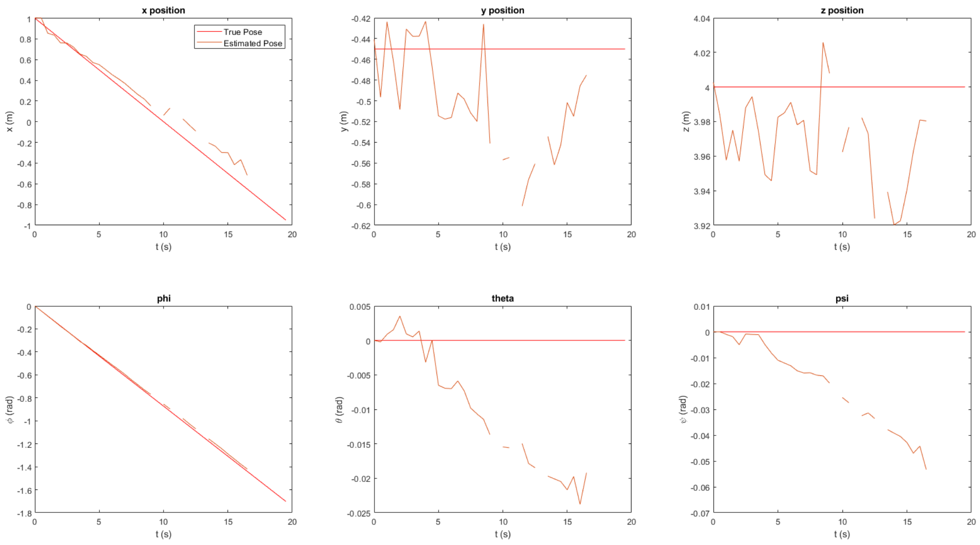


Figure 7.24.: Resulting pose of the algorithm without a feature rejection method, which results in discontinuities in the estimated pose.

Due to the fact that 3D feature locations are used as opposed to 2D feature location, the feature rejection method can use another dimension, namely the depth of the feature. Which appears to be a very beneficial extra dimension of information for the feature rejection, since it allows the rejection of features based on a wrong rotation or depth estimate for the pose.

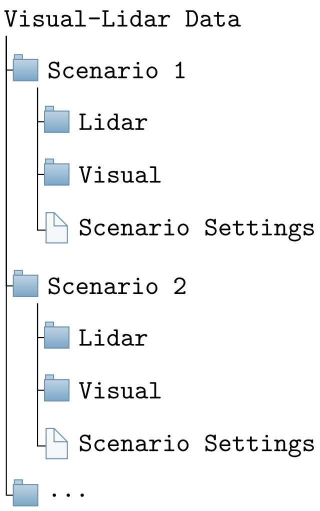
In general, we can see that the particle filter performs better for lower DOF problems. This is due to the fact that a higher dimensional state space requires a higher number of particle to completely cover the state space, increasing the computation time significantly. Perhaps another method can be thought of to determine the orientation of the target based on the determined plane equations of the detected planes. This would allow the decoupling of the position and the rotation, in which case other filters can be considered again, such as an EKF.

Suggested Improvements

Naturally, each research comes with its own limitations, mainly caused by the lack of time and resources. The previous section readily gave some critical insights into the limitations of this particular work. In this section, improvements are suggested according to the major elements handled in the thesis, being the simulation, the algorithm and the experiment. These insights might prove useful for future researchers, to avoid the pitfalls encountered during this work and to improve their own process and results.

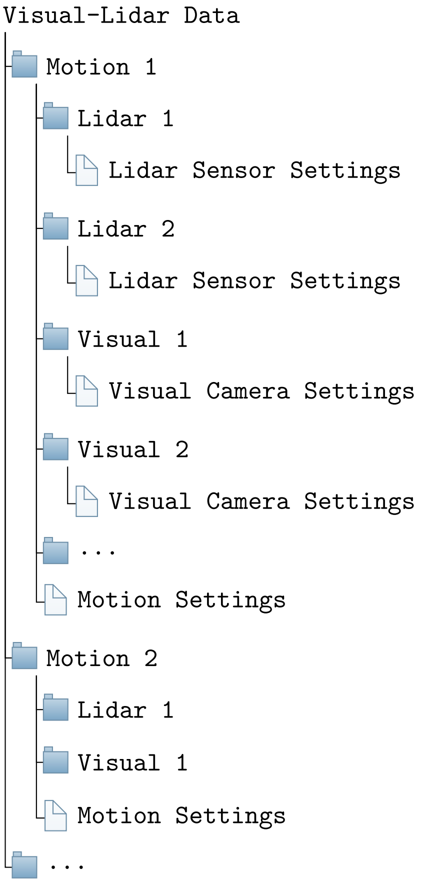
Starting off with the simulation, the suggestions range from Blensor functionalities to the simulation tool as a whole:

- For this thesis work, the last stable version of Blensor was used, based on Blender 2.8. However, it might be possible to port Blensor to a more recent version of Blender, which provides more functionalities compared to Blender 2.8. But most importantly, the current versions of Blender provide better optimised methods for accelerated simulation and workflow.
- For the current implementation of visual-Lidar data acquisition using Blender and Blensor, an option was developed to chain simulations to run sequentially, reducing time spend between roughly 30 minute to 120 minute long to set up new scenarios and starting the simulation. However, currently a different folder is made for each different setting, leading to possible duplicate simulations and data storage. This pipeline could be improved by attaining a different folder structure to decrease unnecessary simulations, as shown in the two directory trees shown below.



In the current structure shown above, each simulation creates a new folder where only the measurements from the same scenario can be combined.

Whereas the structure to the left can combine and match different elements based on the same motion. This can speed up both simulation and subsequent analysis.



- It became apparent that the 3D feature detection required specific tuning to accommodate different recordings of visual-Lidar data, such as the SURF features and the plane detection using RANSAC. There should be several more options of development targets, such as the Delfi-n3Xt or Delfi-PQ satellite covered in a checkerboard pattern. Such that the 3D feature detection method could at least be verified for realistic targets to an accuracy similar to the spatial resolution of the visual camera.
- With respect to the plane detection, it is proposed to include object labels in the visual-Lidar simulation pipeline. Blensor allows the assignment of object labels to each Lidar point throughout the point cloud, each object label indicates which object, or mesh, it belongs to. By dividing the target model in separate components, the object labels can directly be used as ground truth for plane

detection. When this particular ground truth is available, an automated process can be developed to optimise the feature detection settings for different recordings.

- Finally, the most recent version of Blensor still showed troubles with unintended behaviour and bugs. Perhaps a visual-Lidar data acquisition pipeline not based on Blensor can be developed. Such as visual images acquired using Blender 3.3 and Lidar measurements acquired using Glidar [43].

For the algorithm, both multimodal feature detection and relative pose estimation, the following improvements are considered to be beneficial:

- A better abstraction of the code would be possible if the feature register and the particles were coded using an OOP approach. Since it is considered that the nature of these elements of the algorithm are better encapsulated in such an object rather than through a consistent `struct` caused by the functional flow paradigm currently used.
- Over the course of the thesis, unit testing was added to test and verify various methods of the 3D feature estimation, the relative pose estimation filter and other required methods, such as importing data and plotting functions. However, integration tests were not added, therefore the contracts between the methods could not be tested automatically each time changes were made. The addition of these automated integration tests and visualisation would have sped up the development considerably.
- The current implementation of the relative pose estimation filter is susceptible to the gimbal lock ambiguity. This ambiguity can be resolved by following the method of a Kalman filter based on the difference in angle, rather than the absolute angle, as described in earlier literature [44].
- It seems that a better alternative is required for the current weighting function, perhaps even another filter might prove to be beneficial to determine the weights. Moreover, in the current pose estimation filter, the particle noise remains constant, thereby not ceding reliance to sustained correct measurements.
- During the research, it has not been researched how the target pose can be re-acquired after loss of measurements of the target, albeit visual or Lidar. An optimal re-initialisation procedure can be determined, based on earlier measurements from before the loss of measurements and the new measurements.

And finally with respect to the visual-Lidar data acquisition through experiments, the following improvements might benefit the research:

- Unfortunately, the approach to directly verify the 3D feature detection method was only considered after the experiments had taken place. However, if the experiment were to be carried out again, models of the development targets would be used to record new data, similar to the models displayed in Figure 6.3.
- Similar to the visual-Lidar calibration, the checkerboard can be attached rigidly to the Panda. When a recording is made while the Panda follows a trajectory using stop motion, the recording can be used to perform the visual-Lidar calibration. Moreover, perhaps the ground truth of the Panda base can be determined in this way, since the vector from the end effector to the Panda base is known.
- And lastly, the entire experiment process to carry out and record a motion of a target can be greatly improved by enabling time-synchronisation across all elements, being the Panda, the Lidar and the visual camera. This way the correct relations can be made regardless of a stop motion or continuous movement.

8. Conclusion & Recommendations

This thesis started off with the recognition of the ever increasing risks associated with space debris. To take care of these risks, research is done on the removal of these objects through Active Debris Removal missions. An element of these missions is the relative pose determination of the target, of which the features and shape could be unknown, for instance due to damage. During this MSc thesis, research was done on multimodal feature detection using visual-Lidar data, in the context of relative pose estimation of an unknown target.

The multimodal visual-Lidar approach appealed mainly since it can provide direct depth information due to the Lidar measurements. Generally, visual measurements provide higher resolution data compared to the Lidar, which is in turn used to extract and track features on the target. This combination proved to be useful for multimodal feature detection and the subsequent relative pose estimation. The work has been verified and validated through computational simulation.

This chapter will present on the conclusions made based on the results and reflect on the work done during the MSc thesis. First, the conclusions that can be drawn from this research will be presented in Section 8.1. Afterwards, the recommendations for future research are presented in Section 8.2.

8.1. Conclusion

This section will provide the answers on the research questions posed in Section 1.3. The answers are substantiated by the research and associated results of this thesis.

RQ-1 *How can visual-Lidar data be used for multimodal feature detection?*

Instinctively, the visual-Lidar approach provides more data for feature estimation. However, next to the MRF method proposed earlier, this is is the second method which uses multisensor data fusion at feature level for spacecraft relative navigation with a uncooperative unknown target. The multimodal feature detection method proposed can complement the earlier methods and applied when considered more appropriate, e.g. when a target consist mainly of planar features and accurate estimations of the Lidar noise are known. Finally, it should be noted that the multimodal method does provide more availability of measurement, however it has not been researched at what cost.

RQ-1.a *Next to the Markov Random Field, are there other ways to fuse visual-Lidar data to benefit 3D feature detection?*

To answer this research question, research was done for multimodal feature detection methods which made no use of the MRF model. In this work, three different methods have been considered, of which some were inspired by previous research and some less so. Two methods proved to be infeasible, which were based on region growing for plane and planar edge detection, and visual edge and Harris corner detection. The method based on region growing proved to be too computationally expensive for spacecraft relative navigation, due to multiple nearest neighbour searches during each time step. The method using Canny edge detection and Harris corner detection proved to be largely unstable, especially considering complex targets and adverse lighting conditions.

Eventually, a multimodal feature detection method has been proposed which detects a 3D plane in the point cloud to project 2D features on, such that the 3D location of the feature can be determined. This method allowed robust estimation of the 3D position of the specific feature, since the high resolution visual camera provided accurate 2D positioning and a near optimal plane was fitted on low noise point cloud data using a RANSAC method. However, the plane detection method did not meet expectations for noisy point cloud data. Although, the method did provide a straightforward method to filter the background, since only the 2D features which lie within a detected plane are accepted.

RQ-1.b *How can the proposed 3D feature detection method be verified?*

Usually, proposed computer vision methods are verified through analysis of an end-to-end process. For this particular research, the relative pose estimation of an uncooperative unknown target is a natural choice for the end-to-end process. However, an additional method to verify the proposed method has been developed. The method included simulation of a development target with clear visual (Harris corners) and planar features. This method gave valuable insight into the proposed 3D feature estimation method. Although, the method could not be applied to relative pose estimation since the method makes use of SURF features of which the 2D ground truth cannot be determined directly.

RQ-2 *How can the fused visual-Lidar data be used for relative pose estimation?*

The multimodal feature detection algorithm provided the 3D location of the features, which could be tracked from one frame to the other. Additionally, the multimodal feature detection determined the plane equation of one or more planes on each frame, however the application of this information has not yet been researched. It is presumed that this information can be used during eclipse situations where there are no visual 2D features available.

RQ-2.a *How should the 3D feature derived from visual-Lidar fusion be used for relative pose estimation?*

A relative pose estimation algorithm has been suggested, based on the FastSLAM algorithm, which has the capability to estimate the pose and feature locations of an uncooperative unknown target. Some changes were required to the observation model of the EKF to accommodate the 3D features found by multimodal feature detection. After the changes, the FastSLAM algorithm proved to be useful for the estimation of the pose and mapping of the features. However, the biggest shortcoming is the particle degeneracy problem of the current filter. This causes the jagged estimation of the pose and can cause the filter to drift over time if the feature locations are mapped inaccurately.

RQ-2.b *What are the benefits of visual-Lidar multisensor data fusion on relative pose estimation?*

Although recognised in earlier research, the addition of a Lidar can provide the information required to determine the scale of the target [4]. However, when using the propose method, the scale does not have to be determined using a separate estimation process, such as done earlier using the MRF for multimodal feature detection [9].

It was concluded that due to the visual-Lidar data fusion improved feature rejections could be identified since an extra dimension was available for feature rejection. Additionally, even though the information was not used in the proposed algorithm, the plane detection in the feature detection step provided that normal of the detected plane. This information on the normal of the detected plane can be used to refine the estimation of the attitude of the target.

From a more holistic perspective of relative pose estimation, the multimodal approach can be considered the main pose estimation method, which is complemented by separate unimodal information when required. For instance in the case of an eclipse or a sensor failure. However, this should be researched in more detail to verify if it is beneficial.

8.2. Recommendations

As with any other research, the process and results can be improved, the ones specific for this research have already been treated in Section 7.4. According to the experiences of the author, recommendations for potential future research topics are presented in this section.

The recommendations are presented according to their associated subject. Some topics are already actively researched in literature, such as the estimation of the targets moments of inertia ratio, FPGA programming, or even GPU programming, of the algorithm for significant computational improvements. These knowledge gaps are already established throughout the astronautical community and will therefore not be mentioned in the recommendations that follow.

Simulation

- **Research the distribution of shapes and forms of space debris.**
  Next to the number of known debris objects situated in the space environment, how do other (damaged) pieces of space debris look like? Perhaps planar targets are not common at all. During the literature study, information on this topic could not be found, currently only the cross section is known for space debris other than intact defunct satellites. If this is known, accurate simulations of the target can be made, to benefit future ADR missions.
- **Physically-based ray tracing, specifically for modalities other than the visual spectrum.**
  Throughout the thesis, there has been no publicly available physically-based ray tracing for Lidars. This shortcoming in the current available software decreases the opportunities to research Lidar and other sensors in more detail, especially situated in the space environment.

Multimodal Feature Estimation

- **Research other combinations of sensors for multimodal spacecraft relative pose estimation.**
  For instance an IR camera which captures the scene which is illuminated by the Lidar. This could enable use throughout the entire orbit, even in eclipse. Next to the fact that an IR camera is more resistant to adverse illumination conditions.
- **For the current proposed method, the capability to work with other geometric primitives could be researched, such as cylinders or spheres.**
  Currently the proposed method can only accommodate targets with planar features. To increase the encapsulation of the proposed method, the method can be adjusted to other geometric primitives or more complex geometric shapes, such as targets wrapped in Kapton foil.

Relative Pose Estimation Filter

- **Moving window Kalman filter in the context of spacecraft relative pose estimation.**
  It would be interesting to see how a moving window Kalman filter would perform on this pose estimation problem, especially in combination with a particle filter. The moving window Kalman filter is a combination of a batch and sequential filter, which propagates through time. Using such a method, multiple measurements from multiple frames can be related, rather than single frame to frame measurement updates, which was used in this proposed relative pose estimation.
- **Attitude estimation based on detected plane parameters.**
  Currently, there is not yet a method to estimate the attitude of a target based on plane parameters only. This could for instance be beneficial for operation during eclipse. The challenge is to relate the different planes to the correct plane. An ambiguity is apparent for symmetrical targets with plane angles close to the plane detection update rate, i.e. angles between planes of 45° and a plane detection update rate of $\sim 45°$.

Experiment

- **Full relative navigation pipeline with the Franka Emika Panda robotic arm.**
  Now that the Franka Emika Panda robotic arm is available at the faculty of Aerospace Engineering at Delft University of Technology, the robotic arm can be more actively integrated into relative navigation research. Especially fully time-synchronised hardware-in-the-loop experiments using the Franka Emika Panda robotic arm could be interesting to research.
- **Research if multimodal feature detection can increase accuracy of intrinsic calibration.** Specifically using multisensor data fusion, it could be researched if the capability of accurately determining 3D features may allow for better intrinsic calibration of a camera using a lens. Since the scale, or the baseline, of an external object can be determined directly through the Lidar measurements. The same holds for the opposite, intrinsic calibration of a Lidar based on a visual feature detection.

A. Sensors used in Spacecraft Relative Navigation

Sensors are essential for relative navigation as they allow spacecraft to detect their surroundings. In general, a sensor converts an external input to an electrical signal, which is usually digitized for further use. For the use in spacecraft GNC a wide variety of sensors are considered. Although, as described in Section 1.4, this thesis will only consider visual cameras and Lidars in the context of spacecraft relative navigation. This consideration will be substantiated in the first section, Section A.1, where the sensor modalities will be discussed. Afterwards, visual cameras and Lidars will be covered in Section A.2 and Section A.3, respectively. It should be mentioned that most material of this appendix originates from the literature study.

A.1. Sensor Modalities

With respect to sensors, the sensor modality describes the external input which the sensor is able to sense. To make accurate conclusions of an environment, it is always important to understand the sensor modality in a system. For the use in spacecraft relative navigation specifically sensors that can detect electromagnetic radiation are considered. Therefore, first the basics of electromagnetic radiation will be presented, which is followed by a discussion on Electro-Optical (EO) sensors.

Electromagnetic Radiation

Electromagnetic radiation owes its name to the fact that electricity and magnetism are intimately related [45]. A change in an electric field causes a change in a magnetic field and vice versa. Electromagnetism are modelled as waves which travel at the speed of light[1] and can be distinguished by different wavelengths (λ). These wavelengths may vary a multitude of magnitudes across the entire spectrum, ranging from short gamma rays to long radio waves. A region of the electromagnetic spectrum with similar wavelengths is referred to as the waveband. The electromagnetic spectrum is depicted in Figure A.1, with a subdivision of the visible waveband.

[1] It is important to note that the speed of light is different for different types of materials, which is by definition strictly slower than the speed of light in a vacuum.

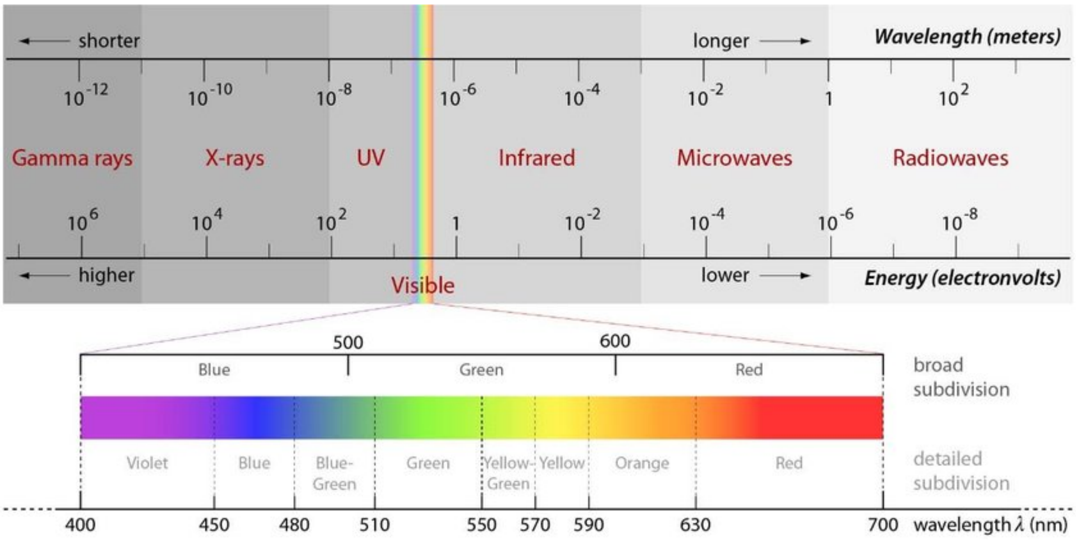


Figure A.1.: The complete electromagnetic spectrum with the spectral subdivision of the visible waveband [45].

In addition to the wavelike properties, the electromagnetic shows particle-like behaviour with different energy levels, which is considered to be part of the discipline of quantum mechanics. This concept can be interpreted as electromagnetic waves to consist of a travelling bundle of indivisible particles, or photons. These photons act as particles and have different energy levels, which vary according to the wavelength. This relation is called the Planck relation and is given in the formula below:

$$E = h \cdot v = \frac{h \cdot c}{\lambda} \tag{A.1}$$

From the first part of this relation it is apparent that the photon energy (E) is directly related to the frequency (v) through Planck's constant (h), which is equal to $6.626 \cdot 10^{-34} J \cdot s$. This in turn is inversely related to the wavelength, since the frequency is defined as the number of wavelengths in the length travelled in a single second at the speed of light, or $v = \frac{c}{\lambda}$ in formula form.

These photons that form the electromagnetic radiation can originate from a variety of sources, either natural, such as the Sun or radioactive materials, or artificial, such as Light-Emitting Diode (LED) or lasers. Moreover, almost all objects emit heat in the form of radiation. Afterwards, these photons can interact in multiple ways with different particles, which is related to the difference between the wavelength and the interacting particle's size [46]. The electromagnetic radiation can be either transmitted, reflected, absorbed, refracted, polarized, diffracted, and scattered[2].

Most sensors produce an electric signal by absorbing the electromagnetic radiation on their respective detectors. Since the detectors are made of a certain material with a certain particle, or molecule, size, the detector is only able to absorb specific wavelengths, and in turn, detect only specific wavelengths. The capability of a sensor to detect a certain wavelength is described by the spectral response of each of its detectors. Thus, the spectral response of a sensor represents the different modalities of the electromagnetic spectrum the detectors can sense. An example of such a spectral response of a sensor is depicted in Figure 6.8.

[2] A more thorough explanation of all the different interactions is considered out of scope for this thesis, a more practical overview will be given.

Electro-Optical Sensors

A multitude of different sensors can be used to sense an extensive part of the entire electromagnetic spectrum. Like each different sensor can sense a different waveband, each different waveband also provides different information, of which some are considered to be useful for spaceflight applications. For instance, within the scope of relative pose estimation with a uncooperative unknown target, only sensors with a spectral response in the Electro-Optical (EO) or radiowave region are considered to be useful [4]. During the literature study it was identified that sensors in the radiowave region were less useful for close range relative navigation compared to EO sensor, due to the lower temporal and spatial resolution of the sensors [47].

The EO region encompasses the UltraViolet (UV) region, the visible region and InfraRed (IR) region of the electromagnetic spectrum, ranging from a wavelength of roughly 0.01 μm to 1000 μm [45]. As stated earlier, different information can be discerned by sensing different wavelengths. Accordingly, an allocation of the utilization of different EO sensors is shown below.

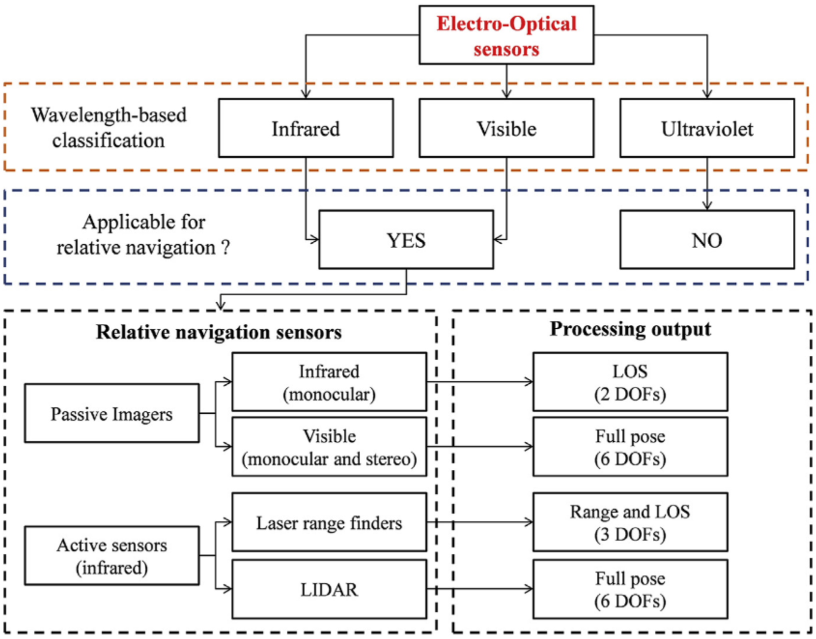


Figure A.2.: Taxonomy of EO sensors for spacecraft relative navigation [4].

From literature, it appears that EO sensors in the UV region are not applied in spacecraft relative navigation, and are more often applied for astronomical observations [48]. Whereas EO sensors in both the visible and IR region are considered throughout the community. It can be seen that both the passive visible and active Lidar provide the capability to determine the 6 DOF pose of the target[3], and are therefore considered to be best suited for this thesis on multisensor data fusion. Lastly, it should be noted that other GNC aspects such as own-pose estimation can be done using EO sensors, such as horizon sensors (visual camera), star trackers (visual camera) and SLR (similarities with Lidar). However, this aspect of GNC for relative pose estimation is concerned to be out of scope for this thesis and therefore these sensors are omitted from the overview that follows.

[3] It should be noted that a single monocular visual camera is not able to retrieve the full 6 DOF pose of the target if the target is unknown, since the scale and/or the range of the target cannot be directly measured.

A.2. Visual Camera

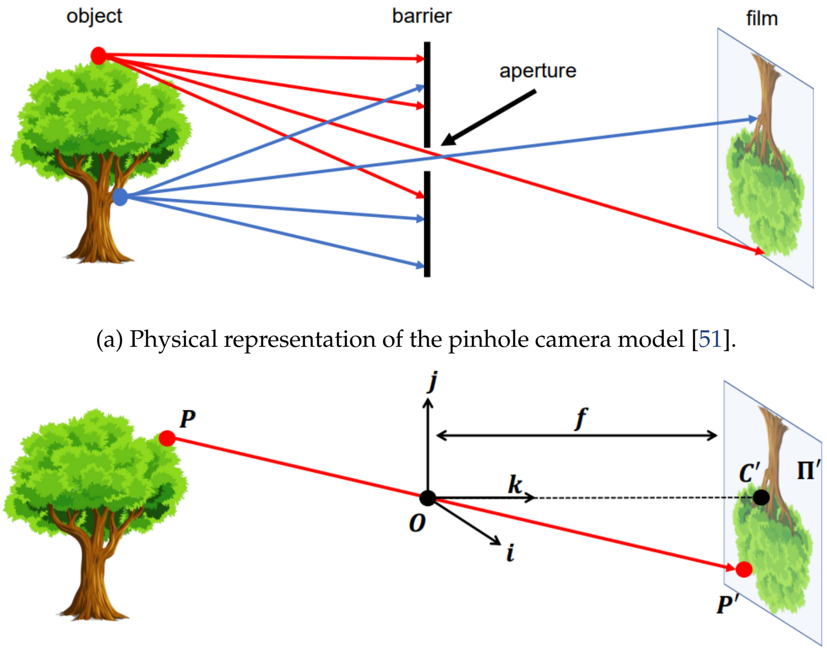
The visual camera is able to capture and record photons with wavelengths in the visible region of the electromagnetic spectrum. The visible region is considered to be the waveband which the human eye is able to detect, which ranges from about 400 nm to about 700 nm, which is commonly referred to as visible light, as was depicted in Figure A.1. These photons, or light waves, are sensed by a detector array or light sensitive chip, which is the main component of the visual camera. The detector array consists of multiple elementary sensing devices, also referred to as detectors, which record the photons reaching the elements in a certain time interval, also referred to as the integration time [49]. This detector array is divided in smaller regions, usually organised in a rectangular pattern, which correspond to the pixels of an image.

Currently, the two most used detector types are Charge-Coupled Device (CCD) or Complementary Metal-Oxide Semiconductors (CMOS). Currently, nearly all commercial visible light cameras produce full color images, which requires three different primary colors to be detected by individual detectors, which require a spectral response for red, green and blue light for the visible spectrum. A more in depth review of the visible light camera detector systems is considered out of the scope of this thesis, more in-depth information is available in literature [50, 49].

Visual Camera Models

To describe the functioning of a camera, camera models have been developed which describe how the 3D rays of visible light are projected on a 2D detector array, resulting in a 2D image. In this overview, two particular models will be discussed, namely the pinhole camera model and the thin lens camera model. These models are used to make the correct interpretation of the origin of the ray of light which is detected by which detector element, which is not as straightforward as one might think.

The first camera model is the pinhole camera model, which only allows visible light that passes through a small aperture on the detector array. In this case, the origin of the detected visible light can be directly interpreted, as is visualised in the figure below, Figure A.3.



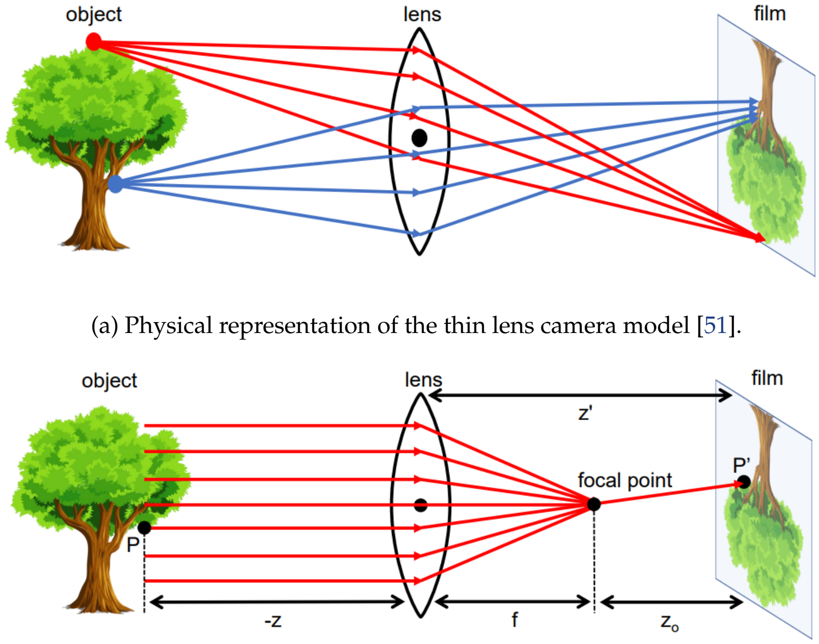
(a) Physical representation of the pinhole camera model [51].

(b) Modelled representation of the pinhole camera model [51].

Figure A.3.: Different representations of the pinhole camera model.

As can be seen in Figure A.3a, most of the incoming light is occluded by the barrier from the detector array, or photographic film. Therefore, only small amounts of visible light reach the detector array, which lead to a higher required integration time of the detector to create a clear 2D image. However, a higher integration time is considered to be detrimental when considering relative motion, as will be discussed later in this section.

The thin lens camera model makes use of a lens that directs the incoming visible light to the detector array. When the parameters of the lens are known, the origin of the detected visible light can be determined, with the exception of the depth. As opposed to the pinhole camera model, the aperture of a lens camera model is significantly larger, allowing more light on the detector array. This is shown in the figure, Figure A.4, below.



(a) Physical representation of the thin lens camera model [51].

(b) Modelled representation of the thin lens camera model [51].

Figure A.4.: Different representations of the thin lens camera model.

External Conditions for Visual Camera

With respect to the performance of the visual camera, there are two main external factors that should be taken into account. Which are the external illumination conditions and the relative motion of the target. Both factors can individually lead to serious performance issues for the visual camera recording and subsequent processes.

Firstly, due to the fact that most visual cameras depend on passive illumination of the scene, the illumination conditions are critical. Especially considering adverse illumination conditions, which can cause over exposure of the detector array, for instance due to direct illumination of the aperture from the sun, or under lighting of the detector array, for instance during eclipses, for both of which visual cameras are particularly vulnerable [41].

Secondly, the relative motion of the target can affect the performance of the visual camera in several ways, with respect to the position and the velocity. When considering the position, just like any other sensor, if the target is located further away from the visual camera, the number of pixels describing the target will be less. With respect to the velocity (both translational and rotational), the integration time of the detector array is crucial, since a higher velocity will lead to significant distortions with high integration times. For this reason, a pinhole camera is generally considered to be impractical due to the high required integration time. The translational velocity with respect to the target can be controlled, however the rotational velocity of the target might by too high to correct.

Next to the type of camera model used, the type of shutter for the detector array is a factor as well. In the case of a rolling shutter, the detector array is scanned over a period of time. This delay can cause a rolling shutter effect, an effect commonly seen with recordings of fast moving scenes using commercial cameras, for instance apparent in mobile phone cameras. In contrast, a global shutter evaluates all pixels instantaneously, negating such effects.

A.3. Lidar

Lidar is an acronym for LIght Detection And Ranging, similar to other acronyms such as sonar (SOund Navigation And Ranging) and radar (RAdio Detection And Ranging). However, other acronyms are used throughout other literature, such as LADAR (LAser Detection And Ranging). Lidars can primarily be distinguished by two different factors, their ranging technology and imaging technology, but before these are discussed, a general overview is given.

The Lidar is an active electro-optical sensing instrument, which makes use of a beam of electromagnetic radiation and uses the reflection to determine some properties of the material that caused the reflection. These materials can reside in a solid structure, such as a satellite, or even in a diffuse material, such as the Earth's atmosphere. The properties that can be measured are quite broad, such as distance and velocity, but also chemical contents of a scanned vapor [52].

Lidar Ranging Technology

The different Lidar ranging technologies can be divided into two broad principles. Either those that measure the range by measuring the Time-Of-Flight (TOF) of a ray of light, or those that measure the range by measuring the angle between two perspectives [53]. These different ranging principles have distinct characteristics, making them useful for particular applications. However, English et al. suggest that even though these principles can be seen as competitive, they are better of seen as complementary [53]. The following different ranging principles can be used to provide the range measurement:

- Time-of-Flight
  - Pulse
  - Phase-shift/Amplitude Modulated Continuous Wave (AMCW)
  - Interferometry/Frequency Modulated Continuous Wave (FMCW)
- Triangulation

As can be seen in the list above, the TOF of the light can be measured using different modulations [54]. The modulation frequency in turn determines the range of the Lidar sensor. According to Glaser, the importance of the transmitted waveform can not be understated [47]. Therefore, careful considerations should be taken when making use of this type of ranging technology. For clarity, it is important to note that the modulated wavelength is completely different than the wavelength of the Lidar.

The principle of triangulation is to measure the angle between the projection of an active source, of which the origin is known. Since the base line and the angle between the laser and the projection are known, the depth can be easily extracted through geometry. In fact, stereo-vision systems use the same technique, however, the stereo-vision systems match features or pixels directly. The basic scheme for triangulation is depicted in Figure A.5 below:

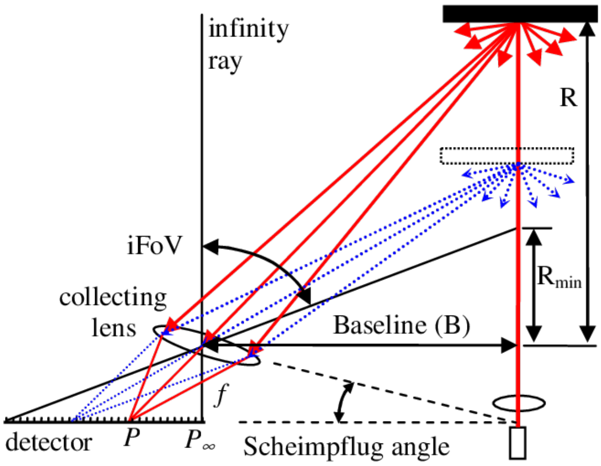


Figure A.5.: Basic geometry of triangulation [53].

Lidar Imaging Technology

With respect to the Lidar imaging technologies, these are similar to visual cameras, for instance using lenses to focus the incoming rays on detectors. However, the difference for Lidar is the required evaluation time to determine the range, dependent on the ranging technology used. In general, three distinct categories of Lidar sensors can be distinguished:

- Scanning systems
- Detector array systems
- Spatial light modulators

The former two categories have readily been applied for relative navigation aboard of flown spacecraft and experiments, whereas the latter is still under development and not available for a practical space application [55]. For this reason, spatial light modulators will not be considered in this thesis. Both the scanning systems and the detector array systems have been implemented with different ranging technologies, with some solutions being more practical than others. In the remainder of this section, four different Lidar sensors will be covered.

First, a scanning system referred to as the flying spot sensor. The sensor makes a single-point range measurement, scanning the entire FOV with a mechanical mirror. This has an advantage that only a single Lidar sensor needs to be calibrated, at the cost of risks of mechanical wear and possibly distorted measurements due to fast target moment. The pattern used for scanning can vary, ranging from simple raster scanning, to complicated patterns such as spirals, rosette or Lissajous patterns [56]. For this particular sensor, both TOF and triangulation can be used to determine the range.

Similar to flying spot Lidars, the line scanning Lidar scans a scene but with a line [53]. This techniques is considered in literature on radar as well, often described as “beaver-tail diagram” or “beaver-tail pattern”, since the fan beam is wide and flat, like the tail of a beaver[4]. This technique is considered to be more suited using triangulation range measurement.

The solid state Lidar makes use of a detector array to sample the incoming light. In general, the range is determined by measuring the TOF of the light by AMCW or FMCW. A high sampling rate is required to determine the TOF, typically at least 3 to 4 measurements per modulated wavelength are needed to provide an accurate 3D measurement.

Finally, the flash Lidar uses a detector array to image the incoming light. The flash Lidar exclusively uses TOF to determine the range, specifically by pulsed modulation of the emitted light, which is detected and timed by the a detector array. Providing an instantaneous 3D image of the scene.

[4] Taken from: https://www.radartutorial.eu/06.antennas/Fan%20beam%20antenna.en.html, accessed at: 05-06-2021.

External Conditions for Lidar

Similar to the visual camera, the same external conditions should be taken into account, namely the illumination conditions and the relative motion. However, especially the illumination conditions play a less significant role in the performance of the Lidar. Whereas, for the relative motion, the effect on the performance is largely dependent on the Lidar type, similar to the shutter type of the visual camera.

Generally, the modality of a Lidar is in the IR spectrum, of which the spectral irradiance from the Sun is significantly less (roughly a factor of 1.5 to 4, for conditions outside the atmosphere) compared to the visible spectrum. This is visualised in Figure A.6.

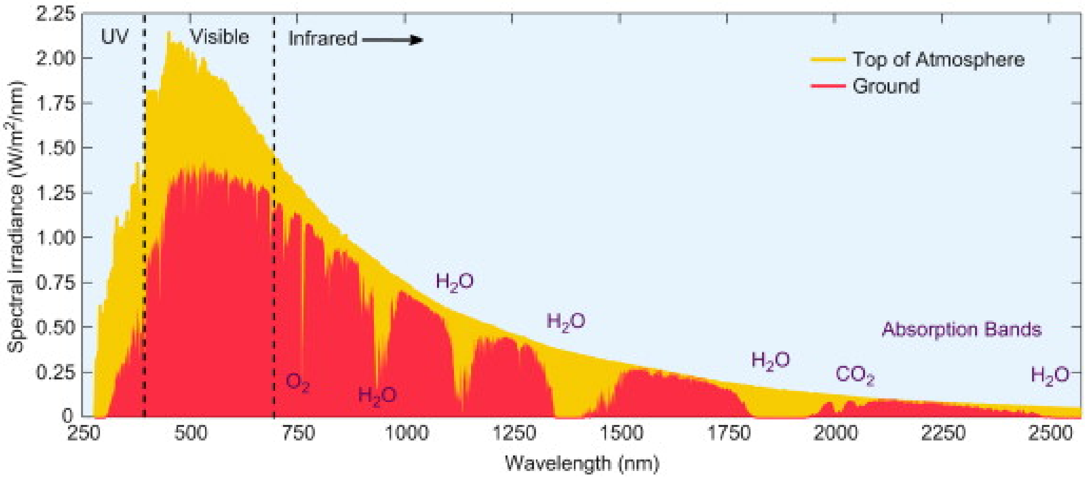


Figure A.6.: Solar irradiance at Earth as a function of the wavelength [57].

In addition, adverse illumination conditions are generally less detrimental for a Lidar since the Lidar actively illuminates the scene, in contrast to the visual camera. Therefore the SNR of the depth signal can be comparably higher, both during eclipse and direct illumination. However, some Lidars have a lower signal strength and can therefore only operate with indoor illumination conditions, such as the Intel Realsense L515, as discussed in Chapter 6. For this reason, for space missions, Lidars should only be considered if at least outdoor operation is allowed.

However, the susceptibility to relative motion largely depends on the imaging technology used. As can be imagined, the scanning Lidar shows, similarly to rolling shutter, distortion with respect to the movement of the target. Whereas the detector array largely negate this effect if a high enough modulation frequency is used and the incoming light is sampled uniformly across the detector array.

B. Algorithm Source Code

The development of this algorithm was shown in this thesis. The algorithm has been implemented in the MATLAB language, of which the code is available from the following GitHub repository: `https://github.com/tchstolk/MScThesis`

For this code MATLAB version 2021b was used, along with the following required MATLAB packages:

- Lidar Toolbox
- Image Toolbox
- Video and Image Blockset
- Navigation Toolbox
- Statistic Toolbox
- Machine Learning Toolbox

Bibliography

[1] Loretta Hall. The history of space debris. In *Space Traffic Management Conference*, 2014.

[2] Donald J Kessler and Burton G Cour-Palais. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637–2646, 1978.

[3] J-C Liou, Nicholas L Johnson, and NM Hill. Controlling the growth of future leo debris populations with active debris removal. *Acta Astronautica*, 66(5-6):648–653, 2010.

[4] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. *Progress in Aerospace Sciences*, 93:53–72, 2017.

[5] Wigbert Fehse. *Automated rendezvous and docking of spacecraft*, volume 16. Cambridge university press, 2003.

[6] Bo J Naasz and Michael C Moreau. Autonomous rpod technology challenges for the coming decade. *Advances in the Astronautical Sciences*, 144:403–425, 2012.

[7] Hugh Durrant-Whyte and Thomas C Henderson. Multisensor data fusion. *Springer handbook of robotics*, pages 867–896, 2016.

[8] Renato Volpe, Giovanni B Palmerini, and M Sabatini. Monocular and lidar based determination of shape, relative attitude and position of a non-cooperative unknown satellite. In *Proceedings of the 68th International Astronautical Congress*, volume 1, 2017.

[9] Gangtao Hao, Xiaoping Du, Hang Chen, Jianjun Song, and Tengfei Gao. Scale-unambiguous relative pose estimation of space uncooperative targets based on the fusion of three-dimensional time-of-flight camera and monocular camera. *Optical Engineering*, 54(5):053112, 2015.

[10] Florian Rems and Heike Frei. Autonomous Rendezvous GNC for On-Orbit Servicing: How small can it be? TU Delft Microsat Engineering Lecture Slides, 2022. Accessed: 12-04-2022.

[11] Marco D'Errico. *Distributed space missions for earth system monitoring*, volume 31. Springer Science & Business Media, 2012.

[12] NASA. On-orbit satellite servicing study project report. Technical report, Goddard Space Flight Center, 2010.

[13] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in aerospace sciences*, 68:1–26, 2014.

[14] Alex Ellery, Joerg Kreisel, and Bernd Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(5-6):632–648, 2008.

[15] Minghe Shan, Jian Guo, and Eberhard Gill. Review and comparison of active space debris capturing and removal methods. *Progress in Aerospace Sciences*, 80:18–32, 2016.

[16] Alan N Steinberg and Christopher L Bowman. Revisions to the JDL data fusion model. In *Handbook of multisensor data fusion*, pages 65–88. CRC press, 2017.

[17] David L Hall and Amulya K Garga. Pitfalls in data fusion (and how to avoid them). In *Proceedings of the Second International Conference on Information Fusion (Fusion'99)*, volume 1, pages 429–436, 1999.

[18] Tristan Tzschichholz. *Relative pose estimation of known rigid objects using a novel approach to high-level PMD-/CCD-sensor data fusion with regard to applications in space*. PhD thesis, Universität Würzburg, 2014.

[19] Tristan Tzschichholz, Toralf Boge, and Klaus Schilling. Relative pose estimation of satellites using PMD-/CCD-sensor data fusion. *Acta Astronautica*, 109:25–33, 2015.

[20] Jianqing Peng, Wenfu Xu, Bin Liang, and Ai-Guo Wu. Pose measurement and motion estimation of space non-cooperative targets based on laser radar and stereo-vision fusion. *IEEE Sensors Journal*, 19(8):3008–3019, 2018.

[21] Kangkan Wang, Guofeng Zhang, and Hujun Bao. Robust 3d reconstruction with an RGB-D camera. *IEEE Transactions on Image Processing*, 23(11):4893–4906, 2014.

[22] Weiguo Zhou, Yunhui Liu, Congyi Lyu, Weihua Zhou, Jianqing Peng, Ruijia Yang, and Haiyang Shang. Real-time implementation of panoramic mosaic camera based on FPGA. In *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 204–209. IEEE, 2016.

[23] Vincenzo Capuano, Kyunam Kim, Alexei Harvard, and Soon-Jo Chung. Monocular-based pose determination of uncooperative space objects. *Acta Astronautica*, 166:493–506, 2020.

[24] Zeming Jin, Ling Wang, Hanhan Liu, Ronghua Du, and Xiang Zhang. Monocular-based pose estimation of non-cooperative space targets using EKF and EKPF. In *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 46–51. IEEE, 2020.

[25] Dylan Conway, Brent Macomber, Kurt A Cavalieri, and John L Junkins. Vision-based relative navigation filter for asteroid rendezvous. In *Proceedings of the AAS Guidance, Navigation, and Control Conference, Breckenridge, CO, Febuary*, 2014.

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[27] Renato Volpe, Marco Sabatini, and Giovanni B Palmerini. Pose and shape reconstruction of a non-cooperative spacecraft using camera and range measurements. *International Journal of Aerospace Engineering*, 2017, 2017.

[28] Andrea Vedaldi, Haibin Ling, and Stefano Soatto. Knowing a good feature when you see it: ground truth and methodology to evaluate local features for recognition. In *Computer Vision*, pages 27–49. Springer, 2010.

[29] ITU-R. Studio encoding parameters of digital television for standard 4: 3 and wide-screen 16: 9 aspect ratios, 2011. URL `https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-E.pdf`. Accessed at: 26 August 2022.

[30] M Muthugnanambika and S Padmavathi. Feature detection for color images using surf. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1–4. IEEE, 2017.

[31] Ebrahim Karami, Siva Prasad, and Mohamed Shehata. Image matching using sift, surf, brief and orb: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017.

[32] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156, 2000.

[33] Zhi Jin, Tammam Tillo, Wenbin Zou, Xia Li, and Eng Gee Lim. Depth image-based plane detection. *Big Data Analytics*, 3(1):1–18, 2018.

[34] Eduardo Vera, Djalma Lucio, Leandro AF Fernandes, and Luiz Velho. Hough transform for real-time plane detection in depth images. *Pattern Recognition Letters*, 103:8–15, 2018.

[35] Sean Augenstein and Stephen Rock. Simultaneous estimaton of target pose and 3-d shape using the fastslam algorithm. In *AIAA Guidance, Navigation, and Control Conference*, page 5782, 2009.

[36] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[37] Michael Gschwandtner, Roland Kwitt, Andreas Uhl, and Wolfgang Pree. Blensor: Blender sensor simulation toolbox. In *International Symposium on Visual Computing*, pages 199–208. Springer, 2011.

[38] *MERCURY2 USB3 Vision Cameras User Manual*. China Daheng Group, Inc. Beijing Image Vision Technology Branch, 2021.

[39] David Rijlaarsdam. Relative navigation for debris removal (RENDER). Master's thesis, Delft University of Technology, Delft, July 2021.

[40] Marti Vilella. Global grasp planning for on-orbit robotic manipulation. Master's thesis, Delft University of Technology, Delft, August 2021.

[41] Lorenzo Pasqualetto Cassinis, Robert Fonod, and Eberhard Gill. Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft. *Progress in Aerospace Sciences*, 110:100548, 2019.

[42] Tim Nachtergaele, David Rijlaarsdam, and Joshua Spaander. Design, construction, verification and validation of a LED solar simulator. Microsat Engineering, 2019.

[43] John O Woods and John A Christian. Glidar: an opengl-based, real-time, and open source 3d sensor simulator for testing computer vision algorithms. *Journal of Imaging*, 2(1):5, 2016.

[44] Abril Poó Gallardo, Hrishik Mishra, Alessandro Massimo Giordano, and Roberto Lampariello. Robust estimation of motion states for free-floating tumbling target capture. In *2019 IEEE Aerospace Conference*, pages 1–11. IEEE, 2019.

[45] Geert Verhoeven. The reflection of two fields: electromagnetic radiation and its role in (aerial) imaging. *AARGNEWS*, 55:13–18, 2017.

[46] Ginger Butcher. *Tour of the electromagnetic spectrum*. National Aeronautics and Space Administration, 2010.

[47] John Glaser. How GaN power transistors drive high-performance lidar: Generating ultrafast pulsed power with GaN FETs. *IEEE Power Electronics Magazine*, 4(1):25–35, 2017.

[48] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. Spaceborne lidar-based system for pose determination of uncooperative targets. In *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, pages 265–270. IEEE, 2014.

[49] Gerald C Holst. *Electro-optical imaging system performance*. JCD Publishing, 2008.

[50] Gerald C Holst. CMOS/CCD sensors and camera systems. *CMOS/CCD sensors and camera systems/Gerald C. Holst*, 2007.

[51] Kenji Hata and Silvio Savarese. CS231A Course Notes 1: Camera Models, 2022. URL https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf. Accessed at: 16 May 2022.

[52] Paul F McManamon. *Field Guide to Lidar*. SPIE Press, 2015.

[53] Chad English, Adam Deslauriers, and Iain Christie. The complementary nature of triangulation and ladar technologies. In *Laser Radar Technology and Applications X*, volume 5791, pages 29–41. International Society for Optics and Photonics, 2005.

[54] Sergi Foix, Guillem Alenya, and Carme Torras. Lock-in time-of-flight (tof) cameras: A survey. *IEEE Sensors Journal*, 11(9):1917–1926, 2011.

[55] John A Christian and Scott Cryan. A survey of lidar technology and its use in spacecraft relative navigation. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4641, 2013.

[56] Attila Kovács. Scanning strategies for imaging arrays. In *Millimeter and Submillimeter Detectors and Instrumentation for Astronomy IV*, volume 7020, page 702007. International Society for Optics and Photonics, 2008.

[57] Cutler J Cleveland and Christopher G Morris. *Handbook of energy: diagrams, charts, and tables*, volume 1. Newnes, 2013.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.