Analysis and optimization of PMD-sensor data for rendezvous applications in space

Dipl.-Ing. Ksenia Klionovska

Vollständiger Abdruck der bei der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktor-Ingenieur (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender der Promotionskommission:
Univ.-Prof. Dr.-Ing. Roger Förstner

Gutachter:
1. Gutachter: Univ.-Prof. Dr. rer. nat. Felix Huber
2. Gutachter: Assistant Prof. Dr. Simone D'Amico

Die Dissertation wurde am 05.04.2019 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 11.01.2020 angenommen. Die mündliche Prüfung fand am 17.06.2020 statt.

To my beloved mother Natalia, brother Kirill and my sweetheart husband Marco

Acknowledgment

I would like to thank Prof. Dr. Felix Huber for the great opportunity given to me to conduct research at German Aerospace Center within a group of On-Orbit Servicing and Autonomy. I am grateful to my supervisor Dr. Heike Benninghoff for her continuous guidance, support and assistance during my doctoral thesis. Her significant tips and advice helped me not only successfully complete my current work, but also will be valuable assets for my professional future.

Thanks to my colleagues Florian Rems and Eicke-Alexander Risse for their nice atmosphere at our office and contribution to my research throughout many discussions and advice. Eicke-Alexander Risse had a huge patience spending plenty of time with me at EPOS laboratory looking for the bugs, compiling software and carrying out lots of tests. A former colleague Dr. Jacopo Ventura greatly forced my research during the second phase of my PhD by inspiring me with his ideas about algorithms. I am also grateful to Anibal Lopez, who thoroughly tried to find the answers for the thousands of questions, which have arisen throughout my research.

I would also mention Fabio Bracci, Dr. Matteo Saveriano, Dr. Sofya Spiridonova and Dr. Gerardo Allende Alba, we have spent plenty of pleasant time during tee/coffee pauses at DLR.

Special thanks to my adorable family - mommy Natalia, brother Kirill and husband Marco - for their invaluably endless support throughout this thesis.

"Flying a spacecraft is very different than flying a plane. There is no true up or down and the dynamics of orbital flight make maneuvering to dock, or rendezvous, two spaceships very complex. I focused my research on solving the problems of speed and centrifugal energy which lead to an "orbital paradox" a situation in which a pilot who speeds up to catch another craft in a higher orbit will end up in an even higher orbit, traveling at a slower speed and watching the second craft fly off into the distance.

The solution to this paradox is counter intuitive, and required new orbital mechanics and procedures. Later, after joining the NASA astronaut corps, I spent time translating complex orbital mechanics into relatively simple flight plans for my colleagues they thanked me (with a mixture of respect and sarcasm) with the nickname Dr. Rendezvous."

Buzz Aldrin-Waterkeeper, Fall 2005

Abstract

In the last decade, people became more aware of the critical situation in space concerning the space debris. Nowadays, there are plenty of uncontrolled objects located on different orbits in the outer space. Some of them present the real hazards to the functioning satellites and to the International Space Station. The researches all over the world are working on optimal solutions for the space cleaning. There are projects which are directed to repair the satellites and extend their operational lifetime, or totally remove a no more useful space object from its orbit. Rendezvous is the one inevitable space operation all these tasks need. We are able to execute a space operation with a target satellite only if the servicer satellite approaches it. The goal of this thesis is to develop a visual navigation system with the Photonic Mixer Device (PMD) sensor for the close rendezvous phase with a non-cooperative target. The PMD sensor has never been used in space missions so far, but it has already been tested on the ground.

In order to achieve the goal, this thesis provides two different pipelines for the pose acquisition and for the pose tracking using the images from the PMD sensor. In this work we show that the use of the PMD's amplitude and depth images together brings a great contribution to the visual navigation system. The pose acquisition is required in order to initialize a pose (position and orientation) of the non-cooperative space object before the tracking takes place. This task is very difficult, because we have to deal only with the PMD images and the known 3D model of the target. We initialize the pose with the depth image and thereafter improve the obtained pose with the amplitude image. As soon as we have an initial estimation, the servicer starts to approach the target in a frame-to-frame mode. The pose for every frame is estimated by a fusion of the states, which are calculated with the developed techniques for the depth and the amplitude images correspondingly. This technique shows a stable tracking with low errors of the estimated pose even if there are some distortions in the depth image. This fact is very important for the close rendezvous phase, because any failures in Guidance, Navigation and Control (GNC) system can lead to an unpredictable behavior of the chaser, and in end effect, create more space debris.

The techniques presented in this thesis are tested with real images of the PMD sensor. The rendezvous simulations are executed and evaluated with a high accuracy hardware-in-the-loop simulator. The tests of the visual navigation with a PMD sensor in a closed loop show promising results. The servicer satellite can smoothly approach the non-cooperative target by only using the PMD sensor for relative navigation.

Zusammenfassung

Im letzten Jahrzehnt wurde den Menschen die kritische Situation im Weltraum bezüglich des Weltraumschrotts bewusster. Heutzutage gibt es viele unkontrollierte Objekte auf verschiedenen Umlaufbahnen im Weltraum. Einige von ihnen stellen eine wirkliche Gefahr für die sich im Betrieb befindlichen Satelliten und für die Internationale Raumstation dar. Die Forscher auf der ganzen Welt arbeiten an optimalen Lösungen für die Enfernung dieser kritischen Objekte aus dem Orbit. Es gibt Projekte, die dazu bestimmt sind, die Satelliten zu reparieren und ihre Lebensdauer zu verlängern oder ein Weltraumobjekt vollständig aus seiner Umlaufbahn zu entfernen. Unvermeidbar ist hierfür der Anflug eines Service-Satelliten an den Zielsatelliten. Das Ziel dieser Arbeit ist die Entwicklung eines visuellen Navigationssystems mit dem PMD-Sensor (Photonic Mixer Device) für die enge Rendezvousphase mit einem nicht kooperativen Ziel. Der PMD-Sensor wurde bisher noch nicht in Weltraummissionen eingesetzt, aber er wurde jedoch bereits am Boden getestet.

Um dieses Ziel zu erreichen, werden in dieser Arbeit zwei verschiedene Pipelines für die initiale Bestimmung der Position und der Orientierung von Objekten und deren Nachfolgung anhand der Bilder des PMD Sensors bereitgestellt. In dieser Arbeit zeigen wir, dass die Verwendung der Amplituden- und Tiefenbilder des PMD-Sensors einen großen Beitrag zum visuellen Navigationssystem leistet. Die erste Bestimmung der Position und Orientierung ist erforderlich, um eine Pose (Position und Orientierung) des nicht kooperativen Weltraumobjekts zu initialisieren, bevor die Verfolgung stattfindet. Diese Aufgabe ist sehr schwierig, da wir nur die PMD-Bilder und das bekannte 3D-Modell des Zielobjekts nutzen können. Wir initialisieren die Pose mit dem Tiefenbild und verbessern danach die erhaltene Pose mit dem Amplitudenbild. Sobald wir eine anfängliche Schätzung haben, nähert sich der Servicer dem Ziel in einem Frame-to-Frame-Modus. Die Pose für jedes Einzelbild wird durch eine Fusion der Zustände geschätzt, die mit den entwickelten Techniken für die Tiefen- und Amplitudenbilder entsprechend berechnet werden. Diese Technik zeigt eine stabile Verfolgung mit geringen Fehlern der geschätzten Pose, auch wenn das Tiefenbild einige Verzerrungen aufweist. Diese Tatsache ist für die enge Rendezvous-Phase sehr wichtig, da ein Ausfall des Guidance, Navigations und Steuerungssystems (GNC) zu einem unvorhersehbaren Verhalten des Verfolgers führen und letztendlich mehr Weltraumschrott erzeugen kann.

Die in dieser Arbeit vorgestellten Techniken werden mit realen Bildern des PMD-Sensors getestet. Die Rendezvous-Simulationen werden mit einem Hardware-in-the-Loop-Simulator mit hoher Genauigkeit durchgeführt und ausgewertet. Die Tests der visuellen Navigation mit einem PMD-Sensor in einer geschlossenen Regelschleife zeigen vielversprechende Ergebnisse. Der Service-Satellit kann sich dem Ziel ebenmäßig nähern, wobei lediglich der PMD-Sensor für die Relativnavigation verwendet wird.

List of Publications

- Klionovska, K., Benninghoff H. Visual Navigation for Rendezvous and Docking using PMD Camera. In Doctoral Consortium, In *2nd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM)*, Rome, Italy, pp. 3-7, 2016. ***Best Phd Project Award.***

- Klionovska, K., Benninghoff, H. Initial Pose Estimation using PMD Sensor during the Rendezvous Phase in On-Orbit Servicing Missions. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, USA, 2017.

- Klionovska, K., Benninghoff, H. and Strobl, Klaus H. PMD Camera-and Hand-Eye-Calibration for On-Orbit Servicing Test Scenarios On the Ground. In *14th Symposium on Advanced Space Technologies in Robotis and Automation (ASTRA)*, Leiden, the Netherlands, 2017.

- Klionovska, K., Ventura, J., Benninghoff, H., Huber, F. Close Range Tracking of an Uncooperative Target in a Sequence of PMD Images. *1st IAA Conference on Space Situation Awareness (ICSSA)*, Orlando, FL, USA, 2017. ***2nd Best Student Paper Award.***

- Klionovska, K., Ventura, J., Benninghoff, H., Huber, F. Close Range Tracking of an Uncooperative Target in a Sequence of Photonic Mixer Device (PMD) Images. *Robotics*, 7(1), 5, 2018.

- Klionovska, K., Benninghoff, H., Risse, E.-A., Huber, F. Experimental Analysis of Measurements Fusion for Pose Estimation Using PMD Sensor. *23nd Iberoamerican Congress, CIARP 2018*, Madrid, Spain, 2018.

- Klionovska, K., Benninghoff, H., Huber, F. More Accurate Pose Initialization with Redundant Measurements. *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Prague, Czech Republic, 2019.

Contents

List of Figures

List of Tables

Chapter 1

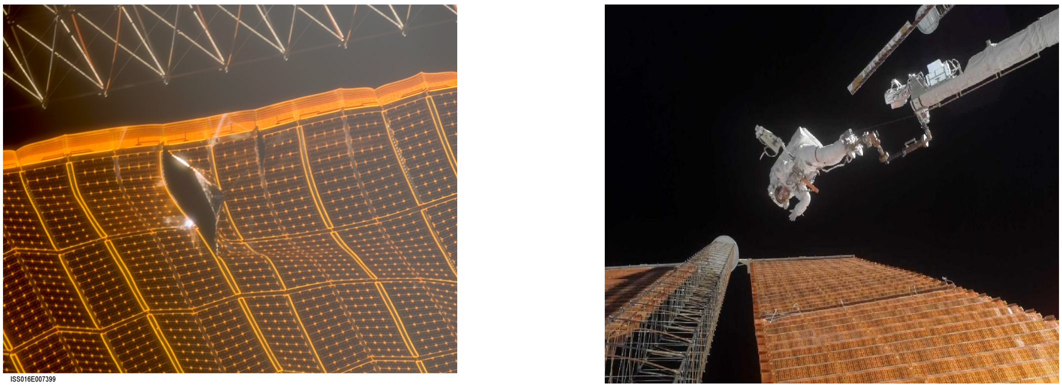
Introduction

1.1 On-orbit Servicing and Active Debris Removal

The mobile systems which are used on the Earth, e.g. airplanes, ships, cars, have the ability to get regularly external support and maintenance. Moreover, the periodic refueling and repairing and/or upgrading of the different parts of the complex systems is considered as the usual procedures. Since the asset of using the space platforms in space, such as International Space Station (ISS) and different types of satellites, is being increased yearly, the explicit demand for On-Orbit Servicing (OOS) technologies in space has become one of considerable issue. The impact of On-Orbit Servicing procedures leave the valuable trace in the technical, scientific and economical branches before and during the space operations [1]. The major OOS activities are planned to include the following goals [2]:

- refueling of propellant
- replenishment of the spacecraft's components (e.g. batteries or solar arrays)
- replacement of failed subsystems (e.g. payload and bus electronics)
- extension of the mission (e.g. software and hardware upgrades)

Up to now, the OOS activities have been provided by involving humans to terminate special tasks at ISS. For instance, more than a dozen of several failures, unexpected problems and different maintenance issues have been solved on the ISS by the astronauts since the existence of the station. One of the accidents happened in 2007, when a torn of a solar panel has been detected during the solar arrays deployment [3]. The mission's spacewalk has been organized and the astronaut successfully repaired the tear. But this spacewalk had been considered as one of the most dangerous, since there was a risk of shock from the electricity generated by the solar panels. Therefore, in cases with high complex tasks, where risk to the human's life occurs, autonomy is one of the possible alternatives. What will happen if a satellite at the Geostationary Earth orbit (GEO) has e.g. a broken solar panel? There are no possibilities to repair it by an astronaut. Consequently, one of the challenging issues of the OOS missions is to create fully autonomous systems. Firstly, they can assist astronauts, and take over dangerous tasks. Secondly, autonomous OOS via a service spacecraft is the only way to recover satellites located at different orbits. The autonomous system should independently gain information about the environment and also work in this environment without human or other engineering system interventions.



(a) (b)

Figure 1.1: (a): The tear in the solar array material of ISS. Image credit: see link in [4]. (b): Scott Parazynski during the process of repair the solar arrays. Image credit: see link in [5].

But if the above mentioned OOS activities no longer will be useful and only the proper disposal of the space object from its operational orbit is a proper solution? In this instance we deal with space debris, meaning that the problem is addressed in terms of remediation of active space debris. According to calculations provided by the ESA [6], more than 7500 tonnes of space hardware are located in space. Creating a practical Active Debris Removal (ADR) technology is extremely necessary in order to mitigate the risk of future collisions and growth of future debris. In particular, the situation on the Low Earth Orbit (LEO) looks crucially dangerous, since the density of the space debris there is measured in hundreds of thousands of fragments.

1.2 Space Rendezvous

Autonomous space rendezvous is an essential operation required in all the missions involving more than one spacecraft [7]. In general, the process of the rendezvous requires that two spacecrafts start at a remote distance (i.e., out of sight of each other), come together into a common orbit and rendezvous to each other [8]. If the mission envisages the joining of the vehicles, the process of docking and berthing occurs. The first rendezvous and docking between two spacecraft dates back on 16 March 1966, when a Gemini vehicle manually approached and then docked an unmanned Agena target vehicle. On 30 October 1967, the first automatic Rendezvous and Docking (RvD) took place between two Soviet space vehicles Cosmos 186 and Cosmos 188 [7]. Since that time the era of RvD operations has been started.

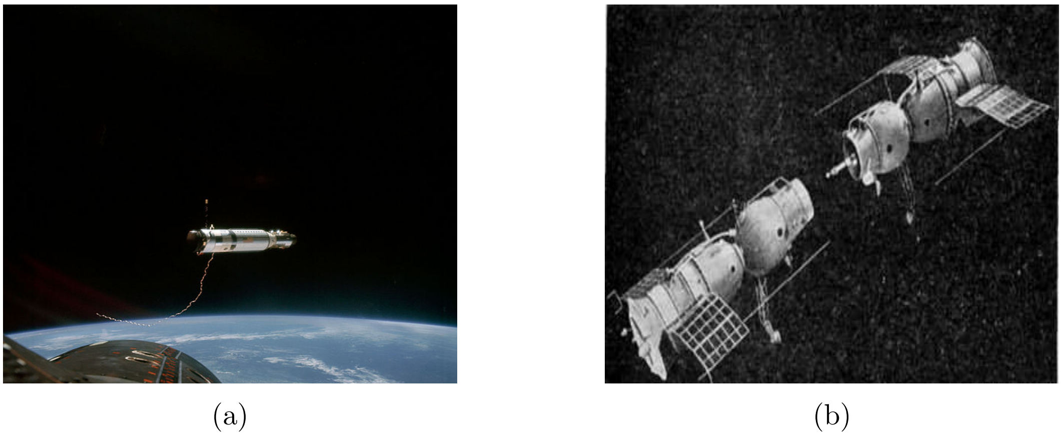

(a) (b)

Figure 1.2: (a): The Agena Target Vehicle as seen from Gemini 8 during rendezvous. Image credit: see link in [9]. (b): First automated rendezvous of Soviet space vehicles Cosmos 186 Cosmos 188. Image credit: see link in [10].

In the rendezvous space scenario between two spacecraft, usually, one space vehicle is active (chaser) and the other one is passive (target). The active spacecraft does the most of activities during the rendezvous phase, since the target space vehicle is completely passive and with a potentially uncontrolled motion. The rendezvous is a crucially important phase, since this process consists of the sequence of different translational and rotational manoeuvres. It starts at long distance apart between both space objects with multiple rendezvous stages in order to bring the active vehicle in the vicinity of the passive one.

One essential challenge appears when we plan to implement rendezvous with space debris - the uncooperativeness of the target spacecraft. It is not equipped with dedicated visual markers for distance and attitude estimation, it may uncontrollably tumble around the principal axes, and initially there is no sufficient information about the kinematics. Usually it is assumed to have some information about the 3D model of the target but no information about the target's pose (position and orientation). For this reason, new technologies, especially in the field of navigation, must be developed for the RvD to non-cooperative targets.

1.2.1 Phases of Rendezvous Mission

A rendezvous mission can be divided into different phases, which differ in type of activity, the relative distance between two spacecraft and also the source of navigation [8]. The main steps illustrated in Figure 1.3 are as follows: phasing, far range rendezvous, close range rendezvous and mating [7]. Each step of the mission presents its own mission with special initial and terminal condition [11].

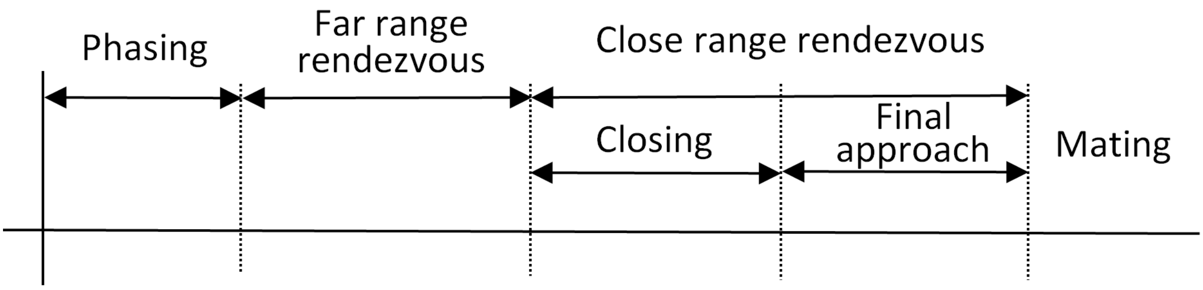


Figure 1.3: Main phases of rendezvous operation.

1.2.2 Precondition: Separate Orbits

Initially, we expect to know at least approximately the target's orbit. The approximate orbit can be determined by tracking methods from ground. It is assumed that the upper stage of the launch vehicle brings the servicer spacecraft into the stable orbit in the target's orbital plane [12]. The difference in the semi-major axes between two orbits creates a relative drift of the phase angle (Figure 1.4) between the spacecraft. In general, the chaser vehicle can be above or below the target, but the position of the chaser on the lower orbit is considered safer, since there is no need for the servicer to cross the target's orbit prior to terminal rendezvous.

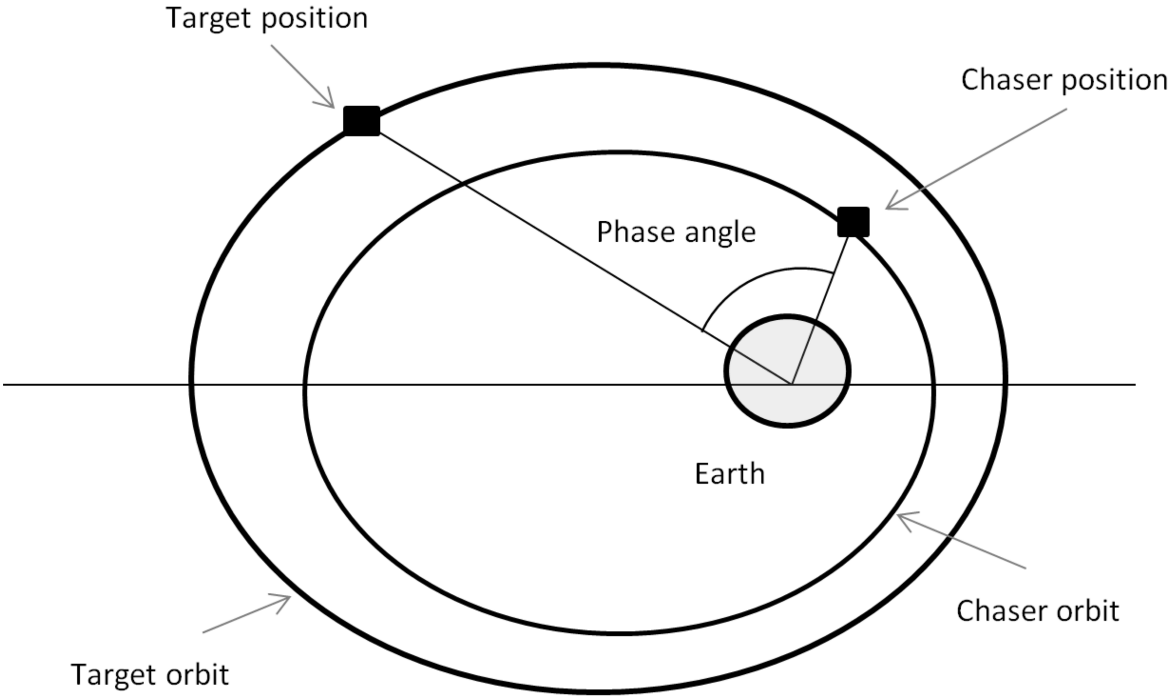


Figure 1.4: Orbits of the chaser and target with defined phase angle.

1.2.3 Phase I - Phasing

The next step is to reduce the phase angle between the target and chaser and reach an "aim point" or "entry gate". There are several phasing strategies, depending on the phase angle to the target, necessary correction parameters and limits of time presumed for the total mission. Trajectories and manoeuvres should be calculated individually for the every single launch, depending on the target spacecraft and goal of the mission. One can find more detailed description of different manoeuvres types and orbits in a work of Fehse [7].

Well planned consecutive phasing manoeuvres will be terminated by reaching the required initial aim point or entry gate for far range rendezvous. The aim point usually locates on the target orbit or somewhere very close to it. The major significant effects on the accuracy of the rendezvous trajectories vary with the orbit. For example, in LEO the strongest disturbance is drag due to the residual atmosphere. Moreover, during the phasing, the effect of the geopotential anomaly causes inaccuracies of the spacecraft's position. In GEO, the most serious disturbances occur due to the solar wind [7].

1.2.4 Phase II - Far Range Rendezvous

When the entry gate is reached, the proximity phase operation starts. The single manoeuvre in order to bring the chaser vehicle from the entry gate to the final point of the mission is hard to implement due to numerous factors: safety of the chaser spacecraft, fuel consumption, target detection and relative navigation [11]. Moreover, in situations where we deal with non-cooperative targets, the motion of the servicer can not be constrained to the orbital plane of the target and it should be capable to execute maneuvers out of this plane. Consequently, the chaser should be able to reach any region around the target spacecraft. The main goal of the far range rendezvous is determination of the target's orbit and estimation of its position. Synchronization of the mission timeline, reduction of the velocity and determination of the lightning conditions are the key tasks. The end point of this phase is defined as the "intermediate hold point" from which the final close approach can be started. Depending on the mission and type of the satellite, the total number of the hold points can vary during the far and close rendezvous. Taking into account the difficulty of the OOS missions, it is better to choose more hold points in order to check more often the working status of the chaser's navigation system. It ensures the safety of the system and leads to the successful completion of the mission.

One of the safe approach techniques based on relative eccentricity and inclination vector separation is described in works [13, 14]. During the approach phase from several kilometers to a few hundred meters, a relative orbit determination is based on angular measurements. The collision risk is measured as a function of separation in the plane orhtogonal to the flight direction and can be prevented in the presence of large uncertainties of angles-only navigation. The far range rendezvous to a non-cooperative object at GEO is addressed in [12]. Co - elliptic rendezvous profile with Hohmann raises brings the servicer satellite to the final co-elliptic drift segment. From that point starts the proximity operation to insert the servicer spacecraft to safety ellipse using Lambert targeting. This trajectory segment is designed to be safe, the servicer will not collide with a target if the ellipse insertion maneuver is missed or canceled.

1.2.5 Phase III - Close Range Rendezvous

The phase of the close approach can be divided into two sub-phases: 'closing' and a final approach, which brings the chaser in very close vicinity of the target until mating. In the closing phase, the main requirements are reduction of the relative distance to the target and achievement of the initial conditions for the final approach corridor. The close range approach is considered extremely challenging. In comparison with the previous phase, the accuracy of the navigation system during close proximity operation must be increased tremendously. The errors in the estimation of the relative position, attitude and velocity should be very small. Different visual sensors for the relative navigation in close range are used in order to guarantee a fault-tolerant navigation system and safe docking, without a collision with the target vehicle. Above all, some time-flexible elements in form of hold points may still be planned and executed for operational reasons. In case of unforeseen circumstances related to the target's or chaser's vehicle, the rendezvous process can be interrupted for some time. In that case, the trajectory design should foresee the return to one of the hold points with a further restart of the closing. In the second sub-phase, i.e. during the final approach, the goal is to achieve docking or berthing capture conditions in terms of relative position and orientation, velocities and angular rates. Additionally, the GNC system of the chaser needs to bring the chaser docking or berthing capture interfaces into reception range. Usually, the chaser approach trajectory is chosen as a closed loop straight line or quasi-straight line trajectory. A straight line approach has the advantage that only one parameter, the distance, is changed while the other ones (like azimuth and elevation angles) are kept constant.

1.2.6 Mating: Docking or Berthing

In the mating phase there is a list of necessary conditions, which must be achieved, before the ultimate docking or berthing takes place. For docking, the parameters, like the approach velocity, lateral and angular alignment, lateral and angular rates, are principal for the evaluation. Whereas for the berthing, the position and attitude accuracy, residual linear and angular rates are the major indicators. The difference between docking and berthing is as follows. During the docking process the target spacecraft is actively controlled until a hard dock with the target. Since only uncontrolled non-cooperative targets are considered, docking is no option in OOS. In case of berthing, some of capturing mechanisms (e.g. manipulator arm, nets, harpoons) located on the servicer satellite play the major role in order to attach the target and the chaser together.

1.3 Objective of the Thesis

Since two decades, the research in the field of ADR and OOS activities has been essentially led worldwide by some space agencies. Various investigations in the branch of proximity operations to non-cooperative space objects have been done actively. As one could notice from the previous Section 1.2, the process of space rendezvous is a demanding process in all space applications regardless of the nature of the target vehicle. Furthermore, the uncooperativeness of the target space object increases the degree of complexity of autonomous proximity operations.

In this thesis, the major focus for investigation is directed to the field of close range rendezvous, especially in the final phase approach. Navigation during the final approach is based on relative measurements (position and attitude) between the chaser and target due to the lack of any absolute navigation information by the target. Generally, different types of vision-based relative navigation systems are used for the important final phase of close range rendezvous. With a decrease of the relative distance between two space objects, there are requirements for a dramatical increase of navigation accuracy. According to this, the principal goal of this thesis is to design and implement robust pose estimation techniques during the final phase range ($<$ 10 meters) of the non-cooperative space target using computer vision solutions with a proposed hardware, namely a Photonic Mixer Device (PMD) sensor [15] [16]. The PMD sensor is a ranging devices, which provides a depth and amplitude image for every frame. Current research aims to find the initial pose with a depth image from PMD sensor and a 3D model of the target. With an acquired guess the pose tracker can be initialized and the servicer approaches the target using subsequent PMD images. The scope of this work relates to the space robotics for the ADR and OOS missions, where there is still a technological gap, particularly in the development of special GNC technologies in order to navigate to the non-cooperative target in the close vicinity. There is a great need to test new hardware and software technologies for RvD within different test campaigns on the ground.

1.4 State of the Art

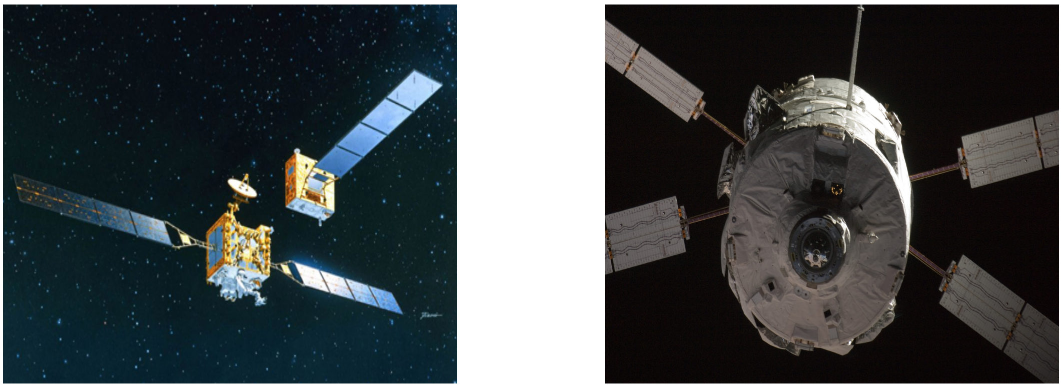
Up to this moment, the necessary tasks in the field of spacecraft servicing have been provided in a manned mode. However, the concept of fully autonomous inspection, rendezvous and docking with differently accurate visual-based sensors have been recently tested in several demonstration missions with a cooperative spacecraft, since its attitude was stabilized and it was equipped with dedicated visual markers. The Sections 1.4.1 and 1.4.2 provide a review of the existing navigation sensors and some missions where they have been tested. The Sections 1.4.3 - 1.4.5 give an overview of state-of-the-art pose estimation techniques.

1.4.1 OOS and ADR Missions

The concept of fully autonomous inspection, rendezvous and docking with differently accurate visual-based sensors have been recently tested in several demonstration missions with a cooperative spacecraft, since its attitude was stabilized and it was equipped with dedicated visual markers. The current and following sections provide a review of the existing navigation sensors and some missions where they have been tested.

The first unmanned spacecraft, called Engineering Test Satellite No. 7 (ETS-VII), to conduct autonomous rendezvous docking operations was developed and launched by the National Space Development Agency of Japan (NASDA) in 1997. Different types of sensors were used during autonomous RvD, such as laser, GPS and a proximity sensor. The proximity sensor consisted of a reflecting marker, which was located on the target, and emitting diodes placed on the chaser. The reflected light form the target was captured by the camera system of the chaser in order to calculate the pose between both space bodies.

The Automated Transfer Vehicle (ATV) [17], developed by the European Space Agency (ESA) has performed five supply missions to the International Space Station (ISS) in fully autonomous mode. The purpose of this cargo spacecraft was to transport various payloads in order to maintain the ISS, support astronaut's lifes and also conduct research experiments. ATV's navigation is based on cooperative rendezvous technology. At long range, the navigation to the station is based on the GPS measurements from the ATV and the ISS. At short range, navigation employs the optical sensors (videometers and telegoniometers) fixed on the ATV and the retro-reflectors which are located on the the Space Station. During the last ATV mission, the LIRIS demonstrator (Laser InfraRed Imaging Sensors) [18], was tested in in-flight experiments on a European vehicle in order to demonstrate the technology readiness for future uncooperative rendezvous in space. The LIRIS system was composed of Infra-red/visible cameras and LIDAR (LIght Detection And Ranging) sensor. This sensor provides the distance to an object by measuring the time between the transmitted and returned back light from a laser beam. After the processing the recorded data from the dedicated sensor, the experiments has been declared successful.

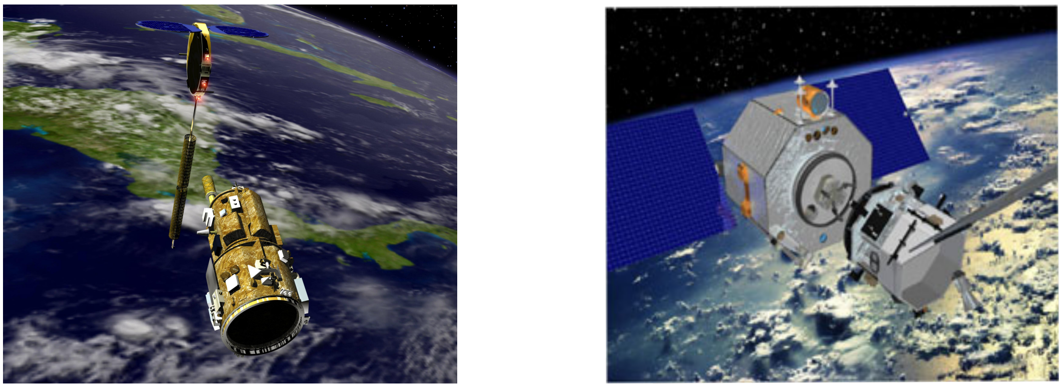


(a) (b)

Figure 1.5: (a): Engineering Test Satellite VII "KIKU-7" (ETS-VII). Image credit: see link in [19]. (b): The Jules Verne Automated Transfer Vehicle (ATV) approaches the International Space Station. Image credit: see link in [20].

In April 2005, NASA launched "Demonstration for Autonomous Rendezvous Technology" (DART) [21] with the goal to demonstrate fully automated navigation and rendezvous technologies. Series of proximity maneuvers have been planned in order to test and validate the "Advanced Video Guidance Sensor" (AVGS) [22]. The AVGS used laser diodes to illuminate retro-reflectors in the target, a solid-state camera to detect the returned signal from the target, image capture electronics and a digital signal processor. During the first 8 hours of the mission, DART operated as it had been planned, but thereafter some anomalies were detected with the navigation system. The mission could not be accomplished and, moreover, DART had collided with the MUBLCOM satellite.

The Orbital Express (OE) mission was started in 2007 by United States Defense Advanced Research Projects Agency (DARPA) and a team led by engineers at NASA's Marshall Space Flight Center. The goal was to validate the technical feasibility of robotic, autonomous on-orbit refueling and reconfiguration of satellites during a three-month mission. The Autonomous Rendezvous and Capture Sensor System (ARCSS) was part of the chaser's (ASTRO) GNC system and was tested during the on-orbit performances. The ARCSS sensors consisted of a set of two visible light cameras, an infrared camera, a laser finder and the Advanced Video Guidance Sensor of OE. The ARCSS and the reliable software provided precision real-time target (NEXTSat) bearing, range and attitude from the range 200 km to soft capture. The results of the ARCSS performance have been outstanding during a total of five scenarios.



(a) (b)

Figure 1.6: (a): Artistic impression of NASA's DART spacecraft approaching MUBLCOM. Image credit: see link in [23]. (b): Orbital Express: ASTRO and NEXTSat. Image credit: see link in [24].

In 2010 the Swedish National Space Board together with the German Aerospace Center (DLR), the French space agency (CNES), and the Danish Technical University have conducted a technology mission called Prototype Research Instruments and Space Mission technology Advancement (PRISMA) in order to demonstrate autonomous formation flying, rendezvous and proximity operations [25]. The Visual Based Sensor (VBS) was built as an extension module of μ - Advanced Stellar Compass (μASC), a fully autonomous CCD-based star tracker platform, which can contain up to four Camera Head Units. The VBS allowed to identify and track the target in far and close ranges. For the far range visual navigation, the standard star tracker was used. The close range autonomous approach deployed the set of the coplanar LED installed on the target spacecraft. It allowed to evaluate sensor performances by considering the target spacecraft as a cooperative, as well as a non-cooperative space object.

During the extended phase of the PRISMA mission, an experiment called Advanced Rendezvous Demonstration using Global Positioning System and Optical Navigation (ARGON) has been performed [14]. The aim of the experiment was to demonstrate far range approach to a non-cooperative passive target using angles-only optical navigation. The efficient and reliable rendezvous from 30 km to 3 km was accomplished, even though some difficulties for the range determination inherent to the angles-only navigation occurred.

In November 2016 the Vision Approach Navigation and Target Identification (AVANTI) experiment was successfully conducted [26] on the BIROS mission, where the goal was the realization of a fully autonomous close approach of an unmanned spacecraft to a non-cooperative target in low Earth orbit. The active servicing satellite used the μASC star-tracker. It hosted two cameras heads and two cold redundant digital processing units. The mission demonstrated a strong achievement in the field of non-cooperative autonomous approach with less than 50 m by using only the monocular camera for the estimation of the relative state.

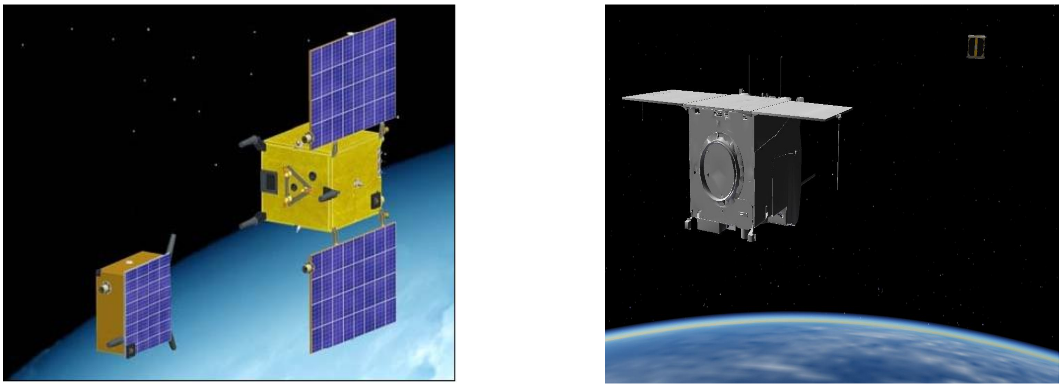

(a) (b)

Figure 1.7: (a): Artistic illustration of the PRISMA launch configuration. Image credit: see link in [27]. (b): AVANTI experiment: BIROS satellite and BEESAT-4 pico-satellite. Image credit: see link in [28].

In the recent past, the Northrop Grumman's Mission Extension Vehicle-1 [29] for the first time performed a docking with a non prepared for dockng satellite. It gave to the satellite five more years of service. Nowadays, there are some new projects, which are still in development and are scheduled to be launched in the near future. A robotic spacecraft OSAM-1 [30], which is able to extend the operational lifespan of the satellite, even if that was not prepared for a such mission. The e.Deorbit mission [31] of the the European Space Agency aims to capture and burn up the ESA-owned satellite (ENVISAT) from the low orbit. The Orbital ATK [32] offers a satellite life extension vehicle that must dock with a satellite and take over maneuvering, relocate the space object or dispose it into graveyard orbit. The Robotic Servicing of Geosynchronous Satellites (RSGS) is proposed by DARPA [33] for demonstration of the different satellite servicing mission operations in or near GEO.

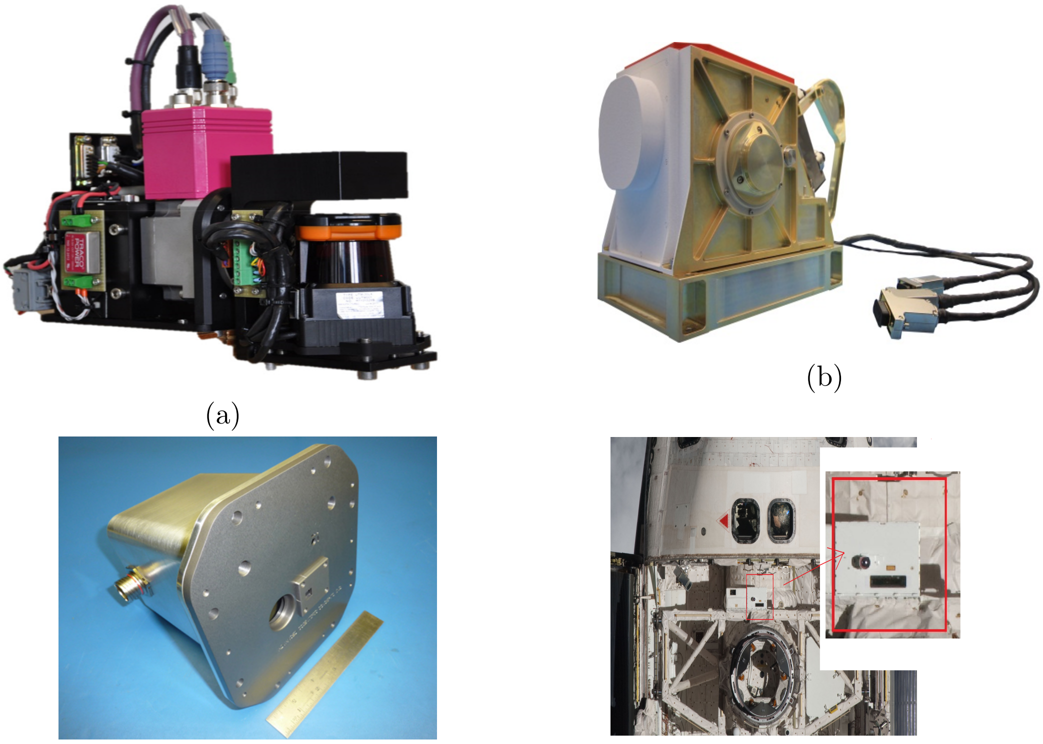
1.4.2 Visual Sensors

As I discussed in Section 1.2.5, due to the operational range of the visual sensors, the autonomous rendezvous usually includes multiple types of systems, e.g. sensors and appropriate algorithms. Some of the rendezvous sensors have already been briefly mentioned in Section 1.4.1, which have been used during space missions. Now, some more detailed information about them is provided.

At the moment, two main groups of optical technologies are taken into operation for RvD missions [34]. They would be categorized as passive and active sensors. The first one contains cameras (monocular- or stereo- cameras) and the second includes some types of LIght Detection And Ranging (LIDAR) sensors (e.g. scanning LIDAR or Flash LIDAR).

The measurement principle of the LIDAR is based on the illumination of the target with a pulsed light and measuring the reflected pulses from the object with a sensor.

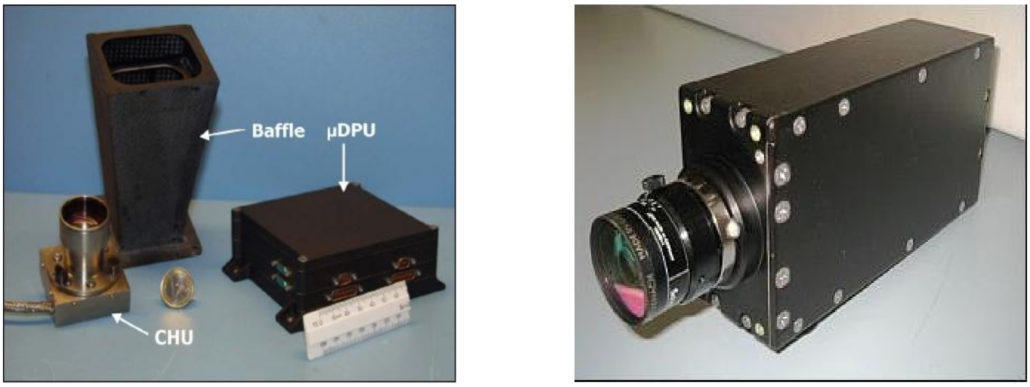
Scanning LIDAR systems provide 3D maps of the picked object by scanning their field of view with a laser and measuring the back-scattered light. In Figure 1.8 (a) the Scanning Lidar is illustrated, which was developed by GSOCs On-Orbit-Servicing and Autonomy group for research purposes [35]. The LIRIS LIDAR sensor depicted in Figure 1.8 (b) was developed by Jena-Optronik and tested during the ATV-5 mission as a new type of sensor for non-cooperative rendezvous and docking, see Section 1.4.1. Using another type of LIDAR sensors, viz. the Flash LIDAR, a full 3D image can be recorded by illuminating the scene with a single laser pulse and imaging the scene onto the focal plane array. The ability of the Flash LIDAR to perform an autonomous RvD application was demonstrated by using the DragonEye (see Figure 1.8(c)) in 2009 and 2011, and the Sensor Test for Orion Relative Navigation Risk Mitigation (STORRM)[36] in 2011 [37]. The advanced version of the LIDAR Camera System was demonstrated by the Neptec Design Group: Triangulation + LIDAR (TriDAR) automated RvD system, which was tested during three missions to track the ISS during docking/undocking and fly-around operations [38]. An image of the TRiDAR system carried on board Space Shuttle Discovery during STS-131 mission to the ISS is depicted in Figure 1.8(d).


(a) (b)

(c) (d)

Figure 1.8: (a): Scanning LIDAR. Image credit: see link in [39]. (b): LIRIS optical head tesdted during ATV-5 mission. Image credit: see link in [40]. (c): DragonEye 3D Flash LIDAR Space Camera. Image credit: see link in [41]. (d): The TriDAR Sensor Package during STS-128 mission. Image credit: see link in [42].

The measurement principle of passive sensors, namely camera systems, is based on the laws of imaging on the focal plane of a lens. The most recent flying PRISMA mission [25, 43] demonstrated the successful usage of the CCD-based high accuracy star tracker platform with Camera Head Units [44, 45], see Figure 1.9(a). In order to provide an accurate relative position and orientation w.r.t. the target, the visual system of PRISMA was equipped with two different cameras in order to cover different distance ranges: the far camera covering from 10.000 km to 100 m and the short camera covering from 500 m to a few centimeters. After showing successful performance during the PRISMA mission, the same star tracker has been used in the AVANTI experiment. A high-definition CMOS imager, shown on image 1.9 (b), have been used in the STORRM experiment as a docking camera.



(a) (b)

Figure 1.9: (a): Star tracker platform with Camera Head Unit used for the PRISMA mission. Image credit: see link in [46]. (b): Photo of the docking camera used in STORRM. Image credit: see link in [47].

Infrared-cameras have also been tested in space for the Proximity Operations. The principle of the Thermal Infrared Camera is to detect the infrared energy (heat) and convert it into electronic signal. Thereafter, a thermal image can be produced. The higher an object's temperature, the more infrared radiation can be received by the camera's optics. This type of camera can work even in complete darkness, since the ambient light has less influence on the image quality compared to visual cameras. The focusing lens of the infrared cameras can not be made from glass as in optical cameras, because glass blocks long-wave infrared light. Usually, they are made of Germanium or Sapphire crystals. Along with the LIDAR sensor, two Thermal Infrared Cameras in Figure 1.10 and one visible camera were also tested during the last ATV-5 mission. The TriDAR system also offers an optional thermal imager for extended range tracking and short range guidance. Nowdays, Neptec UK has designed and built a very small mass (less than 0.5 kg) Infrared Camera for multiple space RvD applications [48].

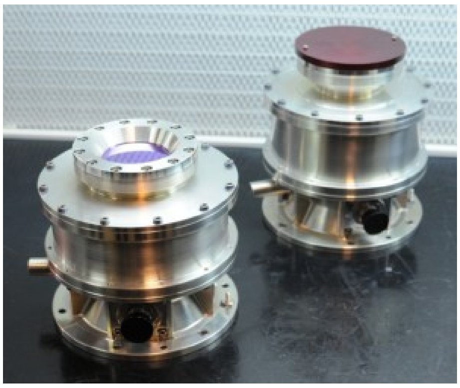


Figure 1.10: Sodern infrared sensors used in ATV-5 mission. Image credit: see link in [40].

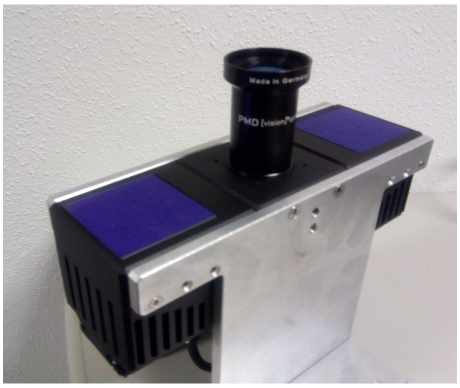


Figure 1.11: PMD Camera "CamCube 3.0".

About one decade ago a new type of ranging systems such as the Photonic Mixer Device (PMD) camera became available. The PMD technology is quite similar to a Flash LIDAR, where 3D imaging provides the distance to the target that can be measured in every pixel of a sensor chip by illuminating the scene with an array of LEDs. To the best of the author knowledge, this type of visual sensor has never been used in space applications so far. But investigations and tests of the PMD camera have already been started on the ground. In an earlier work of Schilling [49], the author performed investigations on the possibilities and benefits of PMD camera in RvD scenarios. In the work of Scharnagl [50] the algorithm of collision detection using PMD data (PMD CamCube 2.0) was presented and verified. More complex experiments were conducted with a tumbling target in order to prove the robustness of the algorithm. In the work of [51] the author has conducted some experiments using a data fusion from a PMD camera and a CCD camera during the close rendezvous scenarios. The PMD CamCube 3.0 camera which was used within the aforementioned experiment at DLR is illustrated in Figure 1.11. Tzschichholz tested different rendezvous approach trajectories under different environmental lightning and various disturbances. He identified that the impact of the sunlight is significant. Getting data from the PMD camera was possible only when the simulated strong sunlight was originating from the back side of the target and when it was located at the side of constellation.

From previous sections one can notice that a lot of work has already been done in close proximity operations to the non-cooperative target. Nevertheless, due to the crucial amount of space debris or uncontrolled spacecraft, there is still a research gap in the field of proximity operations with completely passive targets. In this subsection one has already seen some works, where the PMD ranging device was tested as a possible candidate for visual navigation in OOS missions. In this thesis, I continue to investigate use of a PMD sensor for the 6 degrees of freedom (3D position and 3D orientation) pose estimation of non-cooperative targets during the close rendezvous phase. My contribution in order to decrease the technological gap is a development of novel pose estimation techniques (pose initialization and pose tracking) with a unique PMD sensor for future OOS missions. In the Sections 1.4.3-1.4.5 the review of state-of-the art pose estimation techniques for different types of optical sensors are presented.

1.4.3 Review of 3D Techniques

To this date, many techniques have been developed for the purpose of object recognition and pose estimation using 3D data with the focus to find a robust and efficient manner to identify objects in a scene. In the field of robotic applications this problem is addressed as object recognition, which includes object identification and pose estimation [52, 53]. It should be noticed that I discuss model-based approaches, namely 3D a-priori knowledge about the model is avaliable. According to the extended survey for the 3D object recognition in cluttered scenes in the work of Guo et al. [54], the existing 3D object identification and pose estimation methods can be divided into 2 categories: global feature based methods and local feature based methods. Here, the term feature refers to some information, which can be extracted from the scene in order to solve the problem. Features can be points, edges or even objects. For the local and global methods the features are different. The idea of global methods is to not divide a model into parts or any geometrical objects, but conversely, use the complete 3D known model. A second class of methods, local methods, use so-called point descriptors. The scheme of this type of algorithms identifies point to point correspondence between the model and the scene. Point correspondence is built by comparing visual descriptors of the scene and of the model, which represent visual features of the content from both point clouds. Therefore, the descriptors must accurately and robustly distinguish between the real object in the scene and present noise.

Let us discuss some of the known algorithms for pose acquisition, applicable in robotics and also for some space applications for RvD. To initialize objects the so-called Generalized Hough Transform is suggested in [55], but its application is limited to primitive objects, as the recognition of 6 degrees of freedom (DOF) is computationally too expensive. Another automatic routine based on random sample consensus (RANSAC) to find basic shapes with point cloud input was designed by Schnabel et al. in [56]. In the works of [57, 58], the authors proposed the RANSAC techique for the pose initialisation of the non-cooperative target with LIDAR and stereo vision systems. The input data is a set of observed values, sampled and reconstructed in a model, which can be fitted to the observation. The inliers are generated from a random subset of the original data and tested against the entire set of data in order to determine the correlation between the inliers and the data set. The other approach which uses a Viewpoint Feature Histogram

(VFH) is presented in the paper of Rusu et al. [59] for mobile manipulation and grasping application with a prior segmentation of the scene. The VFH encodes the geometry and the view point for recognition and 6 DOF pose identification. In the work of Woods et al. [60] authors tested the VFH technique for the pose initialization of the non-cooperative target during proximity operations. Mian in his work [61] built three dimensional tensors from multiple unordered range images. The tensors are stored in a hash table and used like a lookup table during the matching phase in order to calculate 6 DOF. In [62], Johnson and Hebert introduce a recognition algorithm based on surfaces correspondence by matching points using the spin image representation. The main idea is to construct and store spin images from points on one surface (model) and thereafter do the same procedure for the spin image of the scene surface. The best point correspondence is established and grouped for the further surface matches in order to get a transformation matrix. In the work of Luis A. Alexandre [63], the author sums up different possible 3D feature descriptors, which exist and can be used nowadays in a Point Cloud Library (PCL) for 3D object recognition and pose estimation. In the work of Ruel et al. [64], the authors propose one alternative to the feature based methods by using a point based technique with a raw data from sensor. They apply a polygonal aspect hashing algorithm, where throughout limited set of poses, which are the most likely candidates, surface fit check between an input scan and known 3D model is performed in order to find best matching pose. The works of Drost et al. [65] and Papazov et al. [52] outline a 3D object recognition and pose estimation framework based on calculation of the features, such as point pairs, which are matched to a 3D known model using a voting scheme. According to the authors, the great advantage of this algorithm is its robustness in the presence of the occlusion and clutter. Some simple deterministic methods as Principal Component Analyses (PCA) and Singular Value Decomposition (SVD) have been used to find the orientation of the main axis of the target (other remaining components can be calculated by additional techniques) for the proximity operations [66, 34, 51, 67]. Both algorithms are pairwise registration based on the covariance matrix and the cross correlation matrix between two point sets. Along with the above referred algorithms, it is likewise quite usual to use different variants of the well known Iterative Closest Point (ICP) [68] algorithm for initial pose estimation [69], as well as for the pose refinement [70, 71], when the pose is already coarsely estimated. The key idea of the ICP algorithm is to iteratively minimize the average squared distance between the nearest points from two data sets (e.g. model and scene point clouds). The algorithm continues until the error converges or the maximum number of iterations is reached.

Once the initial pose is available, estimation of the pose in the next frame can be simplified by restricting the search to solutions, which do not deviate extremely from the previos ones. The model-based pose tracking of the target using the known 3D geometrical shape of the object and the 3D locations of the features in every frame leads to a least squares minimization problem in order to find the motion parameters [72]. The state-of-the art technique which is commonly adopted in 6 DOF 3D model-based tracking is the ICP algorithm [68]. This method has shown its effectiveness in the real-time matching of misaligned 3D shapes, as well as robustness in the real-world application [73, 74]. The ICP algorithm is robust for the coarse initial guess provided to the visual navigation system. The ICP algorithm with its different configurations has been applied for tracking purposes with the data sets from LIDAR and stereo camera systems in many works [60, 69, 74, 34, 67, 75]. In the work [34, 76], the Bounded Hough Transform (BHT) is introduced to track objects in a sequence of sparse range images. The key idea of the algorithm is to consider the coherency between frames, which allows to reduce the 6 DOF solution space in the vicinity of the previous pose with a follow estimation of the transformation matrix. Another approach for non-cooperative pose estimation with flash LIDAR based on edge detection is proposed in the work of Lim et al. [77]. The prominent geometric features, such as corners and edges, can be detected and the edge unit vectors for the detected edges are constructed. With two accurate edge vectors the pose is able to be calculated. One can obtain the most accurate result if two edge vectors are perpendicular. In the work of Opromolla et al. [78], the survey of the tracking techniques of completely model-free approaches are listed. Most of them include variants of Kalman filters, Bayesian filters and Particle filters for state estimation. Along with that there are also model-based tracking algorithms which utilize filter frameworks. For example, in the work of Krull et al. [79] an offline learning model-based approach with particle filter is presented. A 3D online pose tracker of rigid objects with a learning stage using depth images and known model is proposed in the work of [80]. The authors guarantee high robustness of the technique against occlusion, low memory consumption and low computational costs. Simultaneous localization and mapping (SLAM) [81] is another common technique to navigate with respect to so-called landmarks. Sonnenburg et al. [82] proposed SLAM algorithm for spacecraft rendezvous using stereo or monocular cameras.

1.4.4 Review of 2D Techniques

In the literature one can find a variety of the state-of-the-art model-based 2D pose estimation tracking techniques based on point features or contours and edges [83, 84]. One of the first model-based 3D tracker for the real-time application was RAPiD [85]. It uses some model features, and namely, the points located on high contrast edges. The control points located on the weak, error prone edges are ignored, since they can lead to inaccuraces during the tracking. The bad/weak edges can occur due to bad illumination conditions. This is a common problem present in the edges/contour methods. Once the initial pose exists, the control model points match the image points with subsequent linearisation in order to estimate 6 DOF. In a conclusion Harris et al. resumed that the algorithm proved its robustness and fast execution, when the enhancement listed in the paper [85] concerning the control points choice are taken into account. The other edge based approach is proposed in [86]. The visible edge features of the model are identified online and the correspondences are found in the image frame. A Lie group formalism is used in order to simplify the tracking problem. Further, the optimization problem solved by means of iterative reweighted least squares. Petit et al. [87] proposed 3D model based tracking for the space debris removal with edge matching process, where the image edges are found with 1D search along the edge normals. The pose is computed by minimizing the distance between 3D projected edges of the model and detected edges in the image. A set of 2D lines of the satellite has been detected in works of Cropp et al. [88] and D'Amico et al. [89] as a special image feature of the target for the model-based pose estimation.

Optical flow methods based on the assumption that the pixel intensities of the object do not change within consecutive frames and the neighbouring pixels have a similar motion. The Lucas-Kanade [90] method is a widely used algorithm for optical flow estimation. This local method can solve the optical flow equation by a least squares criterion. The regularized optical-flow method is applied in [91] to 3D tracking during the fast motion. Combination of the optical flow and edge detection is used for tracking in the work [92] in order to increase stability and accuracy of the tracker.

Template-based pose tracking is another branch of the diversity of tracking techniques, which corresponds to the global methods. In this contest, the term template refers to an image. In a template-based category the whole template pattern of the object is compared to an observed image and the best match between both of them helps to find the pose. In the work of Hager et al. [93] the tracking is based on the minimization of the sum-of-squared differences (SSD) between two regions. The Fast Normalized Cross Correlation algorithm is presented in the work [94], where the matching occurs by intensity values in the initial target bounding box as a template [83]. The sampled uniform windows around the previous target position are compared with the target template using the cross correlation. The candidate window with the highest score is admitted to be the right new target location. The Mean Shift Tracking [95, 83] uses RGB-color histograms in order to find the target position in each new frame. The comparison between the target model, which is presented as a color distribution, and the target candidate is expressed by a Bhattacharyya metric [96]. The best location of the target in a new frame can be found using mean shift by finding the peak of a confidence map near the target's old position.

Another approach is a probabilistic approach with Bayesian filters for estimating the density of successive states in the space of possible camera poses [84]. The estimation of the state vector occurs by using a dynamic model and given observations or measurements. Kalman fitering [97, 98] is a tool for recursively estimating the state of the process during the 3D tracking. Particle filters with their advantages are presented in the work [99] for relative rolling estimation with a sequence of infrared images. Also Augenstein et al. [100] estimate recursively the 6 DOF and 3-D shape of the target with a Rao-Blackwellized particle filter [101] for the autonomous docking with an unknown tumbling target.

The combination of different approaches is useful and allows to take advantages of each method for the frame-to-frame target tracking. In the work of Marimon et al. [102] the authors use a hybrid tracking system, which combines measurement from a marker-based cue and a feature point-based one with a particle filter. The proposed filter uses different cues within a single motion model. In the work of Panin et al. [103] contour based tracking is proposed, as an integration of local color statistic and intensity of edges. Yol et al. [104] present two approaches for the future space debris removal missions, which combine points of interest and edge features, as well as color-based features. Convolutional Neural Networks (CNN) algorithms are modern methods for the 6D pose estimation of uncooperatibe targets are presented in works of [105, 106]. Usually, the method consists of off-line learning phase and on-line prediction phase. The synthetic space imagery data sets are a prerequisite for the CNN algorithms.

1.4.5 Review of Hybrid Navigation Techniques

The hybrid pose estimation techniques refer not only to the combination of different methods with one visual sensor. Multiple visual sensors and applicable algorithms with the following combination of the measured information for the pose tracking are referred to hybrid navigation. In the work of Padial et al. [107] the authors present the 3D tumbling target reconstruction and relative pose estimation for the RvD via fusion of monocular vision and sparse-pattern range data. They implemented a modified Rao-Blackwellised Particle Filter and demonstrated the test performances. Feature-level data fusion for autonomous localization is presented in the work [108], where the depth information from the infrared sensor is fused with the image from the camera through the edge information. Cho et al. [109] developed a multi-sensor fusion system for moving object detection and tracking. An Extended Kalman Filter (EKF) treats observations from radars, LIDARs and monocular cameras to consistently track objects. Alatise et al. [110] and colleages from DLR [111] propose to use Extended Kalman filter and fuse the data from an Inertial Measurement Unit (IMU) and a camera in order to estimate position and orientation of a mobile robot. The object tracking with two sensors like Stereo RGB-D and LIDAR is given in the work of [112]. Both sensors independently measure the pose and thereafter fuse it with Bayesian filtering. By doing that, the frame-to-frame track losses were prevented by redundant stereo measurements. In the PhD thesis of Ventura [74], three combinations of data fusion with EKF for RvD with an uncooperative target have been tested. The first combination consists of GPS, IMU and Kinect v2 ToF sensor, second one fuses the data from GPS, IMU and stereo vision, and the third one uses data from GPS, IMU and monocular camera. As one can notice, there are different fusion architectures depending on the application's aim, types of sensors and requirements for the fused system. For example, the raw data fusion (fusion of the images from sensors) or fusion of the results of the processed data.

1.5 Contribution of the Thesis

The aim of this thesis can be formulated with the following concept: the developed robotic systems for OOS and ADR must be able to perform an autonomous rendezvous to non-cooperative targets with known geometrical parameters. Non-cooperative targets are lacking rendezvous aids such as retro-reflectors, transponders or other special markers. Moreover, there is no way to exchange telemetry between the service satellite and the target. It means that the GNC system of the chaser should ensure robust real-time motion estimation of the space object without any confidential information about it. For this purpose, the main contribution of this thesis is the development, validation and verification of the novel suitable pose estimation techniques using computer vision technologies with the proposed Photonic Mixer Device visual sensor. In Section 2.3 of this thesis, more detailed information about characteristics, features and possibilities of the sensors with PMD technologies is provided.

Let us describe more specific the problem and outline the objectives of the thesis. In Figure 1.12 the sketch of the geometry of the chaser with a fixed camera and the target for the proximity operations represents graphically a problem. The goal of the pose estimation technique is to determine the relative position vector t^B and relative orientation, i.e. attitude, of the target principal frame B (body frame) with respect to the PMD camera coordinate frame C using the measurements provided from the PMD sensor and the knowledge about the model of the target. Notably in Figure 1.12, the attitude of the target spacecraft principal frame B with respect to the camera frame C is represented by the rotation matrix R_{BC}.

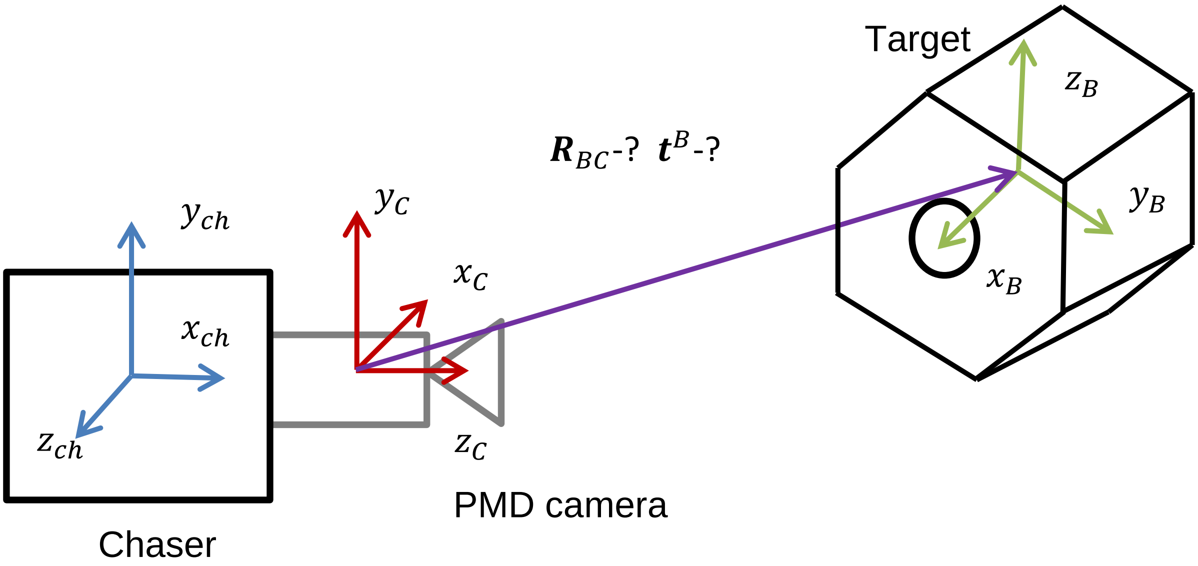


Figure 1.12: Geometry of the target and the chaser with attached PMD sensor.

The PMD navigation sensor of the chaser spacecraft, which is the unique visual sensor used in this thesis, acquires images of the target vehicle during the maneuver with subsequent onboard processing. The tracking of the target is based on the frame-to-frame pose estimation procedure depicted in Figure 1.13.

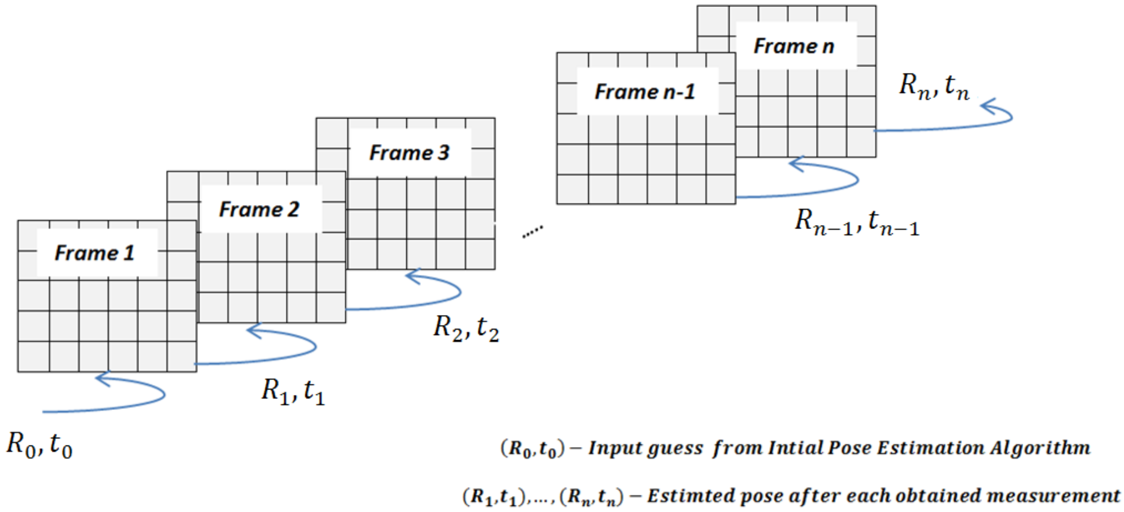


Figure 1.13: Frame-to-frame tracking of the space target in a sequence of PMD images.

The standard approach for uncooperative pose estimation is to detect the known features of the target in the sensor measurements and fit the known 3D model on them. The key challenge here is to retrieve the spacecraft's patterns under the varying illumination conditions in space since high reflectivity of some target spacecraft surfaces may cause corrupted measurements. Besides that, the depth information of the scene can be partially lost. Furthermore, the features of the target may vary from frame to frame due to the uncontrolled rotational motion of the target. Systematic and non-systematic errors of the PMD sensor measurements create the challenge for the choice of pose estimation techniques. Regarding all these difficulties, this thesis outlines the following main objectives.

1. Firstly, this manuscript presents a development of a stable approach for a navigation system, which will allow the chaser to acquire an initial pose of the non-cooperative space object knowing only the geometrical model of target and the measurement from the PMD sensor. The proposed pose initialization pipeline uses the combination of the state-of-the-art 3D pose estimation algorithms, such as Point Pair Feature algorithm and ICP method. The follow 2D refinement pose estimation technique with an amplitude image is applied after careful study of PMD sensor and its pros and cos.

2. Secondly, thesis provides a Hybrid Navigation Technique (HNT), where two sources of PMD data (depth and amplitude) are used for a robust and stable tracking with real-time state estimation during the approach to non-cooperative target. With the proposed HNT the tracker should not fail during the rendezvous phase, even if the depth data is (partially)lost. The proposed technique does not include different sensors as it was mentioned in some works in Section 1.4.5. The concept does have a software redundancy but without a hardware redundancy. The ICP algorithm with a reverse calibaration technique for the neighbor search is proposed for pose estimation with a depth image. Thanks to the organized point clouds of PMD sensor, the computational complexity of neighbor search can be decreased from 3D to

2D space. The image processing with detection of straight lines using Hough Lines Transform and following 3D-2D pose estimation using Gauss-Newton minimization problem is applied for the small resolution amplitude images.

3. Thirdly, the author presents experimental test campaigns for validation and evaluation of the developed methods including real images from the PMD sensor and a high-accurate hardware-in-the-loop proximity simulator. The tests scenarios are conducted with an extra powerful spotlight and in total darkness. The PMD sensor is able to show its functionality in a simulated umbra phase using only its own illumination unit.

The mentioned objectives can not be accomplished without additional steps which play an inevitable role for the accurate assessment of the visual navigation system:

- PMD sensor calibration, including intrinsic and absolute extrinsic parameters in order to accurately extract features from the images
- hand-eye-calibration of the PMD sensor, which delivers the pose of the camera w.r.t. the chaser; it is necessary for the precise verification of the state vector and for the controlled relative approach in a closed loop

In section 2.4 we showed how the PMD sensor for the first time was calibrated using DLR CalDe and DLR CalLab calibration toolbox.

Chapter 2

Hardware Facilities: Rendezvous Proximity Simulator and Visual Sensors

Multiple tests of the developed software and hardware, especially for the costly space missions, are the inevitable pre-step procedures before a real mission can take place. The common element of OOS and space debris removal missions is the proximity operation to the target object. For On-Orbit Servicing operations different configurations of the rendezvous and docking scenarios can be considered dependent on the size and mass of the target, its position in orbit and also from the intentional OOS activities. If there is a special testbed for such applications, one is able to simulate the aforementioned configurations. The European Proximity Operations Simulator (EPOS) simulator located at DLR is presented in this chapter as an available ground based set up for testing the feasibility and safety of the future OOS missions. Moreover, there is also description of the mockup, the PMD sensor and calibration procedure.

2.1 European Proximity Operations Simulator

During the last two decades, DLR has been involved in the process of simulating RvD scenarios. The first European Proximity Operations Simulator (EPOS) had been developed from 1985 to 1998 by DLR and ESA for the simulation of rendezvous maneuvers in the laboratory. In the period from 2008 to 2009 the renewed EPOS 2.0 was developed and established at DLR entirely replacing the previous test bed. EPOS 2.0 is a hardware-in-the loop RvD ground based simulator, which consists of two 6 DOF industrial robots manufactured by KUKA [113].Using EPOS, one has the capability to simulate a 6 DOF relative dynamic motion of two spacecraft in a close range starting from 25 to 0 m. In Figure 2.1 one possible setup for RvD test scenario on the ground using EPOS is illustrated.

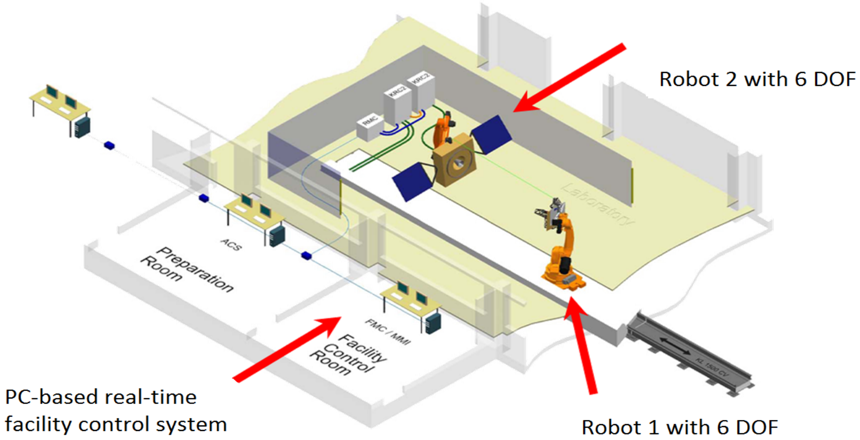


Figure 2.1: Illustration of one possible setup for the RvD simulation in the EPOS laboratory.

The KUKA KR100HA robot (Robot 1 in Figure 2.1) is mounted on a rail system and can move along it simulating the 6 DOF motion of the chaser. The KUKA KR240-2 robot (Robot 2 in Figure 2.1), is fixed at the end of the rail system. Each robot is equipped with a breadboard attached to the flange, see Figure 2.2, which can be used as a mounting board for the mockups and RvD visual sensors [113, 114, 115]. The simulation is controlled and monitored by a set up of computers. To create an almost realistic space environment in the EPOS laboratory, a space-representative lightning using a sun simulator and black covering material for the robot and walls for the dark background are provided.

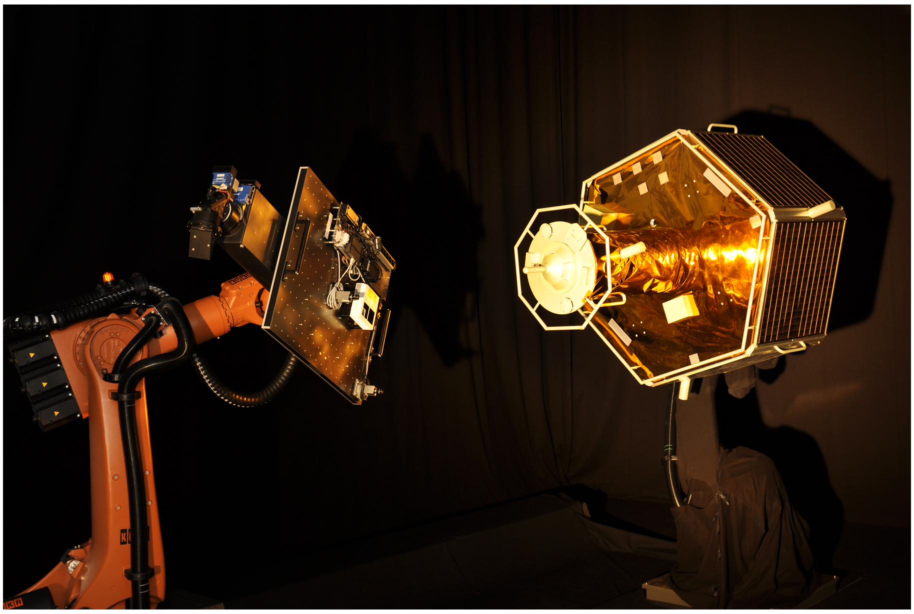


Figure 2.2: The robot on the left side of the image carries a PMD sensor and the robot on the right side of the image is equipped with the target mockup.

2.2 Mockup

For the test scenarios handled in this thesis the client satellite mockup mounted on the robot with its coordinate system is presented in Figure 2.3.

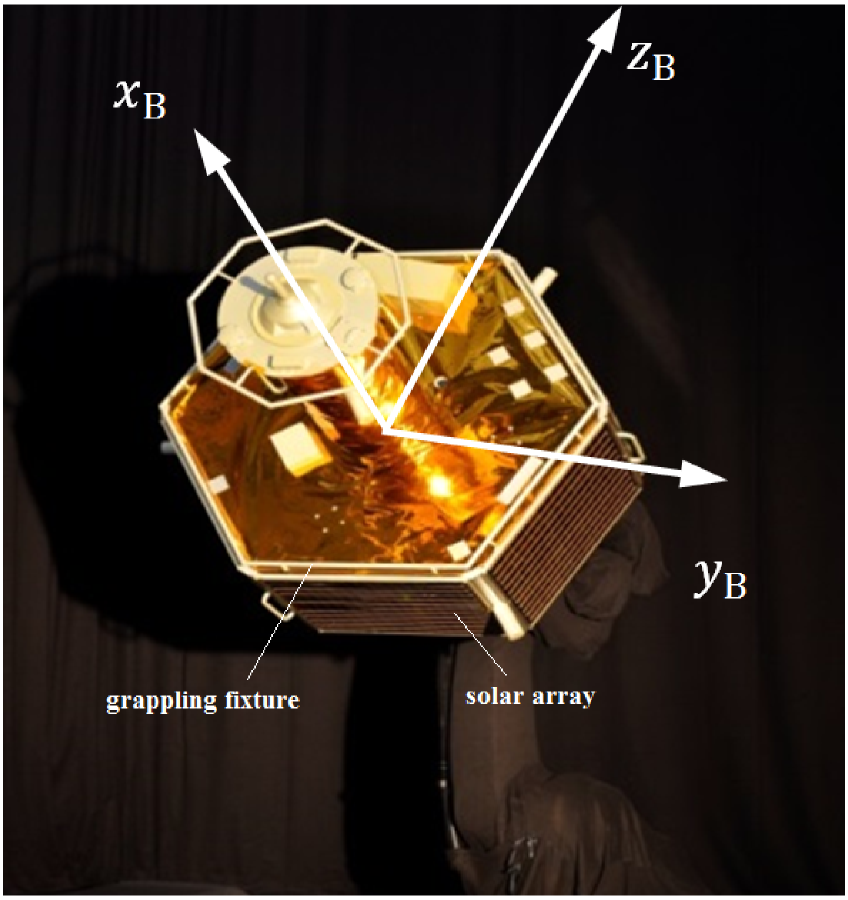


Figure 2.3: The target mockup mounted onto DLR's EPOS laboratory and the target body frame B.

The mockup's surface consists of materials, which have optical properties close to real satellites. This fact is demanded as a high apriori condition by simulating the OOS scenarios in the laboratory with the real hardware visual sensors. Regarding the applicability of the proposed PMD sensor in space missions for the high reflective and specular surfaces of the spacecraft, it is of great importance to test the sensor on the ground within a laboratory and with the complex realistic mockup. Figure 2.4 illustrates the corresponding detailed 3D model of the mockup. The full 3D model contains 70002 vertices.

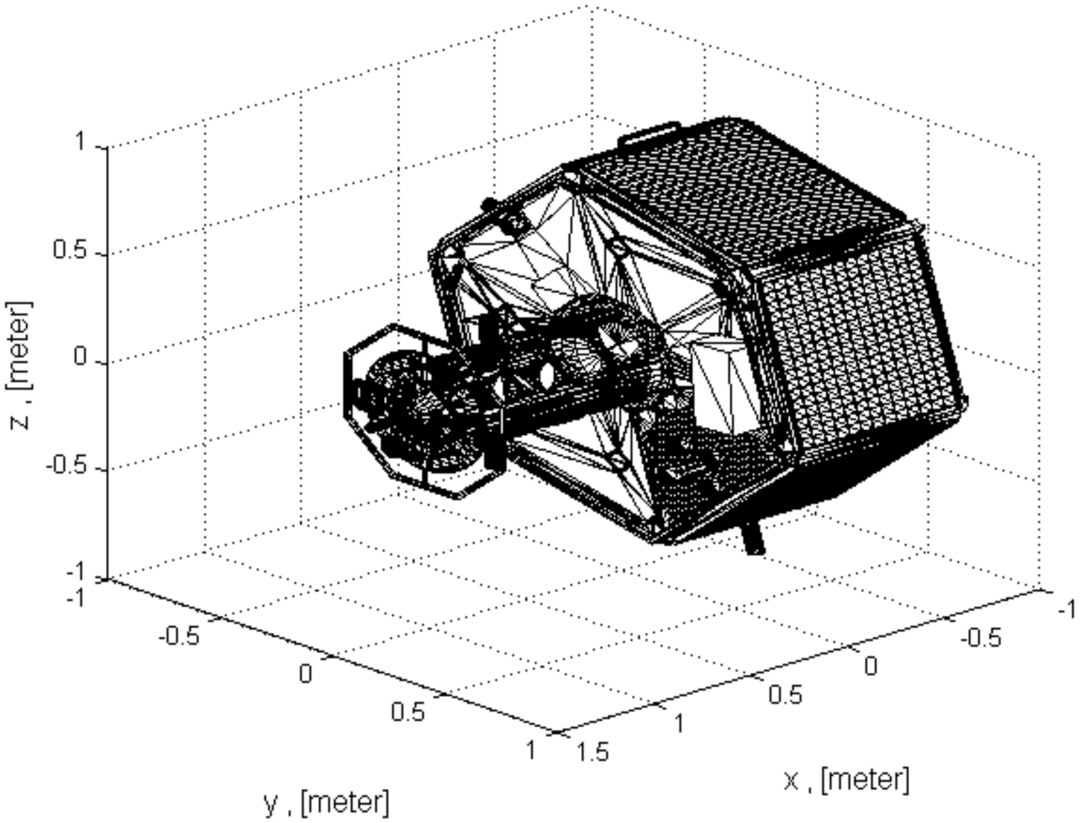
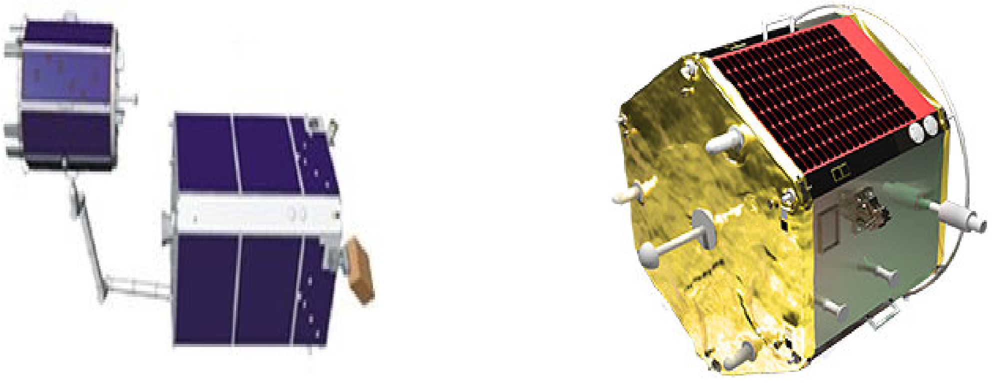


Figure 2.4: The original 3D model of the target mockup.

The mockup's configuration reminds a client spacecraft, which was intended for the Deutsche Orbitale Servicing Mission (DEOS) [116, 117]. Figure 2.5(a) illustrates the configuration of the client and servicer satellite intended for the DEOS mission, whereas Figure 2.5 (b) shows only the client vehicle.



(a) (b)

Figure 2.5: (a): Configuration of the client and servicer satellites. (b): Client vehicle for the DEOS mission. Image credit for both images: see link in [118].

There are many construction features of the mockup, which are very similar to that of the client of DEOS. For example, solar arrays and a grappling fixture in form of hexagon in the front part of the mockup, in case when the sevicer vehicle aims to capture the client. These mentioned interface parts of the mockup are denoted with the text in Figure 2.3. The given mockup has also a "nose" part, which is intended to the docking, if this one is planned. The fixed mockup on the robotic arm in the EPOS laboratory can spin and tumble according to the defined scenario for the non-cooperative target.

2.3 DLR-Argos3D Camera

In Chapter 1, visual sensors for rendezvous have been introduced. As an alternative to the cameras and LIDAR systems, I propose to investigate the PMD camera as a ranging system in order to use it for relative pose localization of the target during the rendezvouse phase. Since the appearance up to today the PMD technology has been improved in terms of the resolution of the PMD chip, measurements accuracy and robustness with respect to the working conditions. In the work of Piatti et al. [119], the authors bring a detailed table with different types of the PMD sensors. One can observe an enhancement of the resolution, where the first PMD sensors (PhotonIcs PMD 1kS) had a resolution of 64 × 16 pixels [120, 119] and today one can get it with a resolution of 352 × 288 [121].

In this thesis, a DLR-Argos3D - P320 camera, see Figure 2.6, is used, which was specially configured and provided as a prototype by the Bluetechnix company [16].

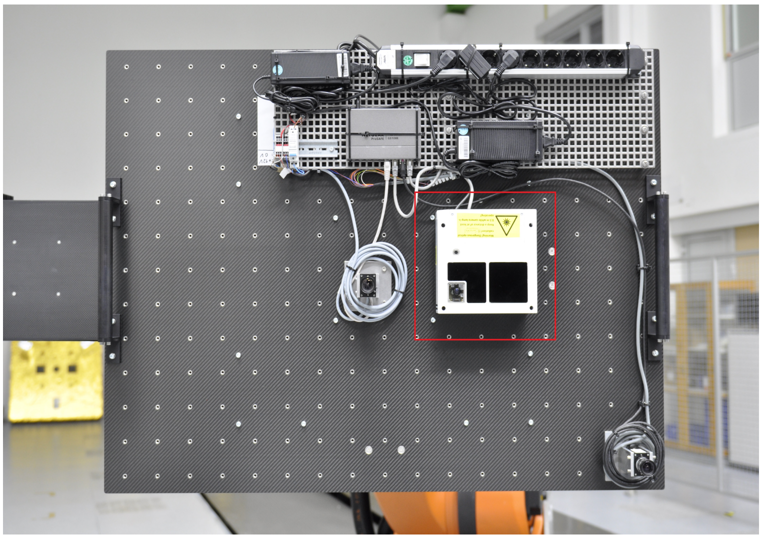


Figure 2.6: DLR-Argos3D - P320 camera fixed on the breadboard in the EPOS laboratory.

The camera is highlighted on the figure with a red frame. The device contains a 3D PMD sensor. For the purpose of this thesis only PMD sensor was employed, its technical characteristics are summarized in a Table 2.1.

The DLR-Argos3D - P320 camera includes also 2 Light Illumination Modules (LIM), where each module includes 6 high-power IR LEDs with a wavelength λ being in the range of $\lambda \approx$ 865nm - 870nm (peak emission). The emitted signal of the IR LEDs is an amplitude modulated (AM) signal, where the frequency of the signal can be set by the user, see in Table 2.1. The output of the PMD is a de-facto 2D image with a depth

Table 2.1: Technical data of the PMD sensor inside the DLR-Argos 3D-P320 camera.

Field of View	28.91 x 23.45 deg
Resolution of the chip	352 x 287 pixels
Integration time	24 ms
Frames per second	45
Modulated frequencies	5.004 MHz, 7.5 MHz, 10.007 MHz, 15 MHz, 20.013MHz, 25.016 MHz, 30MHz
Mass/Power Consumption	2 kg/ < 25.5W

measurements for every pixel. Beside the distance information, usually, a PMD sensor provides additional amplitude values by the same pixel array at the same time. The amplitude image corresponds to the amount of the returned active light and represents the quality of the measurements [122]. The higher the amplitude value of a pixel, the more reliable is the distance value. Since I get the amplitude image of the scene, it can be handled as a gray-scaled image described by the pinhole camera model [123]. Monocular vision is a special case of the PMD sensor and not the primary goal.

2.3.1 PMD Camera Working Principle

The basic time-of-flight principle is to determine the distance from the absolute time delay between the transmitted wave fronts from the sender illumination unit and the wave fronts reflected by the objects surface[15, 124]. Alternatively, the distance can be computed using the principle based on a phase shift of a periodically modulated signal. The phase shift φ is measured between the emitted modulated signal from 12 IR-flash LEDs integrated in the camera and the reflected signal from the observed surface.

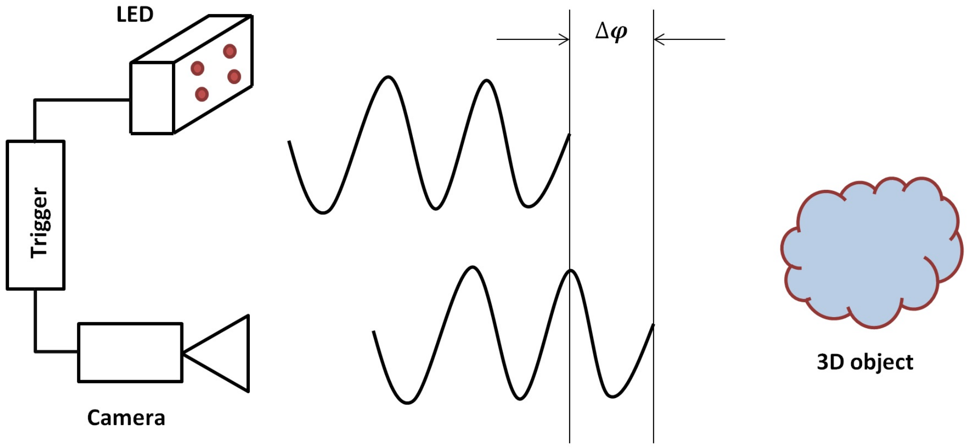


Figure 2.7: PMD camera working principle.

To calculate the phase shift φ , the receiver needs to know the emitted reference signal as a reference [125]. For this purpose, the illumination modules and the receiver are connected to the same signal generator. Usually four samples are acquired with different phase offsets and the phase shift is calculated with the four phase algorithm [126]. Each pixel of the sensor chip has several transistors and capacitors in order to collect electrons.

The amount of electrons is measured at multiple time locations. Assuming that the AM emitted IR light is presented in the form of a sinusoid, the amount of collected electrons $Q_1 \dots Q_4$ is measured usually in phases $\psi_0 = 0°$, $\psi_1 = 90°$, $\psi_2 = 180°$, $\psi_3 = 270°$. Please, refer to the detailed description of PMD operational principle to the follow works [126, 125, 124, 51].

During the fixed integration time the electrons are charged through multiple cycles of the IR signal in order to collect $Q_1 \dots Q_4$. Once $Q_1 \dots Q_4$ are measured, the phase angle between illumination and reflection can be presented as:

$$\varphi = \arctan\left(\frac{Q_3 - Q_4}{Q_1 - Q_2}\right). \tag{2.1}$$

The distance can be calculated as follows:

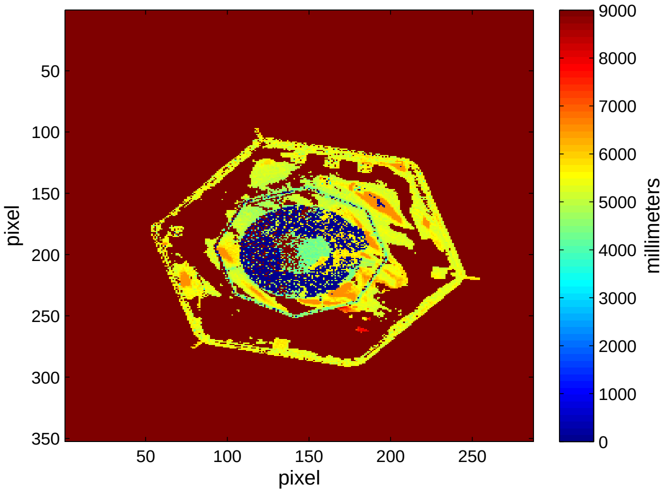
$$d = \frac{ct}{2} = \frac{c\varphi}{4\pi f_{mod}}, \tag{2.2}$$

where c is the speed of light $c = 3 \times 10^8$ m/s, φ is the measured phase shift and f_{mod} is the modulation frequency of the emitted signal. The signal strength, namely the amplitude A of the correlated signal can be obtained if the duration of the signal is determined [51], described as follow:

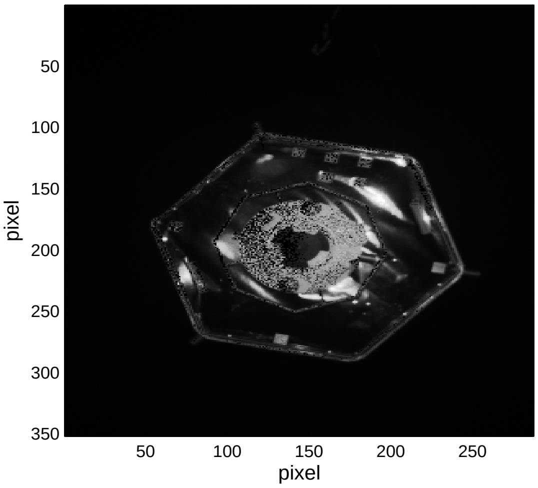
$$A = \frac{1}{2}\sqrt{(Q_1 - Q_2)^2 + (Q_3 - Q_4)^2}. \tag{2.3}$$

2.3.2 Distance and Amplitude Data

The DLR-Argos3D - P320 sensor has up to four channels available in order to transmit the sensor data via its interface. In this work I am interested in the distance and amplitude data, which can be retrieved for each frame from the sensor. Distance data (in millimeters) and the amplitude data from the PMD sensor are sent as 16-bit unsigned (Uint16) values. The amplitude array can be used to generate a gray-scale image of the scene. Example of depth and amplitude gray-scaled images from the DLR-Argos3D - P320 sensor taken in the EPOS laboratory are presented in Figure 2.8.



(a)



(b)

Figure 2.8: (a): Example of a depth image. (b): Example of an amplitude image.

Measurements from PMD cameras (also from other Time-of-Flight (ToF) cameras) suffer from different systematic and non-systematic errors. In the work of Fürsattel et al. [127] the authors listed six dominant error sources dependent on the camera model: temperature related errors, temporal variations, integration time, internal scattering, amplitude related errors, wiggling. In the work of Tzschichholz [51], the author also addressed some other PMD errors, such as multipath interference, jump edge effect, motion artifacts. At the beginning of the development of PMD technologies, improvements in the hardware and software parts have been done including an integration of correction methods for error compensation. Therefore, in this work I do not focus on the PMD error correction, rather than use the software tools of the DLR-Argos3D - P320 camera developed by the Bluetechnix company.

Confidence of Depth Measurements

Every pixel of the PMD sensor provides co-registered range and amplitude data in every frame. The amplitude image corresponds to the amount of the returned active light. The objects with highly reflecting surfaces create large amplitude values, scattering or less reflective surfaces cause low amplitudes. Therefore, the materials and colors play a crucial role in the calculation of an accurate depth measurement. Along with the material properties of the target, the distance measurements are noisy due to the other external and internal errors mentioned above. In the works of Frank et al. [128, 129], Mufti et al. [130] and Dal Mutto et al. [131] it was found out that the standard deviation of the range data is mutual to the amplitude of the signal. The distribution of the pixel noise in the ToF cameras can be approximated by a Gaussian distribution with standard deviation:

$$\sigma_d = \frac{c}{4\pi f_{mod}} \frac{\sqrt{I/2}}{A} \tag{2.4}$$

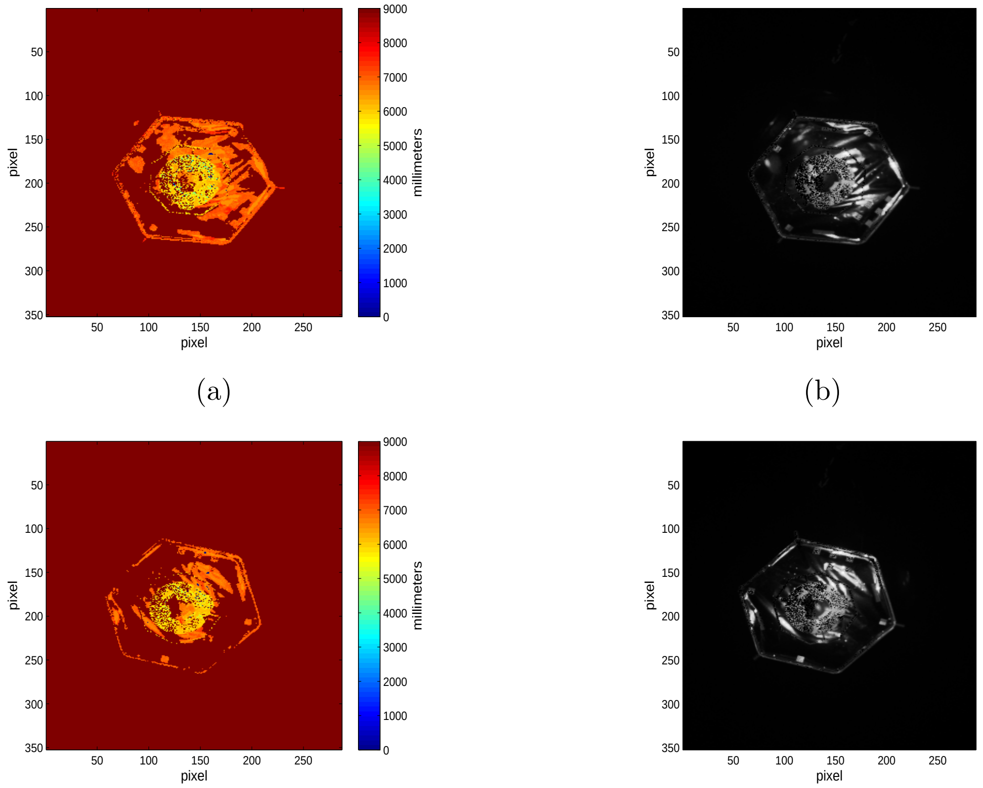
where f_{mod} is the modulated frequency of the signal emitted from the ToF's LEDs; A is the amplitude value measured in a pixel; I is the intensity of the signal in the same pixel; $c = 3 \times 10^8$ is a speed of light. From Equation (2.4) one can observe the following: when the amplitude of the signal increases, the given standard deviation decreases and in turn improves the precision of the measured range data. According to [132], the intensity I depends on two factors: the amplitude A of the received signal and the background illumination. If the intensity goes up, the depth precision drops. As it was already noted before, the increase of the amplitude value leads in total to a better depth precision. Therefore, the important factor in order to improve the accuracy of the measured depth value is to decrease the non-modulated background illumination. The background light is an ambient light, which occurres in space in form of sun light. Consequently, the amplitude of the received signal is the optimal element for the assessment of the distance measurement. Concerning the modulation frequency of the emitted signal - the bigger it can be set, the higher the precision of the measured depth value can be obtained, because the measurement range of the camera system drops. In practice, it is difficult to produce modulated light with high power LEDs and high frequencies [126]. Therefore, the available range of the modulated frequencies for PMD sensors varies from one camera to another.

Amplitude Threshold

By default the DLR-Argos3D - P320 sensor performs amplitude based filtering for every pixel to invalidate unreliable measurements. According to the datasheet of the DLR-Argos3D - P320 sensor, the camera software transfers the confidence array of the pixels in Uint8 format in parallel to the depth and amplitude measurements. The confidence value in this array is a measure of the confidence of the corresponding depth value. A value of 255 (maximum value) means 100% confidence and a value 0 (minimum value) means a confidence of 0%.

On-board check for the invalid distance pixels is regulated by setting two following registers: *ConfidenceThresLow* and *ConfidenceThresHigh*. If the measured amplitude value in a pixel is lower then the threshold *ConfidenceThresLow*, the measured distance in this pixel gets the maximum distance value. This pixel is marked as an underexposed unstable pixel. In case of the overexposed pixel, the measured amplitude value exceeds the threshold *ConfidenceThresHigh* and the distance in this pixel gets the minimum distance value. It means that the relation I showed in a previous subsection "the higher amplitude value, the higher quality of the depth measurement" works only up to a certain maximum confident amplitude value. The overexposed pixels are not able to provide distance measurements. Both of these thresholds indicate that every pixel should have a certain amount of received light for a valid depth measurement.

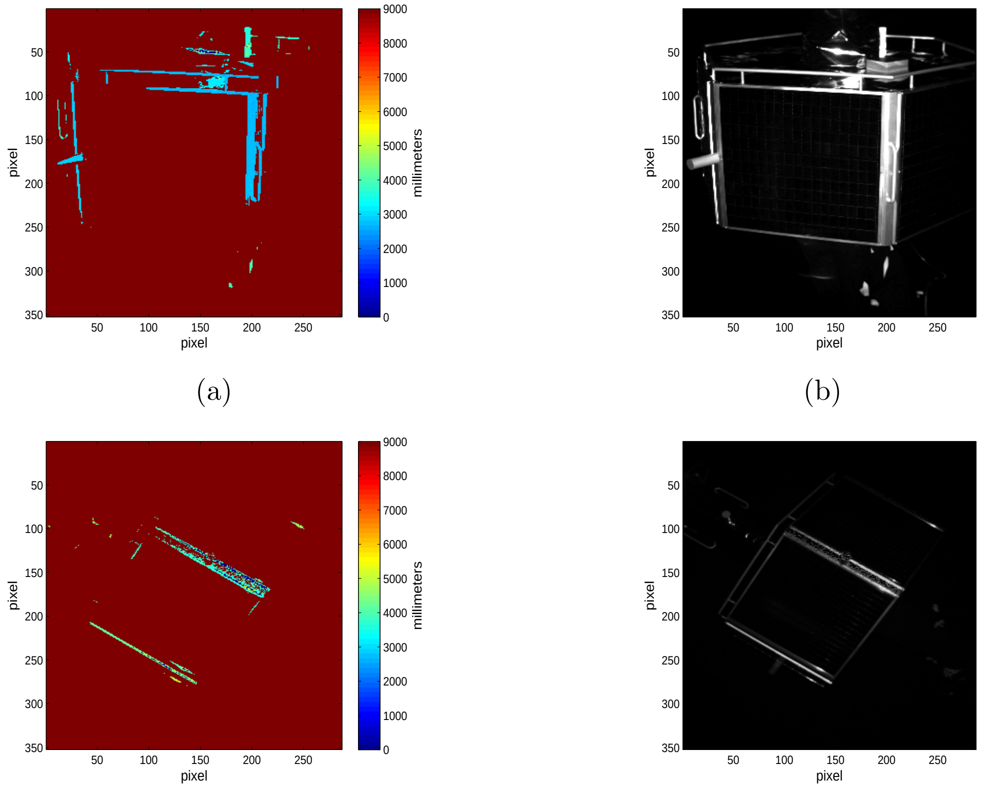
Next, I provide some image examples, where one can observe the physical influence of the amplitude threshold on the depth measurements. Figure 2.9 reflects depth and amplitude images, which have been taken within the same illumination condition in the EPOS laboratory and with the same relative distance between the camera and mockup. The images show the front view of the mockup, where the hexagon, octagon contours and central part of the nose are made from solid non-light absorbing materials (see Figure 2.3).



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

Figure 2.9: (a): Depth image with the *ConfidenceThresLow*=500. (b): Corresponding amplitude image with *ConfidenceThresLow*=500. (c): Depth image with *ConfidenceThresLow*=1000. (d): Corresponding amplitude image with *ConfidenceThresLow*=1000.

Images (a) and (b) on Figure 2.9 were taken when the amplitude threshold *ConfidenceThresLow*=500. Images (c) and (d) on Figure 2.9 were taken when the amplitude threshold *ConfidenceThresLow*=1000. One can notice the evident difference in the depth images (a) and (c), namely the entirety of the hexagon contours. When the threshold is set to a lower value, more valid pixels appear in the raster depth image, see Figure 2.9 (a). The image on Figure 2.9 (c) has thin and broken lines of the contour, meaning an absence of the valid pixels of the measured depth. But one should be very careful with a choice of the ConfidenceThresLow. Because by decreasing this threshold, more depth measurements appears but it doesn't mean that the measurements have a high accuracy and can be considered as confident. The amplitude images for the both cases almost similarly depict the scene object. Figure 2.10 depicts amplitude (*b* and *d*) and distance (*a* and *c*) images of the same mockup, but taken from the side with a certain angle. The images taken with different relative distances, however, with the unique *ConfidenceThresLow*=1000. Images (a,b) in Figure 2.10 correspond to an approximate relative distance of 2.7 meters and images (c,d) to a distance of 4.5 meters between mockup and PMD sensor. It should be noted that the side surfaces of the mockup simulate solar panels (see Section 2.2), therefore the its material absorbs incoming light in order to convert it to electricity. Some of the coating materials intended for real solar panels still reflect some light, but it is not enough for the accurate calculation of the distance per pixel. This effect is evidently demonstrated with the depth images presented below.



(a)

(b)

(c)

(d)

Figure 2.10: (a)-(b): Depth and amplitude images of the mockup's side with *ConfidenceThresLow*=1000 and relative distance 2.7 meters. (c)-(d): Depth and amplitude images of the mockup's side with *ConfidenceThresLow*=1000 and relative distance 4.5 meters.

From the distance images in Figure 2.10 (a) and (c) one can notice that the distance could be calculated in some edgings spots around the solar arrays, because of the different contour material. The amplitude image in Figure 2.10 (b) reflects the good gray-scaled image of the side part. Almost the same situation happened when the distance to the target is increased. The object on the depth image in Figure 2.10 (c) is hardly to detect, whereas the amplitude image gives a visible silhouette of the target. This last case with the side view is really difficult situation, where only the amplitude image can be used for the pose estimation. I am not considering it in this thesis.

2.3.3 PMD Camera vs. Monocular, Stereo Cameras and LIDAR Systems

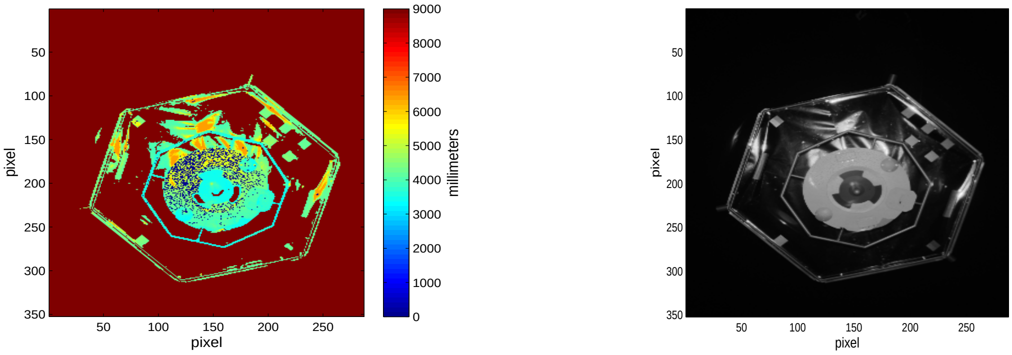
Conducting the research in the field of a practical use of the PMD camera for the future OOS activities, I would like to provide some comparison of the PMD camera characteristics vs. monocular and stereo cameras, as well as PMD cameras vs. LIDAR systems.

For the first sight, comparing the PMD camera with a monocular one, one gets the evident difference - the automatic depth measurement for every pixel of the sensor chip [15]. The range information with the stereo camera has to be found via a triangulation principle. This process demands additional processing time as well as computational power in order to estimate the depth accurately. The correspondence problem appears for the stereo camera during the search for the appropriate pixels from two images in order to get the range value.

The strength of the 2D camera is color data, whereas the PMD camera provides only gray scale output. A change in the gray level values of the both cameras is not necessarily caused by the movement of the object. If illumination conditions vary noticeably, this causes gray level changes. Therefore, object motion detection cannot be described by the gray level values changes. This problem is eliminated in the PMD sensor, because the PMD sensors use a different technique to calculate the range value [133].

Having conducted some research, it was revealed that the resolution of the sensor chip of the PMD sensor has reached up to 352 x 288 pixels, whereas monocular cameras have a resolution up to hundred megapixels. For example, the camera prototype PMD CamCube 3.0 has a resolution 200x200 pixels and the engineering sample DLR-Argos3D, which I am working with, has a resolution 352 x 287 pixels. For the close range approach I would not see this fact as a drawback, because increasing the resolution of the sensor chip, leads to an increase of the computational time in processing of one image from the PMD camera, which is in general undesirable. At the same time, the bigger resolution of the sensor chip, the better the accuracy of the translational components of the position. This is, because a small motion for example to the left/right, which results in a sub-pixel changes, can not be recognized.

The disadvantage of both stereo and PMD cameras for the object acquisition is the sensitivity to specular reflections and lightning conditions. For the stereo camera, different light scatters can cause additional shadows on the some parts of the target from itself or from the chaser. Moreover, with a stereo camera, a sufficient texture is necessary for the accurate stereo vision. Some inaccuracies of the output data from the PMD camera are very noticeable in presence of a strong external IR illumination (e.g. direct sun light), as well as highly reflective objects (specular surfaces) [131, 51]. This phenomenon is inherited by the PMD cameras because of its sensor chip structure. Operating in the full umbra phase can be impossible without any additional illumination for the stereo cameras, but with the PMD sensor it seems doable. In Figure 2.11, I plot the depth and amplitude images from PMD camera, which have been taken in complete darkness. These images look like the images taken with a sun simulator (e.g. Figure 2.8) and contain sufficient details about the object.



(a) (b)

Figure 2.11: (a): Depth image from PMD sensor taken in complete darkness. (b): Amplitude image from PMD sensor taken in complete darkness.

LIDAR systems are more similar to PMD cameras in terms of emitted illumination required in general to calculate the range component. Actually, the PMD sensor operation principle is the same as the Flash LIDAR: usage of a detector array and measuring the range to the object for every pixel. The difference appears in the illumination units: the PMD camera uses LEDs modules and the Flash LIDAR lasers. Laser diods allow to use higher modulation frequiencies and higher optical power. That quality of laser diods incrases the possible measurment range of the Flash LIDAR. LEDs, in its turn, are more cheaper and have a safety eye-class. However, a safety eye-class of LEDs of PMD cameras is important only for on-ground tests and not for real autonomous rendezvous missions. In terms of resolution of the sensor chip there are no big differences, because the Flash LIDARs nowadays have resolutions of 128x128 pixels [134] or 256x256 pixels [135, 36].

Comparing to the scanning LIDARs, the absence of the moving parts in a PMD sensor in order to get 3D measurements helps to avoid additional hardware failures, undesirable artifacts and motion blur during the relative pose estimation. Besides that, the PMD camera along with a range image, provides amplitude image of the picked scene, representing the intensity of the reflected signal from the target. This characteristic will be one of the solution during the pose estimation under the strong sunlight conditions.

2.4 PMD Camera- and Hand-Eye-Calibration

By simulating the autonomous approach using the EPOS facility, the visual sensors (individually or by fusion of several sensors) and the developed pose estimation algorithms can be tested. An accurately calibrated visual camera is a prerequisite in order to extract the information from 2D images and process it for the pose estimation of the target. Like usual mono- or stereo cameras, the PMD camera needs to be calibrated. In this subsection, I consider the camera calibration process as an estimation of the camera model (intrinsic calibration) and position and orientation of the PMD sensor frame in the camera housing (hand-eye calibration) with respect to the breadboard of the Robot 1.

2.4.1 Overview of Calibration Techniques

The well-known calibration toolboxes, such as OpenCV Camera Calibration [136] or Camera Calibration Toolbox (CCT) for Matlab [137] are available in the public domain for estimation of the intrinsic camera and absolute extrinsic parameters. Absolute extrinsic parameters refer to the absolute pose of the camera in the world/object frame. For the current application, I do not use these parameters directly in the thesis, but I mention them in the theoretical part of the intrinsic calibration, since they are tightly connected. In the work of Tzschichholz [51], the author already calibrated the PMD sensor for using it with the EPOS facility. In his work, the CCT for Matlab is applied in order to estimate the intrinsic parameters of the PMD camera. The transformation between the pose of the PMD camera and the Tool Center Point (TCP) of the robots breadboard was determined by involving the knowledge about the relative position and orientation of the other robot in the chain of transformations. The DLR CalDe and DLR CalLab calibration toolbox [138] is proposed in the present thesis as an alternative for the above mentioned methods. This calibration toolbox contains the well-known method of Zhang, Sturm and Maybank [139, 140] for the intrinsic calibration and also algorithm for the hand-eye calibration technique [141]. Critically, this toolbox does not require for the camera to perceive the whole calibration pattern in every image, which on the one hand is convenient for automated acquisition of images and on the other hand increases the accuracy of lens distortion estimation since features are more evenly spread in the images. Using the proposed toolbox the calibration procedure is simplified, in comparision with the technique of Tzschichholz [51]. During the calibration process I was working only with one robot and its data, where the camera is mounted. DLR CalDe and DLR CalLab toolbox was not developed for the PMD sensor calibration and, as far as I know, this is the first time that a PMD camera is calibrated with this technique at EPOS. Therefore, I purposefully give a detailed overview on the included methods of the toolbox and also provide notes to the manual usage of the DLR CalDe and DLR CalLab toolbox. By that, I introduce a new possibility for other users of PMD sensor technologies (or similar ToF sensors) to calibrate this sensor.

2.4.2 Intrinsic Camera Calibration

The proposed calibration toolbox outlines the intrinsic camera calibration approach, which was made by Zhang [139] and Sturm and Maybank [140]. They presented a closed-form solution by linear least-squares techniques for the initialization of the nonlinear optimization [142]. To intrinsic camera calibration parameters belong: focal length (α, β), principal point of the image (u_0, v_0), a skew coefficient γ. In Figure 2.12, I plot coordinate systems involved in calibration process according to the current application. These are camera frame, TCP frame and world/object frame. It is similar setup to the that one presented in the work of Strobl [143].

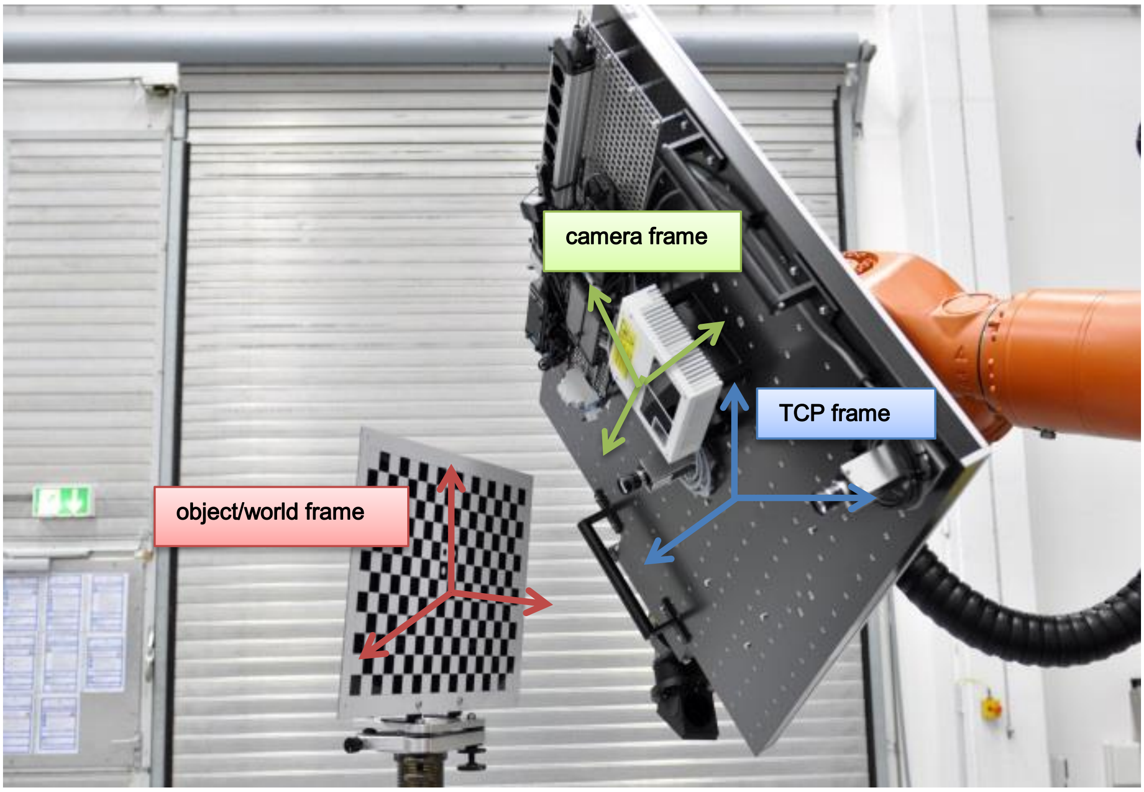


Figure 2.12: The coordinate systems in camera calibration process.

For the accurate camera calibration, one has to detect and identify the visible control point (corners) of the planar calibration pattern

$$ {}_0x_i = [x_i y_i z_i]^T \tag{2.5}$$

projected onto the image frame in every image $n \in (1, \dots, N)$ as ${}_n\tilde{m}_i$ perspectively.

These measured projections are noisy and are compared with the estimated ones

$$ {}_n\hat{m}_i = [{}_nu_i, {}_nv_i, 1]^T \tag{2.6}$$

using the Euclidean decomposition of the perspective projection matrix $P = A\,{}_cT_n^0$ as follows:

$$ s\,{}_n\hat{m} = A\,{}_cT_n^0\,{}_0x_i = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.7}$$

where s is an arbitrary scale factor, ${}_cT_n^0$ the rigid body transformation from the camera frame to the object/world frame in the image n, and A is a camera calibration matrix. Calibration matrix A has 5 unknowns α, β, γ, u_0, v_0, which I am going to determine. A is a homogeneous transformation relating the pose of the object/world frame of the calibration object to the pose of the camera/sensor frame. Since z=0 (because the calibration plate is flat), so that r_3 vanishes and the homography between the calibration plane and the image simplifies to the linear projective transformation $H = [h_1 h_2 h_3]$. The N homographies $\hat{H}_n$ between image projections ${}_n\tilde{m}_i$ and pattern features ${}_0x_i$ can be estimated. Subsequently, the follow equation can be obtained

$$ \hat{H} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}, \tag{2.8}$$

where λ is an arbitrary scalar. Knowing that r_1 and r_2 are orthonormal, I get the following equations:

$$ \left.\begin{aligned} (A^{-1}h_1)^T \cdot (A^{-1}h_2) = 0 \\ (A^{-1}h_1)^T \cdot (A^{-1}h_1) = (A^{-1}h_2)^T \cdot (A^{-1}h_2) \end{aligned}\right\} \tag{2.9}$$

Note that $\omega_\infty = A^{-T}A^{-1}$ describes the image of the absolute conic and the system of equations transforms to:

$$ \left.\begin{aligned} h_1^T \omega_\infty h_2 = 0 \\ h_1^T \omega_\infty h_2 = h_2^T \omega_\infty h_2 \end{aligned}\right\} \tag{2.10}$$

These two equations are taken for every image n, leading to $2N$ constraints for 5 intrinsic unknowns. They can be solved by using a least-square criterion, if at least three different views ($N \geq 3$) are available. Once the camera calibration matrix is known, the absolute extrinsic parameters for every image n are ready to be computed:

$$\begin{aligned}
&r_1 = \tfrac{1}{s} \cdot A^{-1} \cdot h_1 \\
&r_2 = \tfrac{1}{s} \cdot A^{-1} \cdot h_2 \\
&r_3 = r_1 \times r_2 \\
&t = \tfrac{1}{s} \cdot A^{-1} \cdot h_3 \\
&s = \|A^{-1} \cdot h_1\| = \|A^{-1} \cdot h_2\|
\end{aligned} \tag{2.11}$$

2.4.3 Lens Distortion

Up to now I did not consider the lens distortion of the camera, which significantly spoils the linear projective formulation of the camera. Usually the lens systems of the real cameras are affected by nonlinear aberrations. These distortions can be modelled as a symmetric displacement along the radial direction from the principal point [144]. Taking into consideration the radial distortion, the pinhole camera model is extended by an additional distortion model, which is described by a polynomial formulation:

$$\delta_r(\rho) = k_1\rho^3 + k_2\rho^5 + k_3\rho^7 + O(\rho^9) \tag{2.12}$$

where ρ is the radial distance from the center of radial distortion to the expected normalized projection and k_1, k_2, k_3 are the coefficients of the radial distortion [145, 142, 143].

2.4.4 Hand-Eye Calibration

Whenever the sensor is mounted on a robot, it is required to define the rigid-body transformation of the PMD sensor (eye) frame S_S relating to the TCP frame of the Robot 1 (hand) S_R [145]. This problem is referred to as a hand-eye calibration. Thanks to this transformation ${}_ST^R$, we are able to map the sensors measurements into the robot frame for further processing. The common solution of the hand-eye camera calibration is formulated as: move the hand of the robot and observe/perceive the movement of the eye. The mathematical representation is:

$$AX = ZB \tag{2.13}$$

Here X is a desired homogeneous transformation relating the pose of the camera/sensor frame to the pose of the TCP frame of the robot, namely $X = {}_ST^R$; A is a homogeneous transformation relating the pose of the object/world frame of the calibration object to the pose of the camera/sensor frame, which one obtains during the intrinsic calibration phase [139]; Z is the unknown (yet irrelevant) transformation between the object/world reference frame and the base frame of the robot; and B is a homogeneous transformation relating the base frame of the robot and the pose of the TCP frame. At least n=3 stations are required in order to uniquely determine the transformation ${}_ST^R$ [145]. In a nutshell, the rigid body transformation X can be retrieved by minimizing the discrepancies between A and B.

2.4.5 PMD Camera Calibration Process and Numerical Results

The intrinsic and hand-eye calibration techniques, which were described previously, are state-of-the art for monocular and stereo cameras. In this section I show a feasible use of them for the calibration of the PMD sensor within the DLR-Argos3D - P320 camera by using the DLR CalDe and DLR CalLab calibration toolbox. The PMD sensor for the first time was calibrated with the proposed toolbox. A common brief tutorial how to use this calibration toolbox can be found in [138]. I used a rigid checkerboard pattern, which is defined by $n_x = 18$ and $n_y = 12$ squares, where the dimension of every square is specified as $u_x \times u_y \approx 30$ mm × 30 mm. Inaccuracies during the manufacturing or printing of the pattern inherit regularly in the checkerboard and not every rectangle has exactly the mentioned size. Therefore, according to [145], I recalculated the estimated values of the rectangles with the aid of an electronic ruler and acquired the actual size as 29.91 mm × 29.95 mm. The calibration pattern was fixed in front of the robot with the mounted camera on the breadboard plane of the Robot 1, see Figure 2.12.

Robot 1 had been moved to 8 different positions in order to image the calibration pattern. Only the amplitude images of the PMD sensor were used in place of color images of the standard cameras. The follow calibration procedure is the same as the calibration for mono-and-stereo cameras. Nevertheless, throughout available information about PMD sensor calibration with amplitude images, I have not found thorough explanation of the process. Therefore, I suggest it here. In Figure 2.13 eight pictures (2 images from each side) of the amplitude channel are shown.

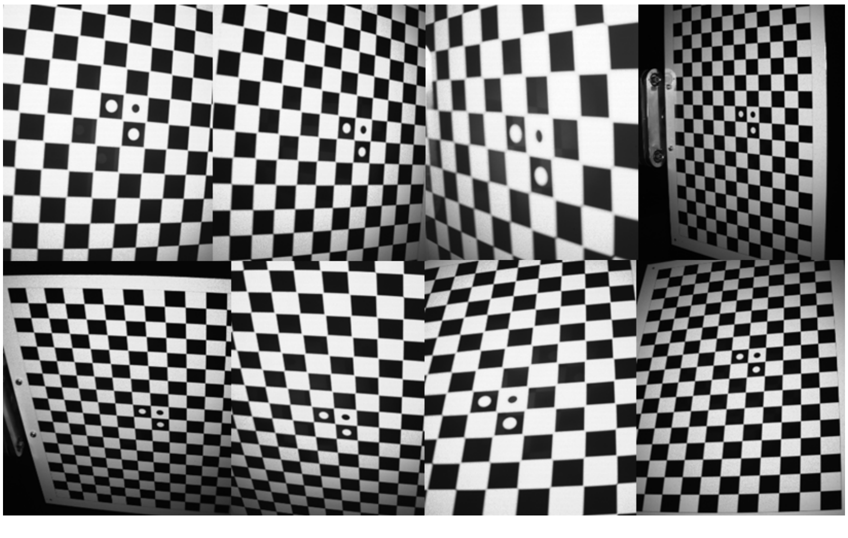
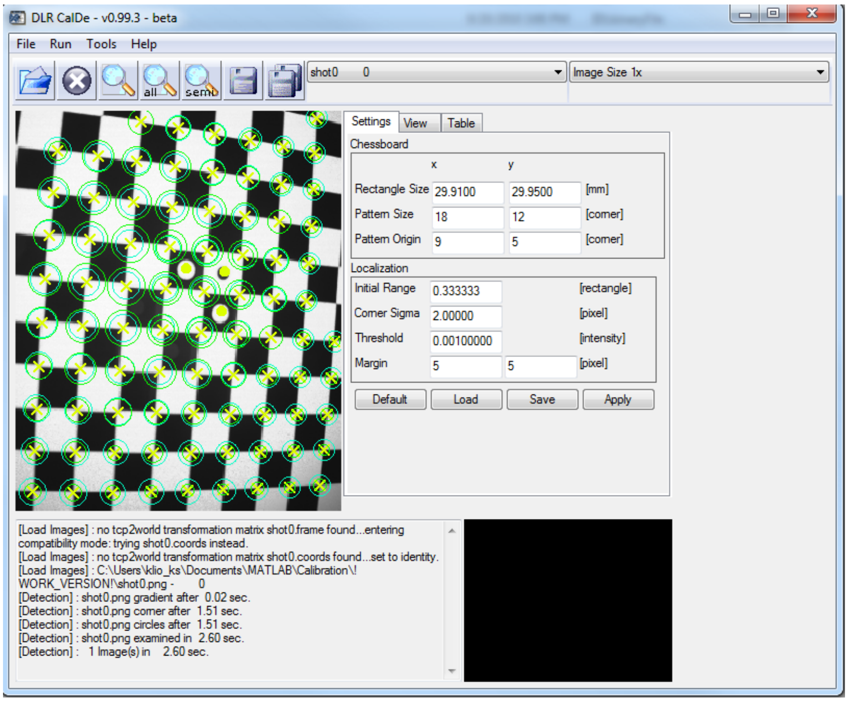


Figure 2.13: Calibration images from the PMD sensor.

There was no need that the whole pattern plate appears in the image, but getting images at a sharp angle is one of the prerequisites for accurate calibration. During the acquisition of the calibration images, the robot pose w.r.t. the fixed Cartesian 3D laboratory frame was stored for every amplitude image respectively.

The information about the robot pose is required in order to calculate the hand-eye-calibration. Having completed the image collection, the DLR CalDe was started in order to localize landmarks and corners on the chessboard with sub-pixel accuracy. It is important to fill up correctly the chessboard parameters on the right side of the main window of the DLR CalDe toolbox (Figure 2.14) in order to let the program detect the corner points precisely and automatically.



Chessboard

	x	y	
Rectangle Size	29.9100	29.9500	[mm]
Pattern Size	18	12	[corner]
Pattern Origin	9	5	[corner]

Localization

Initial Range	0.333333		[rectangle]
Corner Sigma	2.00000		[pixel]
Threshold	0.00100000		[intensity]
Margin	5	5	[pixel]

```
[Load Images] : no tcp2world transformation matrix shot0.frame found...entering
compatibility mode: trying shot0.coords instead.
[Load Images] : no tcp2world transformation matrix shot0.coords found...set to identity.
[Load Images] : C:\Users\klio_ks\Documents\MATLAB\Calibration\!
WORK_VERSION!\shot0.png -       0
[Detection] : shot0.png gradient after  0.02 sec.
[Detection] : shot0.png corner after  1.51 sec.
[Detection] : shot0.png circles after  1.51 sec.
[Detection] : shot0.png examined in  2.60 sec.
[Detection] :   1 Image(s) in    2.60 sec.
```

Figure 2.14: DLR CalDe detects the corner points in the image.

In Figure 2.14 the image with the accurately detected corners of the calibration plate is presented. The recognized points were saved and further used as an initial basis in the DLR CalLab toolbox. To get the intrinsic and hand-eye parameters, one runs the calibration toolbox DLR CalLab. The user can choose and set numerical optimization algorithms and a variety of estimation methods. The calibration process is fully automatic. In Figure 2.15 the main window of the DLR CalLab application with the output results is depicted.

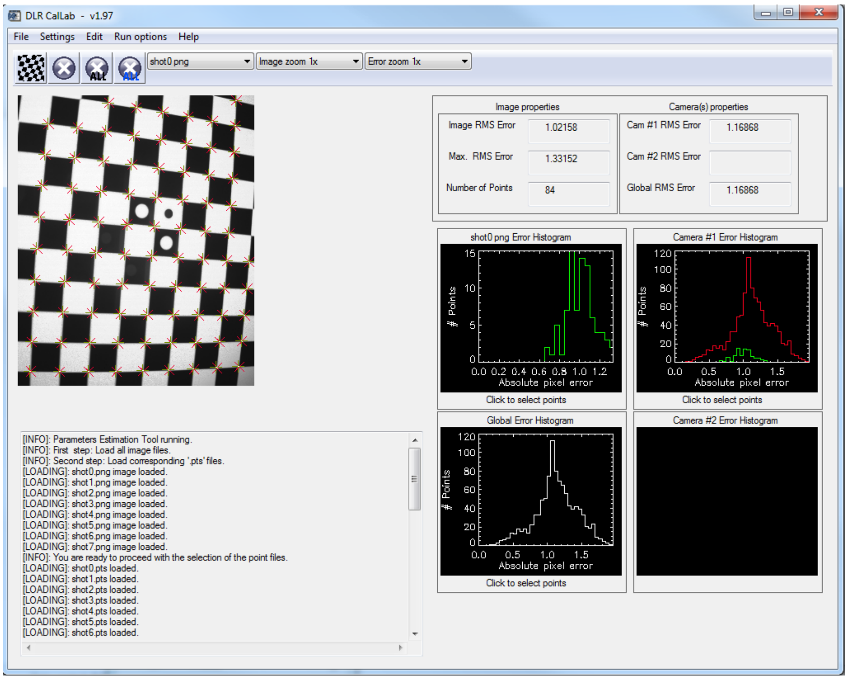


Image RMS Error 1.02158
Max. RMS Error 1.33152
Number of Points 84
Cam #1 RMS Error 1.16868
Cam #2 RMS Error
Global RMS Error 1.16868

```
[INFO]: Parameters Estimation Tool running.
[INFO]: First  step: Load all image files.
[INFO]: Second step: Load corresponding '.pts' files.
[LOADING]: shot0.png image loaded.
[LOADING]: shot1.png image loaded.
[LOADING]: shot2.png image loaded.
[LOADING]: shot3.png image loaded.
[LOADING]: shot4.png image loaded.
[LOADING]: shot5.png image loaded.
[LOADING]: shot6.png image loaded.
[LOADING]: shot7.png image loaded.
[INFO]: You are ready to proceed with the selection of the point files.
[LOADING]: shot0.pts loaded.
[LOADING]: shot1.pts loaded.
[LOADING]: shot2.pts loaded.
[LOADING]: shot3.pts loaded.
[LOADING]: shot4.pts loaded.
[LOADING]: shot5.pts loaded.
[LOADING]: shot6.pts loaded.
```

Figure 2.15: DLR CalLab dialog window with output results.

By running the calibration process, the following results for the PMD sensor inside the DLR-Argos3D - P320 camera have been obtained: Calibration matrix is presented as:

- $A = \begin{bmatrix} 705.748 & 0.581 & 143.578 \\ 0 & 704.082 & 184.228 \\ 0 & 0 & 1 \end{bmatrix}$
- Distortion coefficients $k_1 = -0.4973$ and $k_2 = 0.3251$.
- The transformation matrix ${}_S T^R$, which describes the pose of the sensor inside of the DLR-Argos3D - P320 camera relating to the chosen TCP point of the Robot 1:
  $${}_S T^R = \begin{bmatrix} 0.9999 & -0.0043 & -0.0022 & -42.874 \\ 0.0043 & 0.9999 & 0.0134 & -186.912 \\ 0.0021 & -0.0134 & 0.9999 & 145.847 \end{bmatrix}$$
  The measurement unit for the translation vector is mm.
- A root mean square (RMS) calibration error after intrinsic calibration of 0.177 pixels; after hand-eye calibration it reads 1.192 pixels or 0.21° and 2.92 mm. Note that the former pixel RMS error is not explicitly minimized but the latter position and orientational errors of the robot manipulator as detailed in [145]. In doing so, the reprojection error in pixels might be slightly worse, but the estimation of the hand-eye transformation is optimal since the actual, biggest errors in the system (viz. the positioning errors of the robotic manipulator) are being minimized.

The calibration process with the current DLR-Argos3D Camera was done once, since the camera was fixed mounted on the plate during the whole research phase.

Chapter 3

Pose Estimation

3.1 Introduction to Pose Estimation

The main scope of this thesis is computer vision for the space robotic applications, namely, estimating position and orientation of the target spacecraft relative to the camera by extracting information from an image or sequence of images. The first problem refers to the recognition/detection of the object and the second one to the frame-by-frame tracking of the known object in the sequence of the images. The problem of pose estimation considered in the current thesis belongs to the model-based pose estimation problem since I take into account the knowledge of a 3D model of space debris or the satellite with demand to be repaired. The 3D model is stored on board of the active chaser during the whole mission. In cases, when the inspection is a part of the planned space mission, the model can also be built on board as the target appears in the FOV of the chasers visual system but this task is not considered in this thesis. Both of these problems are connected together in the sense that frame-by-frame tracking can not be started until an initial guess of the target's position is provided to the navigation system. Initial guess can be acquired by the same sensor, which is used for tracking, or provided from other sensors mounted on board (e.g. GPS, radar or other visual sensors). In my case, when I deal with a non-cooperative target, the lack of a GPS receiver on the target spacecraft forces us to use different sensors and techniques for the pose acquisition. Therefore, the object recognition and pose acquisition is a necessary task by RvD with non-cooperative targets. Moreover, in case of the loss of the target during the tracking in a sequence of images, one would need a re-initialization of the vehicle's pose in order to continue with a safe approach. In the next sections I propose pose initialization of the target with the depth image of a PMD sensor without prior knowledge of position and orientation. As soon as the pose has been initialized I employ the hybrid navigation technique, which exploits fusion of completely different pose estimation techniques for 3D depth and also 2D amplitude images of a PMD sensor.

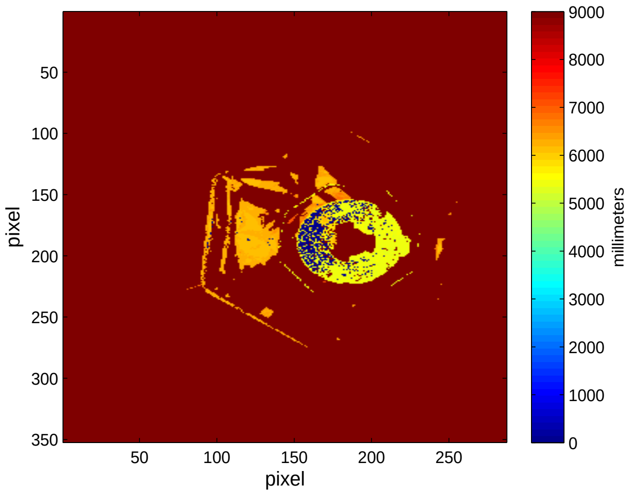
In this thesis, primarily I use only the depth image from the PMD sensor in order to estimate the initial pose. The reason is follow. The pose initialization technique with the PMD sensor was one of the first research questions during the research phase. In that moment, the idea of using the amplitude image for the pose estimation had not appeared yet. Nevertheless, in Chapter 4, I also present the possibility to use a 2D amplitude image for a follow-up improvement of the calculated initial pose with a depth image.

Similarly with the tracking using depth images, one needs to know the initial guess in order to start frame-to-frame tracking in a 2D image sequence. The hybrid navigation technique in Section 3.3 is proposed, which is based on distributed fusion architecture using only a single PMD sensor.

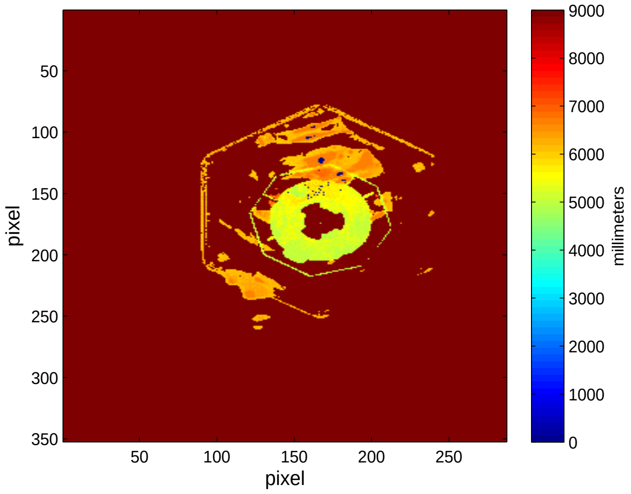
3.2 Pose Initialization

In Section 1.4.3, an overview of the state-of-the-art pose estimation techniques with 3D point clouds was given. In this section, I propose one of the local methods for the object recognition and pose estimation that is applied in order to calculate the initial unknown pose of the non-cooperative target. The initial pose estimation algorithm based on the Point Pair Feature descriptors using ToF cameras is outlined in the work of Drost et al. [65] and Papazov et al. [146]. I provide the explanation of the choice below. For this method it is presumed that the model and the scene are presented in the form of oriented points (e.g. vertices and normal vectors in every vertex): $m \in M$ points belong to the model and $s \in S$ points belong to the scene, respectively. Firstly, the global description model is calculated, which includes point pair feature vectors for different combinations of the model's vertices. Secondly, scene descriptors are calculated when the data arrives from the sensor. This global model is used further for finding corresponding matches between a set of the scene's and model's descriptors. When the matches have been found, the voting scheme and clusterization processes for the best pose of the object are involved. By varying some initial parameters of the algorithm, it was noticed, that the pose refinement process is still necessary, because discrepancies of the rigid transformation between the model and the scene point clouds exist.

The choice of used features, such as point pairs, has some advantages in comparison with other techniques when working with a PMD sensor. First of all, this is a local method, which in general leads to more concrete object identification assuming that the target is in the field of view. The global features, in their turn, are giving the answer if the spacecraft is presented in the scene or not. Furthermore, the point pair features can be extracted for different complex models (e.g. the mockup used in this work) and not only for some primitives objects. By a complex model we should also understand non-cooperative spacecrafts, which have lost some structural parts and we do not know anything about it when mission has already started. It means, the known 3D model on board of the chaser can be different from that of the target spacecraft located in space. Moreover, we need to take in account other factors: illumination conditions, surface of the target material and the approach scenario. Depending on the listed circumstances the quality of the range data received from the PMD sensor is inconsistent, meaning that noise will deteriorate the point cloud by adding false points, see Figure 3.1 (b), or even completely exclude some part of the target from a scene point cloud, as one can see in both images of Figure 3.1.



(a)



(b)

Figure 3.1: (a)-(b): Depth images with some missing depth information of the mockup's parts. Image (b) reflects false distance information of some parts outside of the hexagon borders, which are absent in reality.

It is very important to find an appropriate algorithm, which is able to cope with these problems. These justifications pushed us to create the pose acquisition technique with point pairs features which, actually, provides good results for the current application.

3.2.1 Point Pair Feature Vector

Let us consider m_i as a first and m_j a second point throughout all model points M. Additionally, each of the points has a normal vector - n_i and n_j. Let us discuss in more details, how we get the normals in every point.

Normal Vectors for Oriented Points

The process of the estimation of the normal vector in every point of the model, e.g. m_{any}, will be directly connected to the information about N_{neigh} neighbor points around the point m_{any}. One of the methods for estimating the surface normal at a point m_{any} is to compute the eigenvectors of the N_{neigh}-neighborhood surface patch [147]. The eigenvector corresponding to the smallest eigenvalue gives us the surface normal at a point m_{any}.

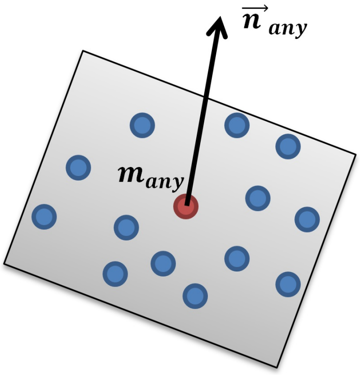


Figure 3.2: Calculation of the normal vector in any point of the point cloud.

Point Pair Feature

In Equation (3.1),

$$F_m(m_i, m_j) = (f_1, f_2, f_3, f_4) = (\| d \|, \angle(n_i, d), \angle(n_j, d), \angle(n_i, n_j)) \tag{3.1}$$

the point pair feature vector for the model F_m is defined as a four component vector and composed by the Euclidean distance between two points $\| d \| = \| m_i - m_j \|$, as well as an angle between vector d and the normal n_i, an angle between vector d and the normal n_j, and an angle between two normal vectors n_i and n_j.

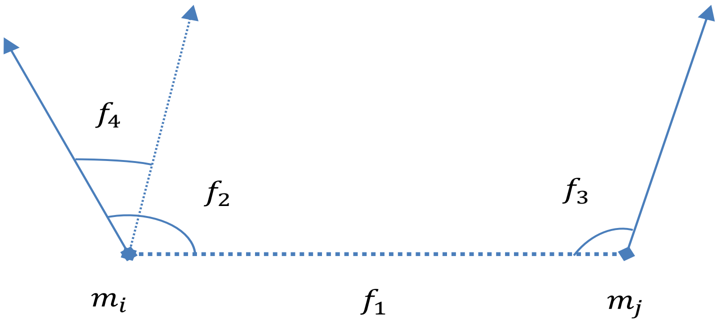


Figure 3.3: Illustration of the point pair feature vector.

Vector F_m and the components from the Equation (3.1) are depicted on Figure 3.3.

The distances and angles are sampled in steps by setting parameters d_{angl} and d_{dist} as follows: $d_{angl} = \frac{2\pi}{n_{angl}}$ with an angle sampling value n_{angl}, which refers to the number of angle bins to be quantized; and d_{dist} dependent on the sampling rate τ_d and model diameter $d_{dist} = \tau_d \cdot diam(Model)$. The model diameter is defined as the Euclidean norm of a vector constituted by the differences between maximum and minimum values of the vertices coordinates. In my experiment, the parameters have following values $n_{angl} = 90$, $d_{angl} = \frac{2\pi}{90} = \frac{\pi}{45}$, $\tau_d = 0.03$, $d_{dist} = \tau_d \times diam(M)$.

3.2.2 Description of the Initial Pose Estimation Algorithm

As it was already mentioned, I have to construct the global description model, by calculating the point pair descriptors. Here, the global description model is presented in the form of a hash table, where the feature vector F_m is used as a key. The model can contain equal feature vectors $F^a_m = F^b_m = F^c_m$, e.g. depicted in Figure 3.4. The pair of points (m^a_i, m^a_j),(m^b_i, m^b_j) and (m^c_i, m^c_j) have the same relation between them.

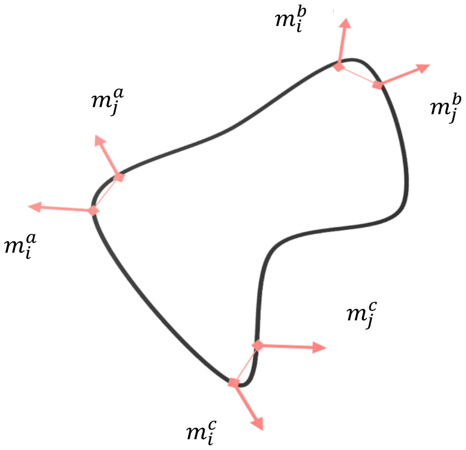


Figure 3.4: An example of similar point pairs of the model.

Reasonably, feature vectors were put together, which are equal to each other, in the same cell reducing the dimension of the hash table. Usually, in one cell of the hash table there is more than one pair of points of the model.

Having created the hash table, the global model is created. Since the point cloud S has been obtained with a PMD camera, the point pair feature vectors F_s (s_i, s_j) are calculated as well for the pairs of points (s_i, s_j). As we opted previously, the feature vectors F_s are the keys for the hash table. All pairs (m_i, m_j) from the hash table are extracted, if the feature vector of the model F_m is equal to the feature vector of scene F_s. This chain of successive actions is illustrated in Figure 3.5.

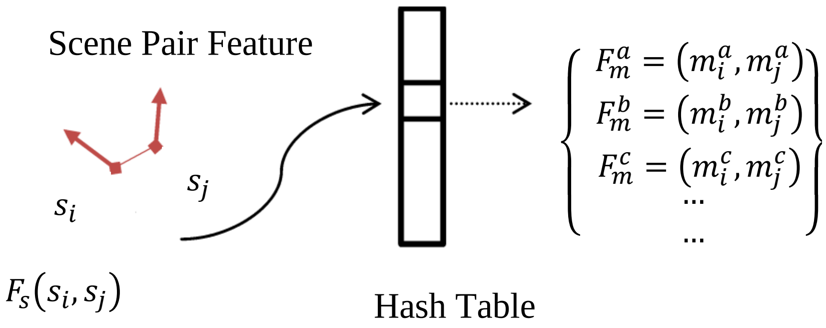


Figure 3.5: An extraction of the similar pairs of points for the scene and model.

Once the match between two pairs of points exists, one is able to compute the rigid transformation that aligns the model to the scene. In the work of Drost et al. [65], the authors introduce the meaning of the local coordinates. In Figure 3.6, the local coordinate system with axes x, y and z is presented. Local coordinates are pairs of the form (m_i, α) with respect to reference point s_i, where α denotes the rotating angle and m_i is a reference point on the model. For local coordinates, one can derive the rigid transformation. The transformation that aligns the model with a scene is computed as follows:

$$T_{m\to s} = T_{s\to g}^{-1} R(\alpha) T_{m\to g} \tag{3.2}$$

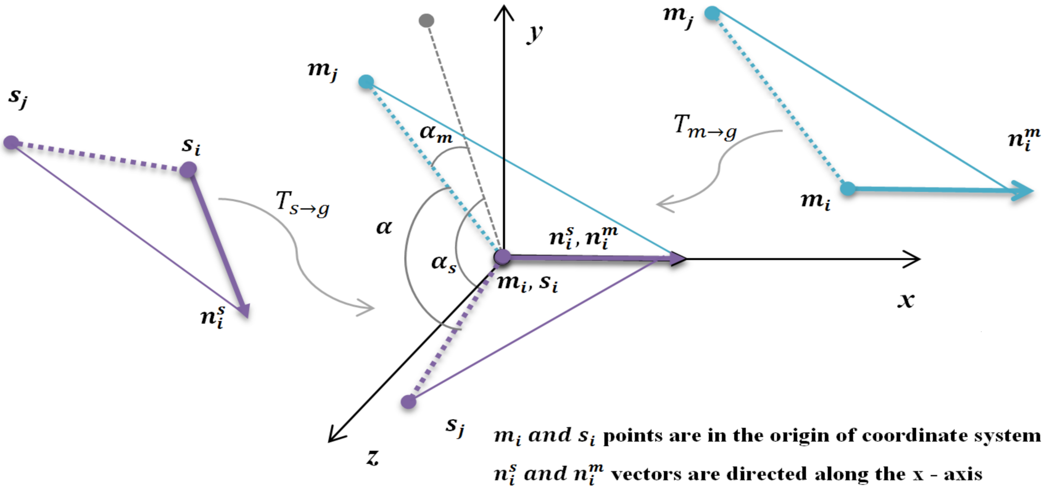


Figure 3.6: A transformation that aligns the model with the scene.

In the Equation (3.2), the transformation $T_{s\rightarrow g}$ moves the scene point s_i into the origin and aligns its normal n_i^s with the x-axis, see the Figure 3.6. The same procedure $T_{m\rightarrow g}$ is implemented for the model reference point m_i and the corresponding normal vector n_i^m allocated to that point. In order to complete the final transformation $T_{m\rightarrow s}$, one more transformation $R(\alpha)$ is needed to align two left points, m_j and s_j. The rotating angle α is determined as follows:

$$\alpha = \alpha_m - \alpha_s, \tag{3.3}$$

where α_m is the angle between the vector $m_i - m_j$ and an upper xy half-plane of the intermediate coordinate system and α_s is the angle between the vector $s_i - s_j$ and the upper xy half-plane of the intermediate coordinate system (see Figure 3.6). The dashed gray line in Figure 3.6 divides the xy plane for upper half-plane and lower half-plane. To find the best local coordinates at a specific point s_i, the number of points in the scene lying on the model has to be maximized. For that purpose Drost et al. [65] suggested to use the 2D accumulator massive, where the hypotheses vote for the local coordinates. The rows of the accumulator massive correspond to the reference points of the model m_i and the columns correspond to the sampled rotation angles α. The size of the accumulator massive is $N_m \times N_{angle}$, where N_m is equal to the number of the model points m_i and N_{angle} is the number of sample steps of the rotation angle α.

When the voting process takes place, all point pairs corresponding to the model (m_i, m_j) are retrieved from the cell for each calculated point feature vector F_s. Using Equation (3.3), the rotation angle α is computed. After completing these steps, it is supposed that we have local coordinates $(m_i,\ \alpha)$, which can move $(m_i,\ m_j)$ to $(s_i,\ s_j)$. The vote (vote is a number) for this hypothesis is thrown in the related cell of the local coordinate $(m_i,\ \alpha)$ in the accumulator massive. It should be noted, that every voting result has a certain number of votes. When all point feature vectors of the scene F_s are processed, the set of local coordinates with the highest number of votes is retrieved from the accumulator massive. The transformations from the model to the scene coordinate frame are calculated for each returned local coordinates.

Consequently, knowing the series of object poses from each reference point, the final pose must be extracted. For that purpose, the object poses are grouped in one cluster if they do not differ in translation and rotation components more than predefined thresholds. For the translation component the threshold is taken as $1/10^{th}$ of the model diameter, $t_{thresh} = 0.1 \times diam(M)$, and for the rotation component as $r_{thresh} = 2 \times \frac{\pi}{30}$. Poses collected in each cluster are averaged and the votes are summed up. The clusters with the highest scores are considered as the applicants for the best estimated relative pose.

3.2.3 Initial Pose Refinement

I include the pose refinement after pose initialization, since in the most cases the extra correction step significantly improves the calculated pose. Once the pose has been obtained by the local method presented in the previous sections, the point cloud of the model can be transformed to the point cloud of the scene. In order to revise the transformation, the Iterative Closest Point (ICP) algorithm [68] was executed, similarly to the works of Hinterstoisser et al. [148] and Birdal et al. [71]. A particular overview of the ICP algorithm will be provided in Section 3.3.1, since it was involved into frame-to-frame tracking for the rendezvous phase.

3.3 Hybrid Navigation Technique

After the coarse estimation of the initial pose of the target spacecraft, the process of frame-by-frame tracking of the target vehicle in the sequence of PMD images takes place. In general, the aim of a frame-by-frame tracking is to measure the change of object's pose in each of two consecutive image frames. The important hint for pose localization in a new arrived frame is provided in form of the known previous pose. Different approaches described in Sections 1.4.3 - 1.4.5 can be employed depending on the sphere of application. Concerning OOS and ADR the main criteria for the choice of the tracking technique are the position of the spacecraft on the orbit together with illumination conditions (sun angles, eclipse, etc.), and also the target's material. Moreover, one of the major aspects for the tracking is reliable visual information or some strong features of the target available during the whole rendezvouse phase even when the obtained images are incomplete because of some occlusions, image noise or blur, etc.

Considering the nature and possibilities of the used PMD sensor, namely, co-registered depth and amplitude images, I propose a tracking architecture with a hybrid nature. In general, there are different classifications of the fused techniques depending on the amount of sensors and relations between each other [149]. The best known are: complementary, where the measurements from different sensors complement each other, because of impossibility of both of them provide the fully needed information of the scene; redundant - the same information of the target is provided and fused in order to increase the reliability of the measured system. In his thesis, the distributed architecture with redundant measurements but without hardware redundancy for the state estimation is proposed.

The idea of creating of a novel hybrid pose estimation technique for the frame-to-frame tracking came across throughout a deep analysis of the features and capabilities of the DLR-Argos3D - P320 sensor. By simulation of different RvD scenarios with a non-cooperative target (mockup) in the EPOS laboratory, the full or partly incompleteness of the depth measurement of the scene (see Section 2.3.2) was discovered. Estimation of 6 DoF of the spacecraft vehicle with a non perfect depth image presents a problem, since the estimated state vector can deviate from the real one and create large errors. So, what I propose is the combination of two different pose estimation techniques for redundant real-time estimation of the target's state vector without additional sensors. The hybrid navigation pipeline employs two completely different model-based pose estimation techniques to improve continuous estimation of the target's pose without a break or abruptions during the close range approach. The depth and amplitude measurements from the PMD sensor are processed independently in order to estimate the local state vectors before they are sent to the fusion node. The sketch of the proposed architecture is illustrated on Figure 3.7.

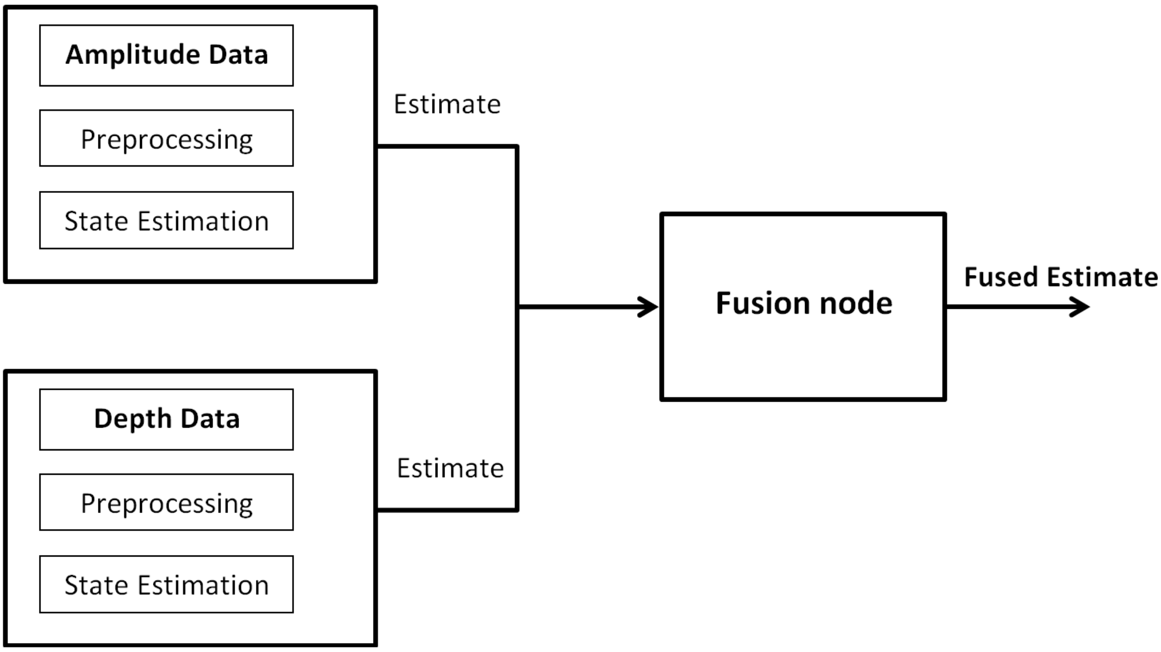


Figure 3.7: Architecture of the Hybrid Navigation Technique.

The first method estimates the pose of the target spacecraft from the depth raster image using the Iterative Closest Point algorithm (ICP) with a reverse calibration technique. The proposed pipeline is outlined in Section 3.3.1. The second method (explained in Section 3.3.2) consists of an image processing procedure that extracts low-level features of the target spacecraft from the 2D amplitude images. The straight lines are detected with a Hough Line Transform and thereafter, the end points of the lines are defined. The current target pose is estimated by fitting the projection of the known 3D model on the detected features with Gauss-Newton minimization problem. In order to get the final fused pose, I combine the best local measurements from two pose estimation techniques with weighted average technique. This solution should balance the pros and cons of the proposed completely different approaches, based on the experience obtained during the test simulations with real data. The description of the technique in a fusion node is explained in Section 3.4.

The great advantage of the proposed architecture is no need for additional measurement sensors. Additionally, by distributed architecture the measurements are not needed to be aligned from the both channels of the sensor. The combination of the standard, very common techniques for pose estimation guarantees a stable tracking tacking into account all challenges of the PMD sensor.

3.3.1 Autonomous Rendezvous using 3D Depth Data

The tracking algorithm with the depth images must be stable to work without abruption even when the completeness of the raster image content can not be guaranteed during the whole sequence of the images. By completeness of the raster image I assume the missing depth data or worng measurements as shown in Figure 3.1. Working with a non-cooperative target I rely only on the known 3D model of the target spacecraft likewise by the pose acquisition. During the tracking phase there is no need to consider the whole 3D model, but good visual features describing the object is the prerequisite to the accurate and fast estimation of the relative attitude and position. In the next subsection, one of the state of the art methods, namely Iterative Closest Point (ICP) algorithm [68], is proposed. The emphasis by applying this algorithm in this doctoral thesis aims at using the raster nature of the PMD depth images. The organized point cloud is the prerequisite to use one variation of the diversity of ICP algorithms - ICP algorithm with reverse calibration. The organized point cloud refers to the knowledge of the 2D position in the depth image of every observed 3D point. Therefore, the advantage of the proposed ICP algorithm with reverse calibration is that I could decrease the assigned task complexity of the computational part by the transfer from 3D to 2D space. Why did I choose this algorithm and not another one? This is mainly because the ICP with reverse calibration doesn't rely on the distance information from the previous frame. I don't try to find the coherence between the chosen visual features in the current and the previous frames since it can cause big discrepancies if some of the depth data in one of the frames is missing or totally noisy. Here the tracking assumes information about the pose in a previous frame.

Iterative Closest Point

In the work of [150] the author summarized some different derivations of the ICP algorithm with the best computational efficience [151]. However, the central idea to find the transformation between two registered points sets remains unchanged.

Let us determine the first data set as model point cloud $A = \{a_1 \dots a_{N_A}\}$, which should be aligned to the second data set, namely scene point cloud $B = \{b_1 \dots b_{N_B}\}$. According to the work of Rusinkiewicz [150], the standard ICP algorithm can be divided into six stages:

- selection of a subset of the points in point clouds A and B;
- matching those selected points;
- weighting the corresponding pairs appropriately;
- rejecting certain points based on distance;
- assigning the error metric based on the point pairs and minimizing the error.

One of the time consuming part throughout the above mentioned steps is the selection of the corresponding points between two point sets. This part concludes the search of the nearest neighbor, where Euclidean distance is calculated for every point in the first set $a_i \in A$ and the second data set $b_i \in B$. The neighbor point b_j from the $b_j \in B$ is found for every point a_i when the distance between point a_i and the point from the set B is the smallest. In the work of [150], the author outlined the neighbor search by using the following techniques:

- neighbor search by using a k-dimensional (k-d) tree (as acceleration of the computation);
- finding the intersection of the ray originating from the source point in the direction of the source point's normal with the destination surface (approach called as "normal shooting");
- project the source point onto the destination mesh, from the point of view of the range camera and provide the corresponding search directly or by limiting the neighborhood and provide local search for compatibility (approach called as "reverse calibration")[152]. The neighbor search is in 2D space.

Using the matrix depth data structure of the raster PMD image, it is assumed that the acquired point cloud corresponds to an organized point cloud. On this basis, the efficient version of neighbor search - "reverse calibration" is presented in more details further. I found this technique appropriate for the target tracking, because of the following aspects: a previous coarse pose estimation after the pose initialization phase exists; the neighbors computing search is not overloaded with many outliers; the computing efficiency is high.

Nearest Neighbor Search using Reverse Calibration Technique

The most efficient way to search the neighbors for the sets of points is inside of a limited space, i.e., only in a certain region of interest. This space can be determined by the search radius on the 2D area pointed from the source point. The search radius should be selected depending on the quality of the raster image, meaning that one can choose a small radius if the scene is presented quite completely. It will be sufficient to find the neighbors. However, when some data of the scene is missing, the radius should be increased in order to expand the nearest neighbor search area. Figure 3.8 depicts the illustration of this nearest neighbor search.

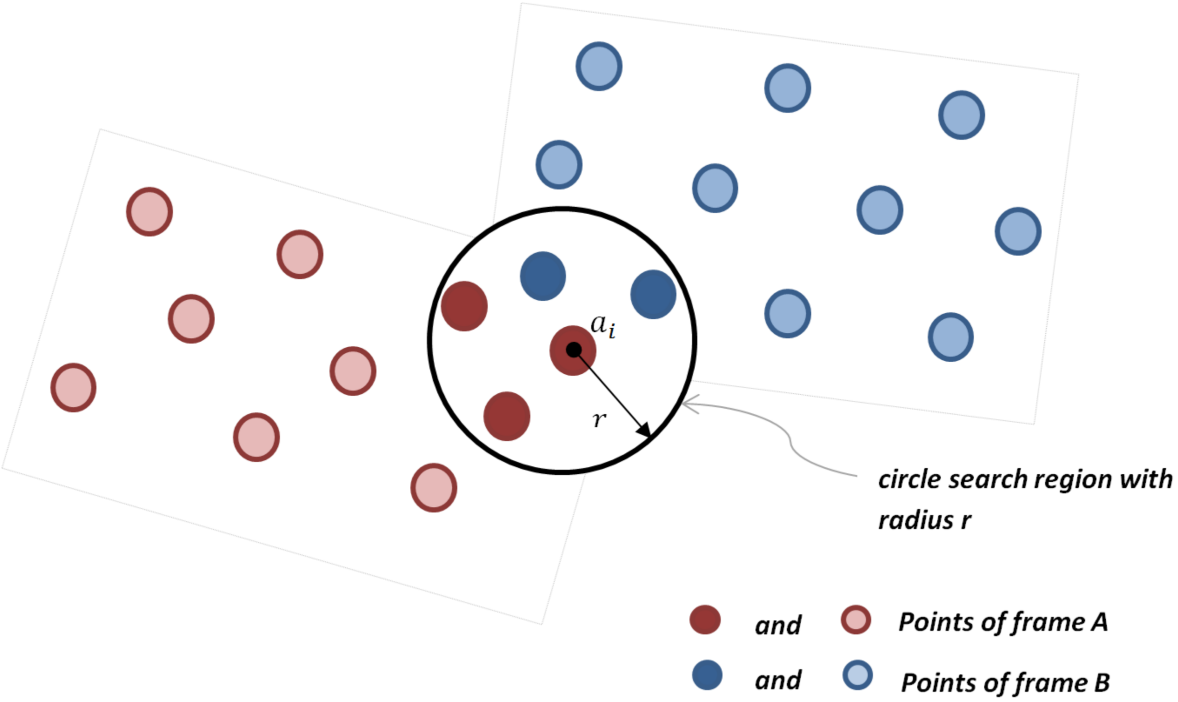


Figure 3.8: 2D Circle search region with a radius r around the point a_i.

Taking the point a_i from the point set of the frame A and using a certain radius, at least one neighbor in the point set of the frame B can be found. One can start with a certain radius and steadily increase it until at least one neighbor will be found. Here the term "reverse calibration" is used intentionally, because the following task for the 2D circle determination around every interested point for the neighbor search can be solved by projection of the 3D point onto the image plane if the calibration matrix of the sensor chip is known. The PMD sensor of the DLR-Argos 3D-P320 camera has been calibrated in EPOS with the technique presented in Section 2.4. The calibration matrix was accurately calculated in the subsection 2.4.5.

Every point a_i from the point set of the frame A can be projected onto the image plane of the frame B, so the 2D point in a frame B in form of a pixel will be determined (see Figure 3.9). The obtained pixel with coordinates (p_x, p_y) indicates the starting point and the chosen radius determines the region of interest, where the nearest neighbor search should be conducted. The region of interest is defined symmetrically around the pixel (p_x, p_y). But in case of points located near the sensor's borders, the search area will be limited from one or another side by the absence of the pixels in an image area. In this case, the nearest neighbor will be absent.

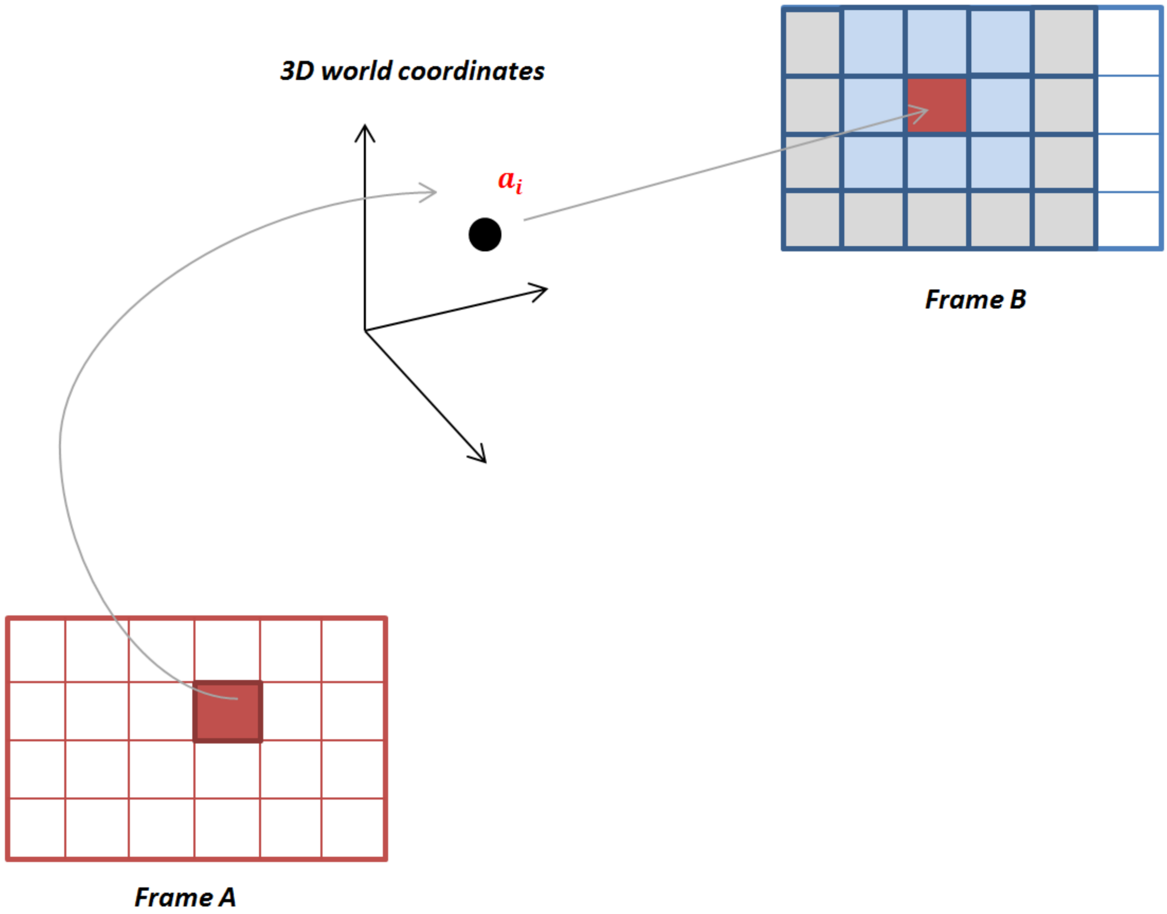


Figure 3.9: Projection of the 3D point to the image plane.

Estimation of Rigid Body Transformation via ICP

The pose estimation between two point sets for every frame l in a sequence of images can be calculated throughout transformations (R_l, t_l), where t_l is the relative position vector and R_l is the rotation matrix. The following main stages are included in the ICP routine in order to minimize the point-to-point distance:

1. The initial transformation (R_0, t_0) provides the coarse alignment between the model and the target at a counter $k = 0$. It also narrows the search field for the neighbor points for the tracking algorithm.

2. Perform nearest neighbor search by using reverse calibration technique with radius r in a 2D space.

3. Find correspondences between points from the frame A (e.g., points a_j, $j = 1, ..., N$) and frame B (e.g., points b_j, $j = 1, ..., N$).

4. Compute the rotation R_l and translation t_l that minimize the point-to-point error function in Equation (3.4) by applying the closed-form solution to the least square problem [153] for the corresponding points from step 2. I describe a step-by-step

procedure of the closed-form solution at the end of this section

$$\min_{R,t} \sum_{i=1}^{N} \|b_i - (Ra_i + t)\|^2. \tag{3.4}$$

When the minimum of Equation (3.4) is computed for the first time, the iteration process from the step 2 is continued.

5. The ICP algorithm terminates when the registration error in Equation (3.4) is less than the predefined threshold value θ_{error}, or when the number of maximum iteration steps k is exceeded.

In order to ensure that the iterative algorithm converges into the global minimum, it is necessary to provide a correct initial guess to ICP. Large errors in the initial guess cause incorrect correspondences between the model and the measured points and as a consequence, the algorithm can be easily trapped into the incorrect first local minimum and an expected pose of the target will be incorrect [154, 155]. In fact, a pose initialization routine can be employed to estimate the pose of the target spacecraft without guess, as described in Section 3.2.

Closed-form Solution of Absolute Orientation

In order to complete the step 4 in the ICP routine, I apply the closed form solution to the least square problem. Here, I give an overview of the closed-form solution to the least-squares problem for three or more points described in [153]. This solution provides in one single step the best possible transformation between two sets of points in different coordinate systems. For the representation of rotation the author uses unit quaternions, where the solution for the desired quaternions is shown to be the eigenvector of the symmetric 4x4 matrix associated with the most positive eigenvalue. A unit quaternion is closely allied to the geometrically intuitive axis and angle notation. According to the carefully analyzed algorithm in [153] the main steps to find the closed-form solution are as follows:

- knowing two sets of corresponding points, e.g., a_j, $j = 1, ..., N$ and b_j, $j = 1, ..., N$, one finds the centroids r_a and r_b.
- Thereafter the centroids are subtracted from all measurements, so that one deals with the measurements relative to the centroids.
- For each pair of coordinates the nine possible products of the components of the two vectors are computed $x_a^{'} \cdot x_b^{'}$, $x_a^{'} \cdot y_b^{'}$,...,$z_a^{'} \cdot z_b^{'}$. The products of coordinates are summed up in order to obtain S_{xx}, S_{xy},...,S_{zz} as:

$$S_{xx} = \sum_{i=1}^{N} x_{a,i}^{'} x_{b,i}^{'} \qquad S_{xy} = \sum_{i=1}^{N} x_{a,i}^{'} y_{b,i}^{'} \qquad ... \quad S_{zz} = \sum_{i=1}^{N} z_{a,i}^{'} z_{b,i}^{'} \tag{3.5}$$

These in total nine components contain all necessary information in order to find the solution. Next step is to create the 4x4 symmetric matrix P by combining the sums of the product from the previous step.

$$P = \begin{bmatrix} (S_{xx}+S_{yy}+S_{zz}) & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & (S_{xx}-S_{yy}-S_{zz}) & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & (-S_{xx}+S_{yy}-S_{zz}) & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & (-S_{xx}-S_{yy}+S_{zz}) \end{bmatrix} \tag{3.6}$$

It is required to calculate the coefficients of the fourth-order polynomial in order to obtain the eigenvalues of P. By solving the quartic equation, the most positive root is used to obtain the corresponding eigenvector. The quaternion representing the rotation from one coordinate system to another is a unit vector in the same direction.

- The translation component is computed as the difference between the centroid of one point cloud and the scaled and rotated centroid of the other point cloud.

3.3.2 Pose Estimation with 2D Gray-Scaled Images

Throughout the variety of the model-to-image registration techniques in computer vision using monocular cameras (see Section 1.4.4), here the focus is on designing the pose tracking technique based on feature correspondences between the 3D wireframe model and the detected features in the 2D image. The choice of the image features is a very important prerequisite for the accurate pose estimation. The image features tend to be chosen on the grounds of simplicity and noise insensitivity in order to be recognizable in every image frame. The lack of visual markers on the target satellite complicates the problem of pose estimation with monocular vision. Moreover, one needs to keep in mind that even in cases when the depth data is (partly) lost and only the redundant amplitude image information exists, the quality of the amplitude image can be restricted. The restriction appears because amplitude images contain also some occlusions, image noise and blur. These factors complicate the process of feature extraction. Using the prior known pose, the search space for the correct feature correspondence can be narrowed, when a new frame arrives.

In this work, the model-to-image registration based on two types of low-level features is proposed. First, I detect straight lines and thereafter end-points of those lines. The detected lines construct the frontal hexagon and octagon, and also parallelogram, when the servicer approaches from the side of solar panel. Once a new image is acquired by the sensor, the edges of the mockup from the image are obtained using standard edge detection algorithms such as Canny detector [156]. Subsequently, the Hough Transform technique is employed to detect the straight lines [157].

Throughout the whole investigation phase I found out that this approach is very suitable when working with the existing mockup, since it has accurate straight contours to be tracked. Moreover, the edges of the spacecraft mockup can be distinguished and localized in the gray-scaled amplitude image even under different illumination conditions. The great advantage of using the Hough Transform is the resistance of the technique to short line gaps (e.g. straight line is broken) due to noise or some occlusions. Despite this, the line will be still detected.

The template matching is not considered since it can happen that the generated template will not contain key points determined in the arrived image and it leads to false or inaccurate pose estimation. I am also not going to pay attention to optical flow methods in this thesis. First, because it is always required to have two consecutive frames for the pose estimation, and they should not be saved on board. I want a pose estimation algorithm for the 2D image to be completely independent on the previous image itself, so that I am taking into account only the knowledge about the estimated previous 6 DoF pose. Second, because of the low resolution of the PMD sensor chip, a small motion between frames based on the pixel displacement can not be properly distinguished. The CNN are omitted in the thesis because of following reason. The target spacecraft could be very different from what we are expecting, e.g. there are some damages. This information we only get after the mission already have been started. In this case, the training data set of the CCN will not perform well for the real images during the rendezvous. The training data set is not well constructed and does not contain the real features of the target.

Standard Hough Line Transform

The Hough transform can be used in order to identify the parameters of a curve which best fits a set of given edge points. A line can be represented by the equation

$$y = m \cdot x + b \tag{3.7}$$

where the parameter m is a slope of line and b determines the point, where the line crosses the y-axis. By describing the line in a polar coordinate system the line can be detected using the pair (ρ, θ).

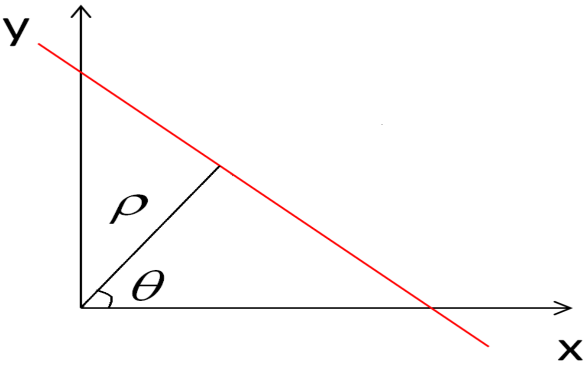


Figure 3.10: Line representation in the image space.

The parameters θ and ρ are the angle of the line and the distance from the line to the origin.

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \tag{3.8}$$

A line in a two dimensional Hough space (space determined by θ and ρ) is represented by a single point, corresponding to a unique set of parameters (θ_0, ρ_0), see Figure 3.11.

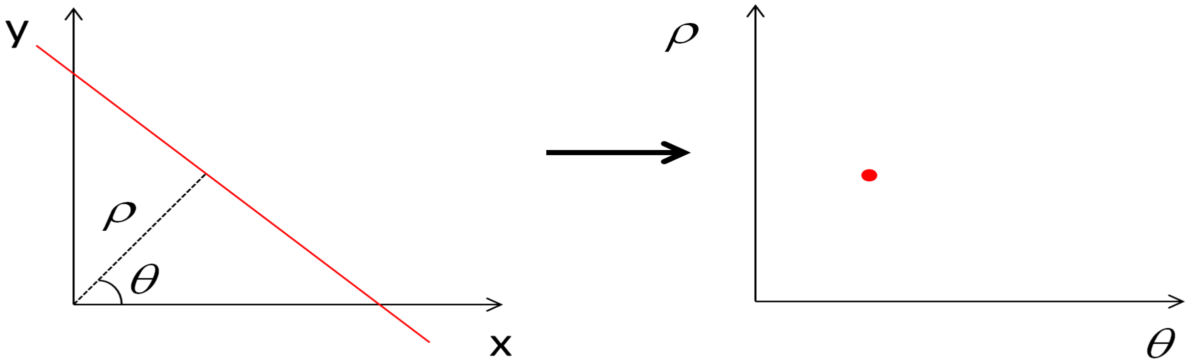


Figure 3.11: Straight line in the image space is presented as a point in the Hough space.

Drawing multiple lines intersecting at one common point (x_0, y_0) in the image space will result in a continuous sinusoid in Hough space (see Figure on 3.12).

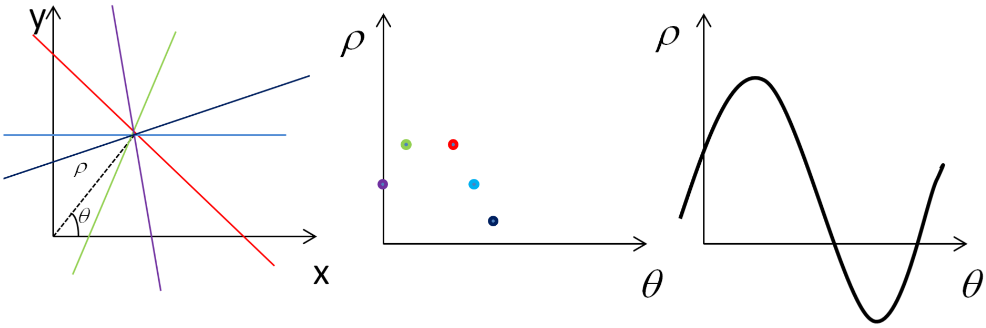


Figure 3.12: Bunch of lines intersecting at one point in the image space corresponds to the points which form a sinusoid in a Hough space. The intersecting point in the image space corresponds to the sinusoidal curve in the Hough space.

A collection of the points, which create a line in the image space, will construct different sinusoids in the Hough space. The crossing of the sinusoids in one point characterizes that points in the image space are located on the same line. Figure 3.13 reflects the foregoing explanation. Three sinusoids in the Hough space intersect in two common points (see Figure 3.13(b), consequently, three image points create the line in the image space like in Figure 3.13 (a).

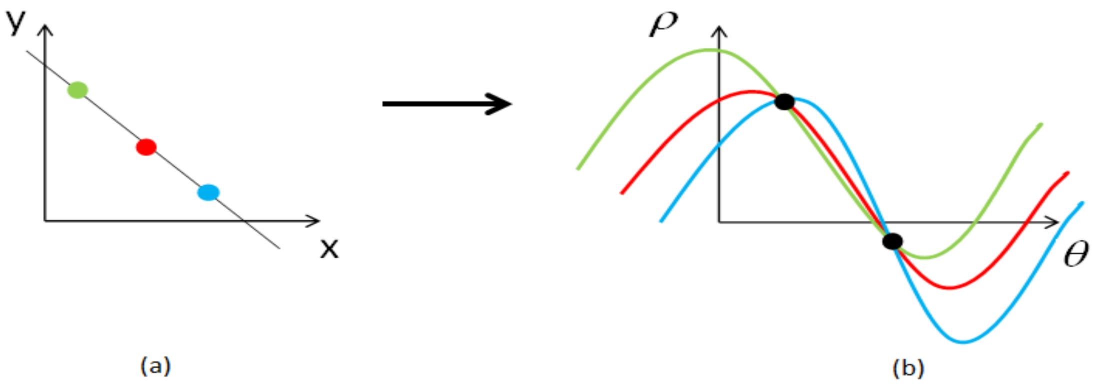


Figure 3.13: (a): Points in the image create a straight line. (b): Sinusoids in the Hough space intersect in two points.

In other words, in order to find the straight line in the image space, one needs to look for the intersection of the sinusoids in Hough space. In order to determine the areas where the most Hough space lines intersect, the two-dimensional accumulator is used. For every pixel (p_x, p_y), the Hough Transform algorithm determines if the straight line at that pixel exists or not. If there is enough evidence of a straight line at that pixel, the corresponding parameters (ρ_0, θ_0) will be calculated and the accumulator's bin where these parameters fall into will be incremented. The most likely line can be extracted by finding the bins with the highest values.

Progressive Probabilistic Hough Line Transform

The classical Hough Transform detects infinite lines given by the parameters θ and ρ. In order to find the finite lines in the image, the variant of the common Hough Transform can be applied, which is called Progressive Probabilistic Hough Line Transform [158]. The Probabilistic Hough Transform is the optimized version of the Hough Transform. The idea of this algorithm is transforming only a subset of the pixels in the image space into the accumulator instead of all the considered points. When a bin in the accumulator corresponding to the particular infinite line contains certain amount of votes, in the image space the searching process starts if one or more finite lines are present. In order to detect all finite lines on the image, one needs to carefully determine the minimum length of lines to be detected [136].

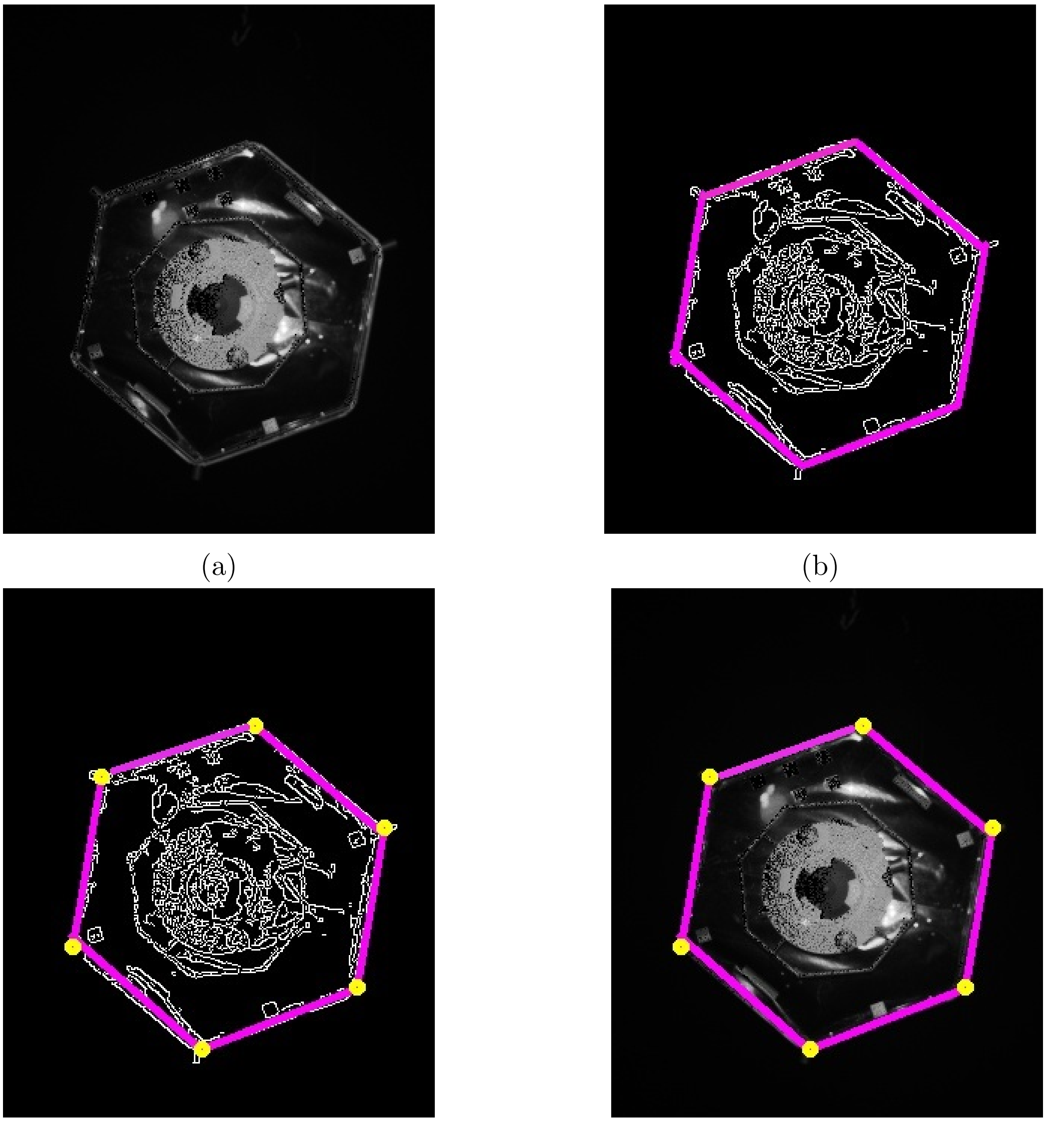
Image Processing

In this thesis, the straight line detection technique is realized with the open source library OpenCV [136]. The Standard Hough Line Transform is included in the $HoughLines()$ function and the Progressive Probabilistic Hough Line Transform is implemented inside the $HoughLinesP()$ command. Below, I provide a step-by-step description of the applied image processing procedures with corresponding images taken from the front view of the mockup (in Figure 3.14) and from the side part (in Figure 3.15). In summary, the line detection procedure consists of the following steps [159]:

2. Low-pass filtering is executed in order to reduce image noise.

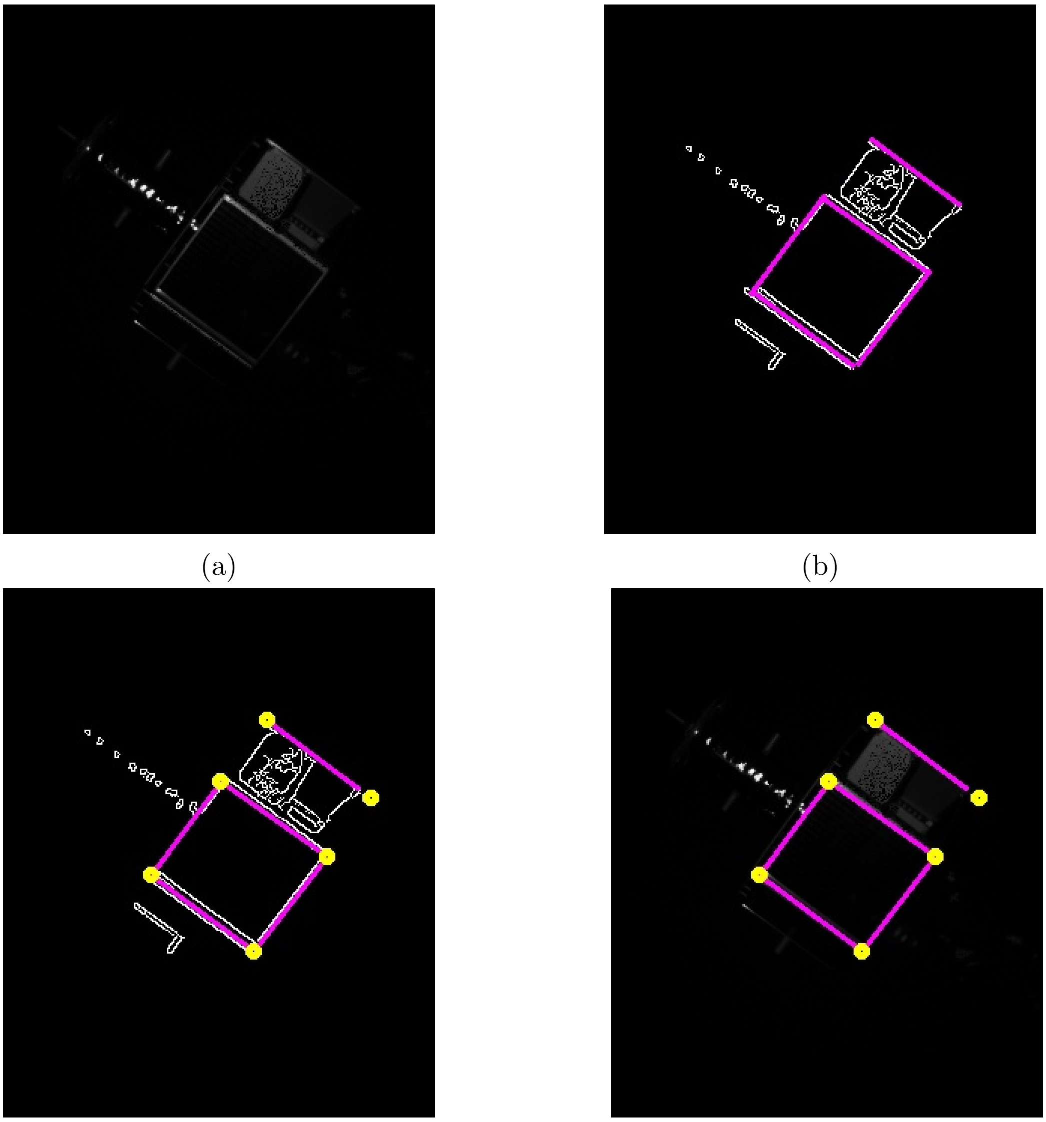
3. The Canny-edge operator is used in order to extract edges of the amplitude image, see in Figures 3.14 - 3.15(b) .

4. The Probabilistic Hough Line Transform is employed for the detection of finite lines, see Figures 3.14 - 3.15 (c).



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

Figure 3.14: Front view images. (a): 2D gray-scaled image from the amplitude channel. (b): Image with detected Canny edges and Hough Lines. (c): Image with detected Canny edges, Hough Lines and end points of the straight lines. (d): 2D gray-scaled image with detected Hough Lines and end points of the straight lines.



(a) (b)

(c) (d)

Figure 3.15: Side view images. (a): 2D gray-scaled image from the amplitude channel. (b): Image with detected Canny edges and Hough Lines. (c): Image with detected Canny edges, Hough Lines and end points of the straight lines. (d): 2D gray-scaled image with detected Hough Lines and end points of the straight lines.

The Hough Line Transform performs really well with small resolution amplitude images of the PMD sensor (see Figures 3.14 - 3.15). By finding straight lines on the 2D image, I need to generate a list of corrected feature pairs between these detected lines and given lines of the 3D model. In order to do that, primary, is is necessary to project the known 3D wireframe model onto the image plane using the pose solution obtained from the precedent pose estimation. The projected wireframe model in the image plane represents the set of the linked lines. However, in some cases due to the space environmental illumination or features of the mockup's material the detected edges from the image can be partial and broken lines, merged lines or even completely wrong detected lines, which do not belong to the model. These inaccuracies can increase the number of false feature correspondences between lines in the image and the model [89]. Furthermore, it definitely induces an incorrect motion estimation. To cope with this problem, a nearest neighbor search among lines is involved in the chain of image processing pipeline. The line neighbor search is based on the differences in (ρ and θ) representation of lines. The subset of lines that meets the threshold requirements is assumed to be correct.

Presuming the straight line as a vector between a start and an end point, one can represent detected lines in the image as one set of corresponding points. The same procedure is implemented for the lines of the projected model. And the current step is to sum up the point-to-point correspondences of all the start and end points from both the model and the image.

Pose from Points

After the image processing phase a set of feature correspondences between the image and the wireframe model of the target was generated. The following step is to calculate the pose of the spacecraft with respect to obtained feature correspondences. I assume that during image processing with the Hough Line Technique a set of image points $\rho_{img} = [u_{img}, v_{img}]$ and a set of corresponding model points $p^T = \left[p_x^T, p_y^T, p_z^T\right]$ expressed in the target frame T were obtained.

Let us consider the pose (R_T^C, t^C) as a 6 parameters vector $x = [t^C, \theta]$, where t^C is position vector of the target with respect to the camera frame and $\theta = [\theta_1, \theta_2, \theta_3]$ is a set of the Euler angles, which determines the orientation of the target spacecraft. The projection of the point p^T on the image is obtained through the 3D-2D true perspective projection equations:

$$p^C = R_T^C p^T + t^C \tag{3.9}$$

$$\rho_M = \begin{bmatrix} u_M \\ v_M \end{bmatrix} = \begin{bmatrix} \frac{p_x{}^C}{p_z{}^C}\alpha + u_0 \\ \frac{p_y{}^C}{p_z{}^C}\beta + v_0 \end{bmatrix} \tag{3.10}$$

In the Equations (3.9) and (3.10), p^T is the feature point of the target model defined in the target frame, p^C is the same point in a camera frame after applying transformation (R_T^C, t^C) , (u_M, v_M) is the pixel of the image corresponding to the feature, (α, β) focal lengths of the camera and (u_0, v_0) the principal point of the image. Equation (3.10) describes the simple camera model, which uses only focal lengths and the principal point of the camera. Skew factor, radial and tangential lens distortions are not included here. The camera model which is presented in Section 2.4 is more complex than this one. For each coupled feature correspondence image-model it is possible to define the following residual error:

$$e = \rho_M - \rho_{img} = \begin{bmatrix} u_M(x) - u_{img} \\ v_M(x) - v_{img} \end{bmatrix} \tag{3.11}$$

where ρ_M is the projection of the geometric feature of the target model, whereas ρ_{img} is the end point detected with the Hough Line Transform. The error in the Equation (3.11) has 6 unknown parameters, which are described by state vector x. The state vector contains three Euler angles, which define the rotation matrix and the three coordinates of the translation vector. Each feature correspondence is defined by two conditions, therefore, at least three pairs of matches between detected endpoints and projected features are required to solve the system equation for the defined error function. Let us assume that one has N feature correspondences between image and model points. The Gauss-Newton approach iteratively minimizes the sum of squared errors in order to find the position and orientation defined by x.

$$S(x) = \sum_{i=1}^{N} \|e_i(x)\|^2 = \sum_{i=1}^{N} [(u_{M_i}(x) - u_{img_i})^2 + (v_{M_i}(x) - v_{img_i})^2] \tag{3.12}$$

Given the first guess x_0, the pose that minimizes Equation (3.12) is iteratively obtained as

$$x_{k+1} = x_k - (J_k^T J_k)^{-1} J_k^T E_k \tag{3.13}$$

where

$$E_k = \begin{bmatrix} e_1(x_k) \\ e_2(x_k) \\ .. \\ e_N(x_k) \end{bmatrix} \tag{3.14}$$

is the error vector with e_i defined in the Equation (3.11) and J_k is the Jacobian of e calculated at x_k and defined as

$$J = \frac{\partial E_k}{\partial x} \tag{3.15}$$

The Equation for the Jacobian (3.15) for point correspondences can be written as follows:

$$J = \left[\frac{\partial E_k}{\partial t^C}, \frac{\partial E_k}{\partial \theta} \right] = \begin{bmatrix} \frac{\partial e_1}{\partial t^C} & \frac{\partial e_1}{\partial \theta} \\ .. & .. \\ \frac{\partial e_N}{\partial t^C} & \frac{\partial e_N}{\partial \theta} \end{bmatrix} \tag{3.16}$$

The size of the Jacobian is 2Nx6 since each residual error in the Equation (3.11) is defined by two components along u and v coordinates of the image.

The general expression of the rows of the Jacobian being

$$J_i = \left[\frac{\partial e_i}{\partial t^C}, \frac{\partial e_i}{\partial \theta} \right]. \tag{3.17}$$

In the Equation (3.17) $e_i = \rho_{M_i} - \rho_{img_i}$, $i = 1 \dots N$.

The first element of the row can be rewritten as

$$\frac{\partial e_i}{\partial t^C} = \frac{\partial \rho_{M_i}}{\partial p_i^C} \frac{\partial p_i^C}{\partial t^C} \tag{3.18}$$

where

$$\frac{\partial p_i^C}{\partial t^C} = I_{3\times3} \tag{3.19}$$

and the follow equation obtained from Equations (3.9) and (3.10)

$$\frac{\partial \rho_{M_i}}{\partial p_i^C} = \begin{bmatrix} \frac{\alpha}{(p_z^C)_i} & 0 & -\frac{(p_x^C)_i}{(p_z^{C2})_i} \alpha \\ 0 & \frac{\beta}{(p_z^C)_i} & -\frac{(p_y^C)_i}{(p_z^{C2})_i} \beta \end{bmatrix} \tag{3.20}$$

Alternatively, the second element of the expression 3.17 can be presented as

$$\frac{\partial e_i}{\partial \theta} = \frac{\partial \rho_{M_i}}{\partial p_i^C} \frac{\partial p_i^C}{\partial \theta} \tag{3.21}$$

with

$$\frac{\partial p_i^C}{\partial \theta} = \left[\frac{\partial p_i^C}{\partial \theta_1}, \frac{\partial p_i^C}{\partial \theta_2}, \frac{\partial p_i^C}{\partial \theta_3} \right] \tag{3.22}$$

and

$$\frac{\partial p_i^C}{\partial \theta_j} = \frac{\partial R_T^C}{\partial \theta_j} p_i^T \qquad j = 1, 2, 3 \tag{3.23}$$

In the Equation (3.23), the rotation matrix is defined in terms of Euler angles $[\theta_1, \theta_2, \theta_3]$ as

$$R_t^C = \begin{bmatrix} c\theta_1 c\theta_1 & s\theta_1 s\theta_1 & -s\theta_2 \\ c\theta_1 s\theta_2 s\theta_3 - s\theta_1 c\theta_3 & s\theta_1 s\theta_2 s\theta_3 + c\theta_1 c\theta_3 & c\theta_2 s\theta_3 \\ c\theta_1 s\theta_2 s\theta_3 + s\theta_1 s\theta_3 & s\theta_1 s\theta_2 s\theta_3 - c\theta_1 s\theta_3 & c\theta_2 c\theta_3 \end{bmatrix} \tag{3.24}$$

where $c\theta = \cos\theta$ and $s\theta = \sin\theta$.

3.4 Fusion of Measurements

At the beginning of Section 3.3 the sketch (Figure 3.7) of the proposed fused architecture of a visual navigation system was depicted. In the previous Sections 3.3.1 and 3.3.2, I have described two completely different pose estimation techniques for the amplitude and depth images, where each of them gives an estimate (pose) of the target. Let us consider the fusion node block from the Figure 3.7. One of the simplest ways for the combination of measurements is to take a weighted average [160] of the pose vectors which is obtained after two different pose estimation techniques. The simple arithmetic mean of all measurements does not perform good enough, since one measurement can be more reliable than other [161]. Taking in account this fact, it is better to assign more importance and greater weight to an observation x_i from one output channel that is more reliable, whereas a less accurate observation from the other output channel will receive minor weights. The weighted average for the fused estimate of n different measurements x_i with non-negative weights ω_i looks as

$$x_{fused} = \frac{\sum_{i=1}^{n} \omega_i x_i}{\sum_{i=1}^{n} \omega_i}. \tag{3.25}$$

An Equation (3.25) can be simplified when the weights are normalized and sum up to 1:

$$x_{fused} = \sum_{i=1}^{n} \omega_i^{'} x_i, \sum_{i=1}^{n} \omega_i^{'} = 1. \tag{3.26}$$

From the mathematical point of view the weights ω_i for every single member of the pose vector can be related to an estimated variance of the measurement error σ_i^2 that occurred during pose estimation with one of the suggested methods

$$\omega_i = \frac{1}{\sigma_i^2}. \tag{3.27}$$

In the work of Elmenreich [162], the author shows that this variance of the fusion result x_{fused} is minimized and always smaller than the input variances

$$\sigma_{fused}^2 = \sum_{i=1}^{n} \omega_i^2 \sigma_i^2 = \sum_{i=1}^{n} \frac{1}{\sigma_i^2}. \tag{3.28}$$

3.5 Guidance, Navigation and Control System

Conducting research in the field of OOS at DLR, one has a great opportunity to implement hardware-in-the-loop (HIL) simulations of the rendezvous scenarios. It is an invaluable step for the on-ground simulations before the real mission takes place. In the works of Benninghoff et al. [163] and Rems et al. [164] the authors give a detailed overview of on-orbit servicing end-to-end simulation project. During this project on-board and on-ground systems are developed and verified. The colleagues from OOS group of DLR, who work on this project, are developing an advanced rendezvous GNC system, which includes sensors, pose estimation algorithms, navigation filter, guidance and control functions [164]. It should be noticed that this doctoral thesis is only a part of the current GNC system. Let us turn to Figure 3.16, where a hardware-in-the-loop simulation system for the rendezvous scenarios is represented.

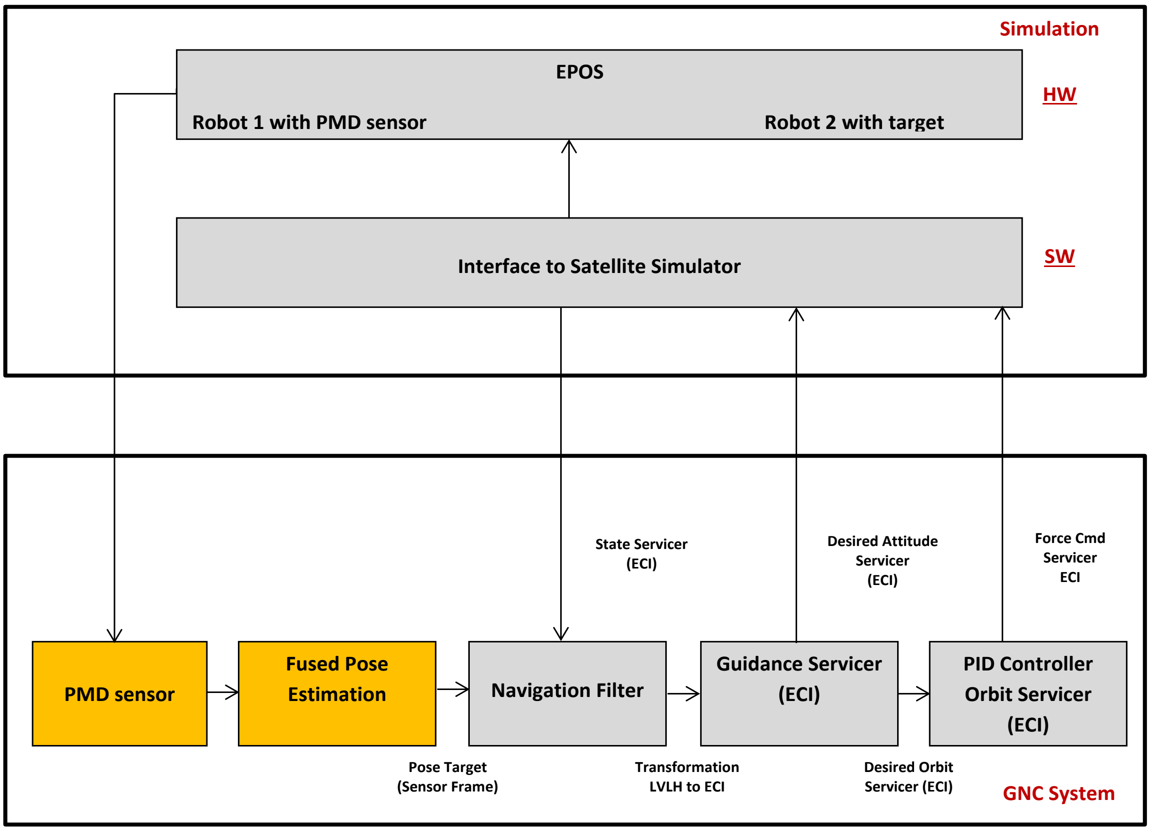


Figure 3.16: Overview of GNC system.

I have marked with the yellow colors two blocks of the GNC system: the PMD rendezvous sensor presented in Chapter 2 and pose estimation techniques described in Chapter 3. It gives an idea, where the place of the current research inside of the complex system is. The other blocks on Figure 3.16 are shadowed in gray color, since they are out of scope in this thesis. Using other visual sensors, the yellow block with PMD sensor can be replaced by any other optical sensor or by a group of sensors. In this case, the second yellow block of fused pose estimation could be replaced by another appropriate algorithms as well. The goal of HIL rendezvous tests is to ensure the stability of the GNC system with real-time measurements of state vector with PMD sensor during the whole approach phase.

The hardware-in-the-loop simulations are performed at EPOS, see Section 2.1, and include a software-based satellite simulator in order to compute the dynamics of the servicer and the target. The data from the satellite simulator consists of the state vector (position, attitude, velocity and angular velocity in Earth central inertial (ECI) frame). The PMD sensor with correspondent pose estimation algorithms provides the position and attitude of the target relative to the servicer frame. The estimated target pose is an input to the modifided extended Kalman filter [164], which output is a noise-minimized guess in ECI frame. According to Benninghoff et al. [163] the filter provides the transformation from the local vertical - local horizontal system (LVLH) to ECI system. Having this information, the GNC system calculates control forces and desired servicer attitude, such that the servicer can follow a trajectory generated by a guidance function. The developed rendezvous GNC system is integrated into the EPOS hardware-in-the-loop simulator. The performance results of the visual navigation in closed loop with the PMD sensor are presented in the next Chapter 4.

Chapter 4

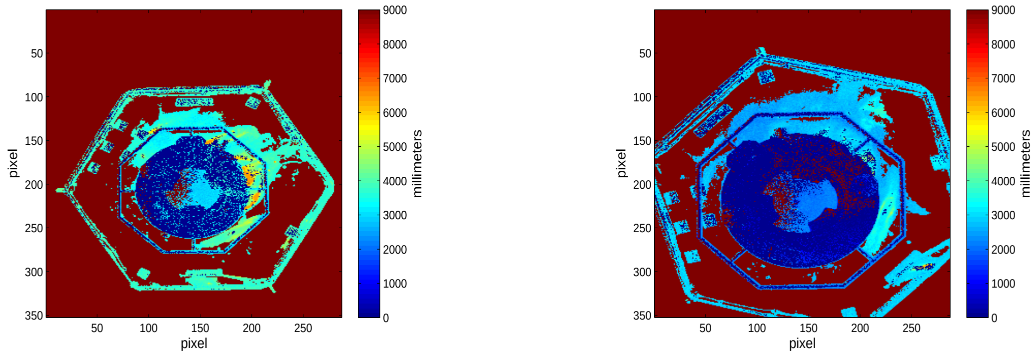
Tests, Analysis and Discussion

This chapter provides analysis of extended tests of the algorithms presented in the previous chapter. The practical use of the pose estimation techniques with the real data from the DLR-Argos3D - P320 sensor allows to characterize the competence of the developed algorithms with the sensor's data; find appropriate conditions and parameters for the better performance; analyze limits of sensor usage and computational time complexity. Finally, I provide the results of the closed-loop pose estimation in a close range. All experiments conducted with the EPOS simulator, which allows high-accuracy rendezvous simulations.

4.1 Analysis of Limitations using DLR-Argos3D - P320

4.1.1 Position Limits

In this section I am going to define some limits according to the operational range of the given DLR-Argos3D - P320 sensor. Especially, I am interested in the minimum and maximum distances where the sensor can work properly. According to the main approach scenario, namely front side rendezvous, it is very important to have the whole form of the mockup in the field-of-view (FOV) of the camera. It means that the front hexagon must be observable in order to let the algorithm detect the contours and features. Figure 4.1 shows two images of the target, one, where the target is completely in the FOV (Figure 4.1 (a)), and one, where some parts of the mockup are out of the FOV (Figure 4.1 (b)).



(a)

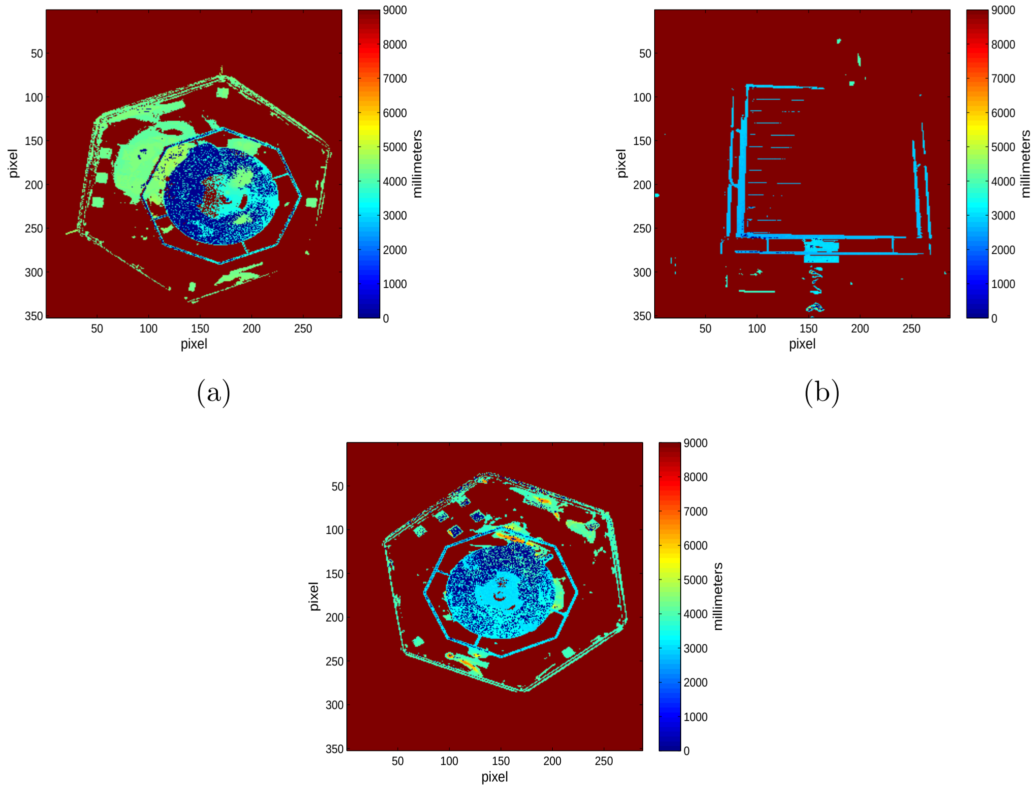
(b)

Figure 4.1: (a): The entire mockup is in the FOV. (b) Some parts of the mockup are not in the FOV of the camera.

Since the FOV of the existing camera is less than 30 degrees (28.91 x 23.45), one is able to detect the whole hexagon contour with the diameter of 1.66 meters within a distance starting from approximately $d_{min} > 5$ meters. The maximum operating range of the given sensor is defined by the following conditions: modulation frequency of the emitted signal, resolution of the sensor chip and power of the illumination unit of the camera. Depending on the modulation frequencies presented in Table 2.1, the maximum measurement range can be defined with following equation: $d_{max} = \frac{c}{2f_{mod}}$. It means that the maximum calculated distance d_{max} of the DLR-Argos3D - P320 can be varied from 29.9 meters to 5 meters. Nevertheless, the current modulation unit of the camera is suitable for the close range simulations (< 10 meters) and not for the long one. Moreover, the resolution of the current PMD sensor is small in comparison with existing CCD sensors, the features of the imaged object become not to be observable and it leads to big errors in pose estimation. Taking into account all these constrains, I observed that the maximum operational distance of the current sensor is in practice d_{max}= 8 meters. For the rendezvous scenarios d_{min} is a final point and d_{max} is a starting point.

4.1.2 Surface Material Limits

In Section 2.3.2 I discussed the features and characteristics of the PMD sensor in relation to different surfaces. The strong weakness of the DLR-Argos3D - P320 sensor were brought out when working with high absorbing (Figure 4.2 (a)) and high reflective materials (Figure 4.2 (b)) - loss of the depth information.



(a)

(b)

(c)

Figure 4.2: (a): The high reflective MLI material of the front side doesn't provide depth measurements. (b) The high absorbing material of the solar panels doesn't provide depth measurements. (c) A good image for the pose initialization.

Consequently, the loss of depth information leads to the impossibility to use the pose initialization technique proposed in Section 3.2, since it uses a depth image as a primary information. Therefore, for my working test case with the given mockup and current PMD sensor I define limitation concerning the appearance of the target on the image. For the pose initialization technique the image of the target mockup must be taken from the front side, where the hexagon and long "nose" part are well represented (Figure 4.2 (c)), because they are made from solid materials which don't absorb the emitted light from the sensor.

With the tracking phase the situation looks a bit different. The proposed tracking pipeline is developed on the basis to avoid full dependence of the visual navigation on the depth image. And in case the distance information is completely or partly lost, one can rely on the amplitude image where the surface materials of the current mockup do not disturb it. In this thesis I consider this identified feature of the PMD sensor as incredibly important, since it can lead to the possibility to get rid of some limitation with the surface materials.

4.2 Offline Test: Pose Initialization

In Section 3.2 I presented the pose initialization technique using the depth image. To use this technique it is necessary to have on board of the chaser the 3D model of the space object. In the paper of Klionovska et al. [165] the pose acquisition pipeline with the current sensor was tested. It was revealed that there is a great need to use proper features of the used 3D model (see Section 2.2), such as "nose" and front hexagon, in order to get an accurate position and attitude. Let us define some necessary parameters and run the pose initialization technique. The errors are processed in the camera coordinate frame, where the Z-axis is taken along the optical axis of the camera. See Figure 1.12, in red color. The ground truth for every logged image with DLR-Argos3D - P320 sensor is provided by EPOS.

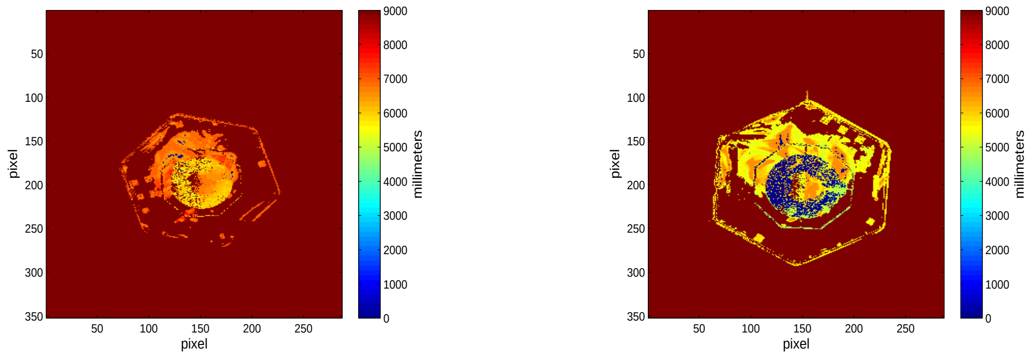
Both point clouds, the model and the scene, must be down-sampled, since there is no need to process massive point clouds. Moreover, it can lead to wrong estimation of the pose if the density of point clouds is very different. The algorithm for the down-sampling proposed in the work of Bridson [166], called Fast Poisson Disc Sampling, is used in this thesis for point cloud size reduction. Briefly, this algorithm needs a sampling step in order to down-sample a point cloud. Two sampling steps were defined: for the model *sampling_step_model* and for the scene *sampling_step_scene*. The sampling step of the model point cloud *sampling_step_model* = 0.01 is a constant, meaning the data of the model was reduced once and continue using the obtained minimized model point cloud during all simulations. The parameter *sampling_step_scene* is not usually the same. It changes depending on the distance between camera and the mockup. The closer the camera, the bigger the point cloud and there is a need to increase *sampling_step_scene*.

Let us provide some results after evaluation of the initial pose only using depth images when the target is placed on the distance $d_1 = 7.7$ meters (Figure 4.3 (a)) and $d_2 = 6.5$ meters (Figure 4.3 (b)) from the camera. The *sampling_step_scene* is also different for both distances and presented in Table 4.1.

In offline tests, the best parameter *sampling_step_scene* for every range is chosen by iterating over the parameters. For the future real missions, this parameter must be also chosen on the ground during the preparation tests. It will be important to analyze the sizes (density and also the diameter) of the scene point cloud in different ranges by the selected resolution of the PMD sensor. The selected values can be stored on board for further use.

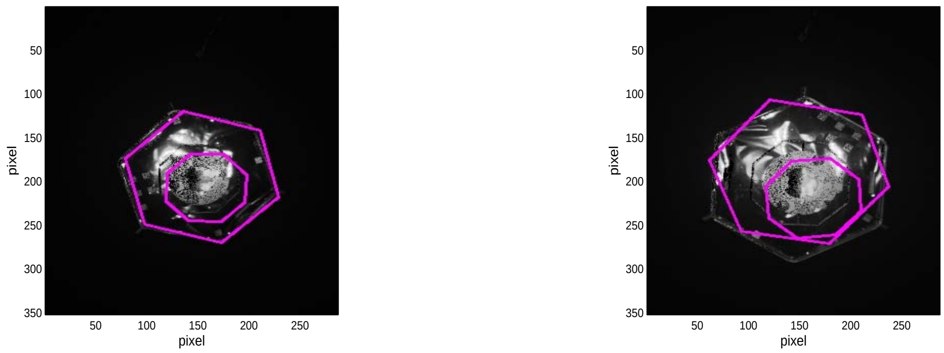
Table 4.1: Errors for estimated initial pose.

errors	$d_1 = 7.7$ meters *sampling_step_scene* = 0.03	$d_2 = 6.5$ meters *sampling_step_scene* = 0.035
roll, deg	3.763	23.614
pitch,deg	0.914	7.770
yaw,deg	5.251	11.040
z,m	0.141	0.368
y,m	0.030	0.072
x,m	0.058	0.006


(a)

(b)

Figure 4.3: (a): Depth image from the PMD sensor at the distance $d_1 = 7.7$ meters. (b): Depth image from the PMD sensor at the distance $d_2 = 6.5$ meters.


(a)

(b)

Figure 4.4: (a): Re-projection of the model with estimated initial pose onto the image, $d_1 = 7.7$. (b): Re-projection of the model with estimated initial pose onto the image, $d_2 = 6.5$.

The co-registered gray-scaled amplitude image is used for the outcome representation. The magenta re-projection of the 3D model shape onto the 2D image is the result of the estimated initial pose. From the results presented in Figure 4.4 and Table 4.1 it is not hard to see that a determination of the attitude of the target using only the point cloud from the depth image leads to the big errors. The acquired pose of the target spacecraft at the distance $d_1 = 7.7$ meters shows better results than at the closer distance. It is due to the relation between the distance and the number of scene points - the closer distance, the bigger point cloud. The scene point cloud contains more outliers in the depth images, which can deteriorate the process of pose estimation since they create wrong pairs of points, which are not presented in the model point cloud. In the depth image of Figure 4.5 one can observe a massive part of wrong distance measurements, circled with a black contour. In my opinion, this is due to the surface reflections. From the first sight, one can think that these measurements are reliable and belong to the proper front hexagon part, but in reality it's not. Since I cannot avoid it at the moment, I look for an improvement of the pose technique.

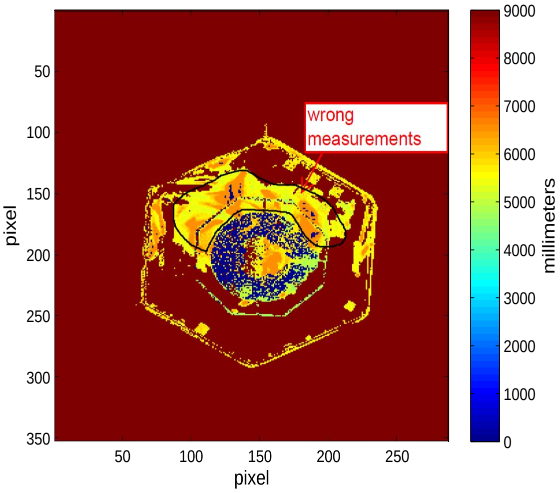


Figure 4.5: Depth image with outliers.

4.2.1 Improvement of the Initial Pose with Amplitude Image

During the test phase of the algorithm for the pose initialization, many test cases have been made with various parameters of the algorithm. A test case with the correspondent parameters above is the best setup which was achieved. Since the obtained error values of the position and orientation are not accurate enough, I propose to use an additional pose refinement. The idea to use an amplitude image for the correction of the estimated initial pose came across after a deep investigation of the appropriate tracking method, and especially, after a solid decision to use the amplitude image along with the depth one for the hybrid navigation technique. In the papers [167, 168], I showed the advantages of using the amplitude image in conjunction with the depth one for pose estimation in every frame.

On this basis, I came up with an idea to use supplement processing of the co-registered amplitude image as follow-up refinement of the calculated initial pose. The algorithm for the 2D image from Section 3.3.2 is involved here as the follow refinement. The only prerequisite for this technique is to have an initial guess. In this case the initial guess is the previously obtained state vector with the depth image.

So, let us assume the results of the initial state vector from the Table 4.1 as the guess input for the pose estimation technique. The amplitude images used for the image processing are the same gray-scaled images as in Figure 4.4. I run the image processing and the pose estimation procedures.

Table 4.2: Errors for corrected initial pose.

errors	$d_1 = 7.7$	$d_2 = 6.5$
$roll$, deg	1.681	2.391
$pitch$,deg	0.368	0.179
yaw,deg	0.586	1.621
z,m	0.109	0.250
y,m	0.002	0.010
x,m	0.030	0.029

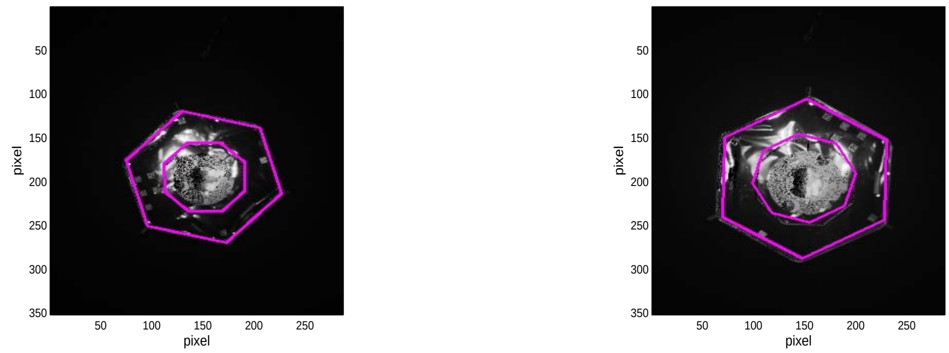

(a) (b)

Figure 4.6: (a): Re-projection of the model with an estimated corrected initial pose onto the image, $d_1 = 7.7$ meters. (b): Re-projection of the model with a estimated corrected initial pose onto the image, $d_2 = 6.5$ meters.

In Table 4.2, I provide the errors of the position and attitude after an implementation of the correction technique. It can be explicitly noticed, how the results of both poses have been greatly improved. First of all, the attitude of the target. The numerical assessment of the improved roll angle for the bad second case at the distance $d_2 = 6.5$ meters shows the decrease of the error up to 10 times, from 23.614 *deg* to 2.391 *deg*. Moreover, an evaluated distance to the target along Z axis without correction had a difference with the real value in more than 30 *cm*. After the improved procedure the position along Z axis was improved up to 25 *cm*. When the object locates farther away at $d_1 = 7.7$ meters, one can also notice some improvements, even if they are not so big. This is because the errors after the first pose estimation with the depth image were also much lower.

In order to get the statistical overview of the improvement technique, I do tests with a data set of 200 images in a range from 8 to 5 meters. In Figure 4.7 I plot some images with the results of the pose initialization technique - some pairs of initial pose with and without correction. The mean values for the estimated parameters of the state vector for both cases are presented in Table 4.3.

One can definitely notice the improvements of the estimated position and rotation components. The rotation around the principal axis of mockup is difficult to estimate only with a point cloud. The mean error in a roll angle reaches up to 10.92 *deg*. After the improvement technique with the amplitude image, the mean error of the roll angle has been decreased up to 1.22 *deg*. The mean errors for other rotation components

Table 4.3: Mean errors for the initial pose with and without correction.

errors	without correction	with correction
μ_{roll}, deg	10.929	1.221
μ_{pitch},deg	4.517	1.236
μ_{yaw},deg	4.113	1.555
μ_z,m	0.117	0.068
μ_y,m	0.022	0.022
μ_x,m	0.030	0.011

are 3 times less with the proposed pose correction. Having a look at the translation components, especially in z direction μ_z, one can see that the estimation of the distance can be improved with the follow-up 2D improvement technique. The mean error for the initial pose calculated only with the depth image $\mu_z = 11.7$ cm, whereas for the corrected initial pose $\mu_z = 6.8$ cm.

4.3 Offline Test: Weights Definition

Before I start the closed loop simulations using the proposed hybrid navigation technique, it is necessary to find correct the weights defined in Section 3.4 for a fusion of both estimates from the depth and amplitude images. I present here an offline test, where I provide the frontal approach scenario from 8 to 5 meters. In this data set, see Figure 4.8, the target is rotating around its principal axis of inertia at a rate 1 deg/s. The modulation frequency of the camera was set to 5 MHz. The camera coordinate frame (see Figure 1.12 in red color) is used in order to evaluate the results, where Z-axis is the optical axis of the camera.

4.3.1 Test Scenario 1 and Test Scenario 2: Pose Estimation using Depth and Amplitude Images Separately

I run the algorithms for the depth and amplitude images separately using the provided dataset. Let us define: a test scenario 1 presents the frame-to-frame pose estimation technique using depth images and ICP with the reverse calibration technique (see Section 3.3.1); a test scenario 2 concludes the results of the pose estimation algorithm with image processing of amplitude images (see Section 3.3.2). I plot the errors for the translation and rotation components from the test scenario 1 in Figure 4.9 and Figure 4.10.

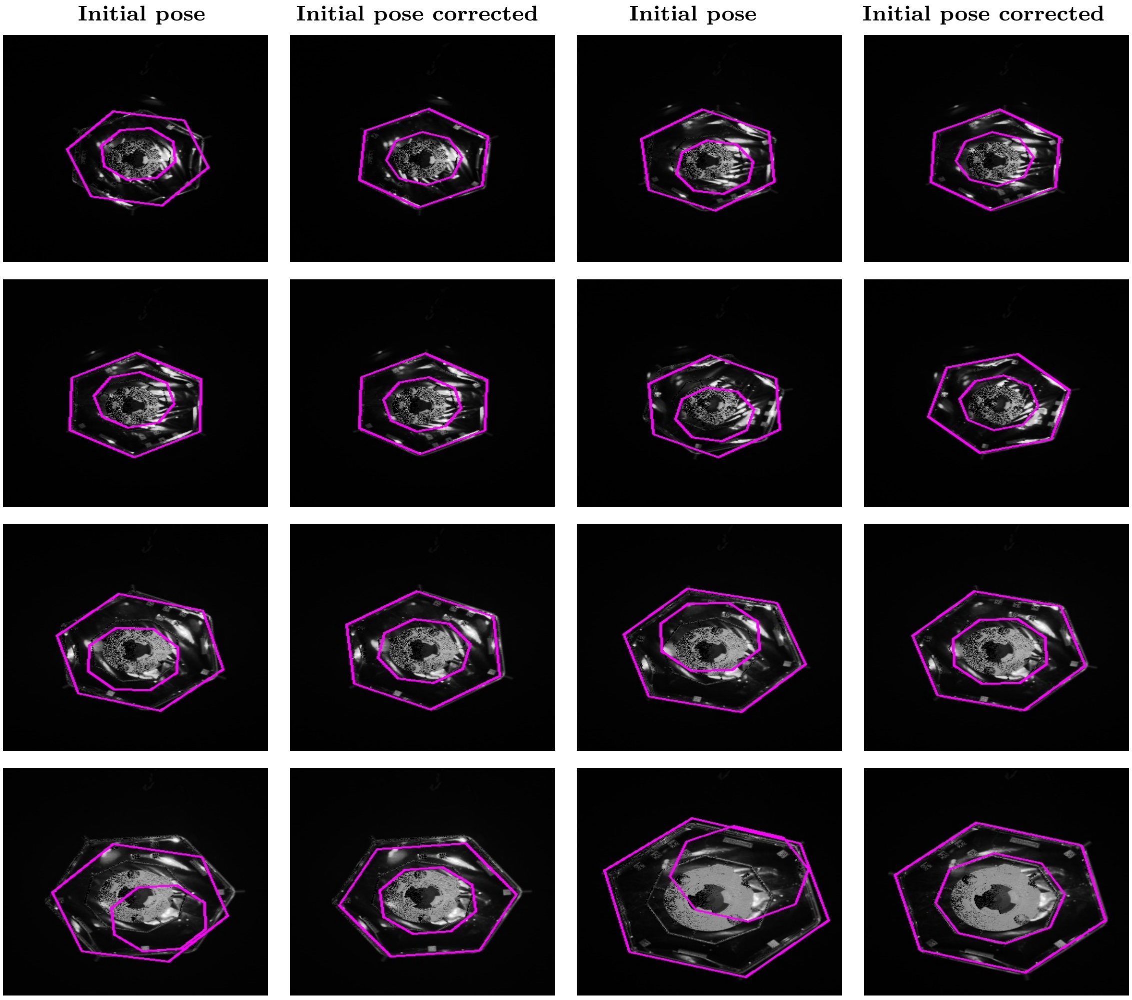


Figure 4.7: Results of pose initialization technique without and with correction.

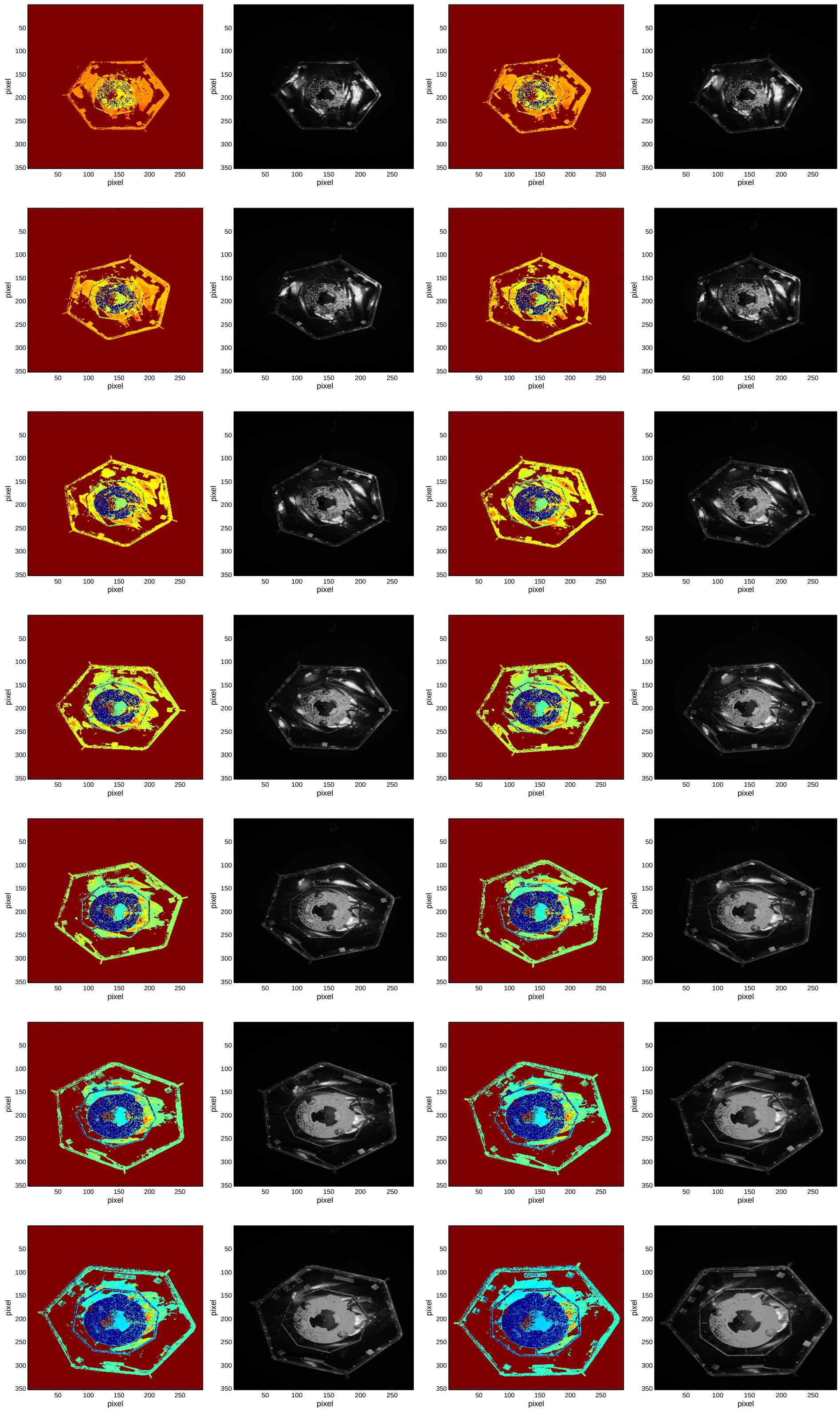


Figure 4.8: Some depth and amplitude images taken at the distance from 8 to 5 meters from the data set.

Figure 4.9: Translation errors during close approach for the test scenario 1 (depth image).

Figure 4.10: Rotation errors during close approach for the test scenario 1 (depth image).

In Figure 4.11 and Figure 4.12 I plot the translation and rotation errors for the test scenario 2 (amplitude image).

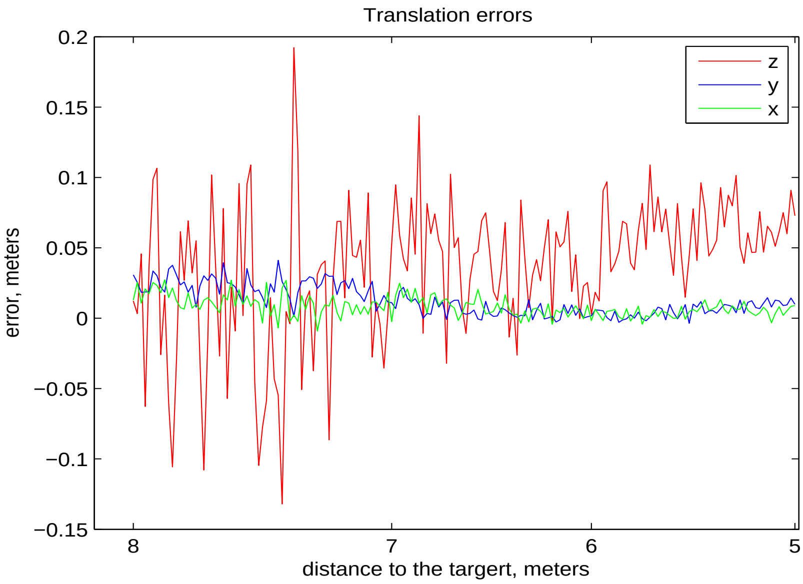


Figure 4.11: Translation errors during the close approach for the test scenario 2 (amplitude image).

Figure 4.12: Rotation errors during the close approach for the test scenario 2 (amplitude image).

In Table 4.4 the mean errors for the attitude and position vectors are illustrated. One can notice that in the test scenario 1 the mean errors for roll, pitch and yaw angles are 0.719, 2.830 and 4.913 degrees, and for the position errors along z, y and x axis they are 0.0277, 0.0154, 0.0108 meters. We can see strong deviations in the pitch and yaw angles, when the pose is estimated only with the depth data. The bad estimated orientation of the target in the previous frame, will lead to the wrong choice of the neighbors in the following frame for the model and scene point clouds. This fact, in its turn, will cause an influence on the correct estimation of the position vector as well.

For the test scenario 2, I also present the mean errors in Table 4.4. For the translation components z, y, x - 0.0526, 0.0126 and 0.0088 meters; for the rotation part (roll, pitch, yaw) - 0.883, 1.300 and 0.919 degrees. The results for the attitude from the test scenario 2 are definitely better than from the test scenario 1. Nevertheless, when considering the mean value μ_z for the distance component estimated with the amplitude images, one can clearly observe a worse result than with the depth images.

Table 4.4: Mean errors for the test scenario 1 and the test scenario 2.

μ	test scenario 1 (depth image)	test scenario 2 (amplitude image)
μ_{roll}, deg	0.719	0.883
μ_{pitch},deg	2.830	1.300
μ_{yaw},deg	4.913	0.919
μ_z,m	0.0277	0.0526
μ_y,m	0.0154	0.0126
μ_x,m	0.0108	0.0088

Table 4.5 presents the result of the standard deviations for three rotation angles and for z, y, x components of the position vector.

Table 4.5: Standard deviations for the test scenario 1 and the test scenario 2.

σ	test scenario 1 (depth image)	test scenario 2 (amplitude image)
σ_{roll},deg	0.569	0.434
σ_{pitch},deg	2.342	0.143
σ_{yaw},deg	2.530	0.105
σ_z,m	0.0232	0.0319
σ_y,m	0.0078	0.0065
σ_x,m	0.0132	0.0102

4.3.2 Test Scenario 3: Fusion of Pose Vectors with Weights

Taking into account revealed tendency after the test scenarios 1 and 2, I propose the following concept for the data fusion. In the test scenario 3, I am going to estimate pose during the close approach for the same dataset and use the weighted average technique for the translation components. The rotation is completely taken from the pose vector estimated with the amplitude image. In other words the weight ω_{pitch}=ω_{yaw}=ω_{roll}=1 for the amplitude estimate and ω_{pitch}=ω_{yaw}=ω_{roll}=0 for the depth estimate. This is so, because the algorithm with the amplitude images is less sensitive for the estimation of the orientation and usually provides better results. Actually, the comparison of test scenario 1 and 2 proved it. In the test scenario 3 I apply weights for the z, y and x coordinates presented in Table 4.6, which are calculated using the Equation (3.27) and variations from the Table 4.5.

Table 4.6: Weights for the translation components.

Weight	Depth estimate	Amplitude estimate
ω_z	0.65	0.35
ω_y	0.40	0.60
ω_x	0.37	0.63

In Figure 4.13 and Figure 4.14 I plot the results of the angular and position measurement errors after execution of the test scenario 3.

Figure 4.13: Translation errors during close approach for the test scenario 3.

Figure 4.14: Rotation errors during close approach for test scenario 3.

The numerical results in form of mean errors and standard deviations for the tracking with fused states are listed in Table 4.7.

Table 4.7: Mean errors and standard deviations for the test scenario 3.

mean error	values	standard deviation	values
μ_{roll}, deg	0.884	σ_{roll}, deg	0.453
μ_{pitch}, deg	1.001	σ_{pitch}, deg	0.545
μ_{yaw}, deg	0.919	σ_{yaw}, deg	0.306
μ_z, meters	0.0367	σ_z, meters	0.0261
μ_y, meters	0.0240	σ_y, meters	0.0075
μ_x, meters	0.0096	σ_x, meters	0.0121

In fact, as I expected, the fused technique with its measurements errors presented in Table 4.7 overcomes the drawbacks of pose estimation techniques. The mean errors for the estimated components of the orientation vector are almost the same as in the test 2, whereas the mean errors for the components of the position vector are more similar as in the test 1. Since I use the rotation part only from the amplitude estimate, one would expect that the attitude errors in Figure 4.14 should look like in Figure 4.12. However it looks slightly different, because it is a tracker and the result of the position estimate of the previous image influences also the attitude result of the current image.

For the current thesis the weights have been chosen with a post-facto procedure. For the real missions it should be avoided to have a hard-coded parameters. Therefore, the weights and other parameters of the algorithms and sensor are commandable and can be changed during the whole mission per telecommand. Other sophisticated way can be to let the algorithm calculate and set the weights already during the pose estimation phase. That approach would be interesting to investigate in the future.

4.4 Visual Navigation using PMD Sensor in a Closed Loop

In this section I am going to present the results of the visual navigation in a closed loop using PMD sensor. The GNC system, which is integrated in a hardware-in-the-loop simulator, is presented in section 3.5. In Figure 3.16, the DLR-Argos3D - P320 sensor and the hybrid navigation technique with weighted fusion are parts of the system. The close approach is in a range from 8 to 5.5 meters, the servicer approach velocity is 0.01 m/s. The rendezvous scenario has been repeated 10 times. During these tests I used the sun simulator as a spotlight. Let us define some errors constraints similarly as it was used in ATV rendezvous mission [169]: the rotation errors are not higher than 5 deg and translation errors should not increase 10 cm.

The nearly circular orbit (perigee 771.7 km, apogee 773.5 km), a semi-major axis of 7143 km, an inclination of 98.2° and an orbit period of about 100 min has been chosen for the current test scenario. This real orbit belongs to the current ENVISAT satellite, which can be potentially considered as a client.

In Table 4.8, the root mean square (RMS) errors for 10 approach trajectories are presented.

Table 4.8: Root mean square errors for position and orientation using the PMD sensor in a closed-loop for 10 approach trajectories.

RMS	1	2	3	4	5	6	7	8	9	10
$roll_{rms}$, deg	0.6490	0.6088	0.5579	0.5485	0.5495	0.5693	0.5436	0.5418	0.6095	0.5657
$pitch_{rms}$, deg	0.4633	0.4869	0.5030	0.5577	0.5147	0.5499	0.5661	0.5653	0.5043	0.4826
yaw_{rms}, deg	0.8508	0.8778	0.9026	0.9260	0.9360	0.9929	1.0665	1.1555	1.2009	1.2206
z_{rms}, meters	0.0311	0.0288	0.0305	0.0271	0.0287	0.0276	0.0249	0.0261	0.0347	0.0324
y_{rms}, meters	0.0185	0.0185	0.0198	0.0164	0.0180	0.0154	0.0149	0.0151	0.0205	0.0224
x_{rms}, meters	0.0068	0.0075	0.0073	0.0066	0.0070	0.0048	0.0052	0.0044	0.0053	0.0061

From the Table 4.8 one can see that the rotation errors in all 10 scenarios are quite small. The RMS errors for the attitude components don't increase 2 deg. The maximum RMS error for the measured distance is z_{rms}=0.0347 meters (test 9) and the minimum is z_{rms}= 0.0249 meters (test 7).

In Figure 4.15 and Figure 4.16 two plots with the rotation and position errors for the test number 8 are presented. In Figure 4.16, one can observe that the target was approached without any breaks in the visual navigation system during the rendezvous.

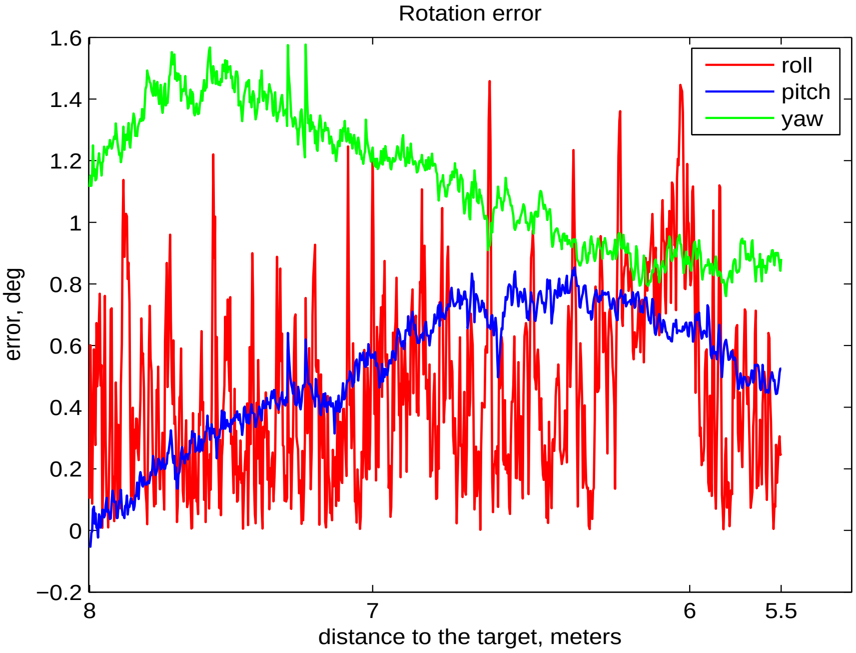


Figure 4.15: Rotation errors during rendezvous with PMD sensor in the closed loop for the test 8.

Figure 4.16: Position errors during rendezvous with PMD sensor in the closed loop for the test 8.

Let us have a look at the distance component separately. I plot three distances together in Figure 4.17: real distance in a red color, measured distance with the algorithm in a blue color and smoothed distance with the Kalman filter in a green color. One can notice that the rendezvous approach with the visual sensor is stable, without big jumps and deviations from the real trajectory. In Figure 4.18, I provide two images with a zoom into Figure 4.17. These two plots clearly show that at the far distances between servicer and target the deviations of the measured Z coordinate and the real trajectory are much bigger than at the close distances. This may be caused due to the small resolution of the PMD sensor, where sub-pixel changes cannot be accurately detected. In total, the visual navigation in the closed loop with the smoothing Kalman filter provides acceptable errors and ensures stable approach from 8 meters to 5.5 meters.

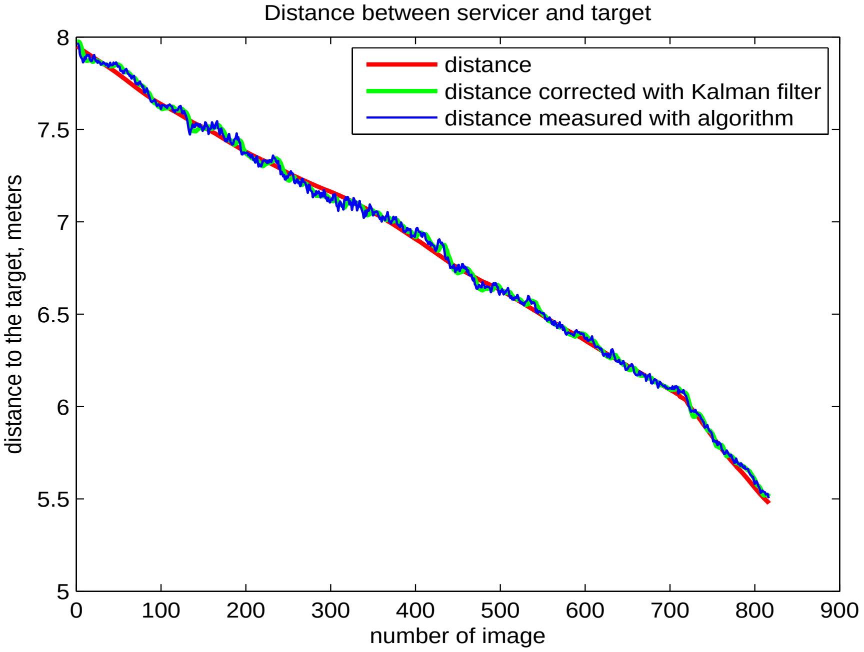


Figure 4.17: Distance during the approach, measured distance with the algorithm and distance corrected with Kalman filter.

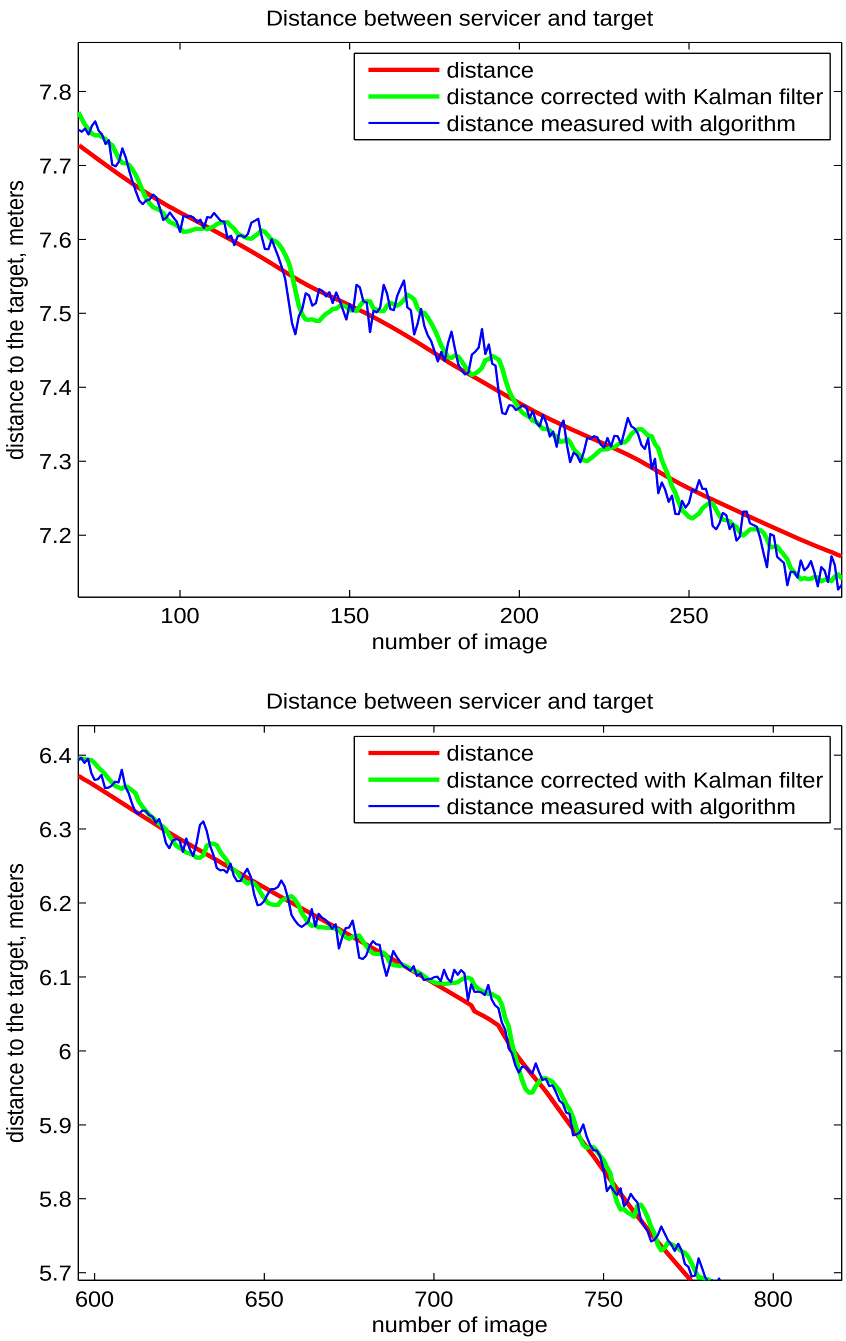


Figure 4.18: (a): Zoom into the distance component at the beginning of the tracking in the range from 7.75 meters to 7 meters. (b). Zoom into the distance component at the end of the tracking in the range from 6.35 meters to 5.7 meters.

4.4.1 Tests with PMD sensor in a Total Darkness

Since the PMD sensor is an active camera, which has its own illumination unit for calculation of depth measurements, I am going to introduce the results of the visual navigation in a closed loop without any additional illumination. Thanks to the EPOS laboratory, the complete darkness environment was created by switching off all lightning in the hall and also the sun simulator. The close approach trajectory is the same as I used above. I repeated the close approach three times. In Table 4.9, the root mean square (RMS) errors for the position and orientation components for three approach scenarios are shown.

Table 4.9: Root mean square errors for position and orientation using the PMD sensor in a closed-loop for 3 approach trajectories without any illumination in the hall.

RMS	1	2	3
$roll_{rms}$, deg	0.4385	0.4481	0.4715
$pitch_{rms}$, deg	0.2285	0.1999	0.2321
yaw_{rms}, deg	1.1191	1.0538	1.0114
z_{rms}, meters	0.0401	0.0399	0.0391
y_{rms}, meters	0.0202	0.0201	0.0169
x_{rms}, meters	0.0053	0.0068	0.0075

From Figure 4.19, one can observe that the approach with the single PMD sensor in a complete darkness can be successfully performed. The tracking is stable. The PMD sensor is totally adaptable to work without any ambient light.

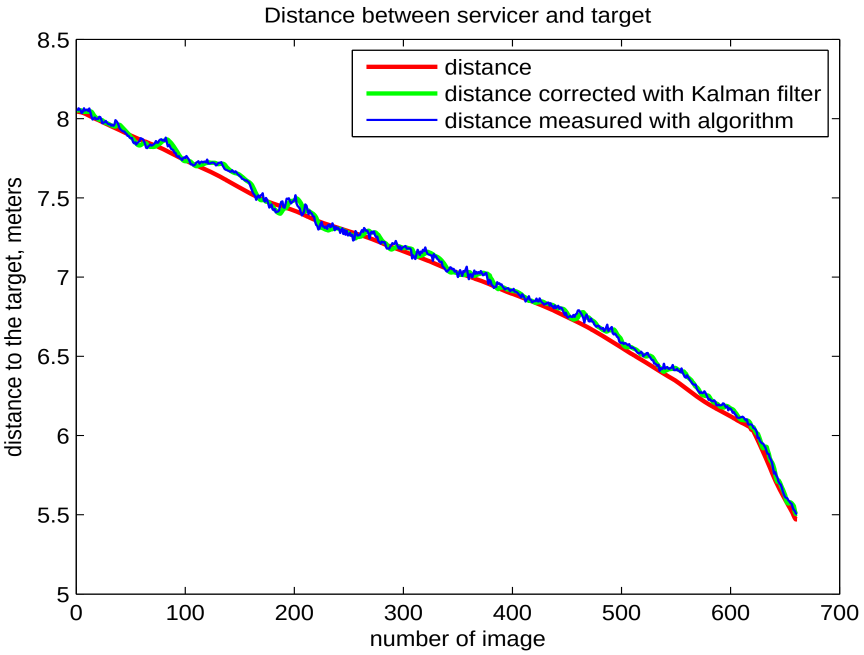


Figure 4.19: Test in a total darkness. Distance during the approach, measured distance with the algorithm and distance corrected with Kalman filter.

The rotation and position errors for the test number 1 are presented in Figures 4.20-4.21. The rotation errors are just few degrees. The position errors in this test look similar to the results presented in Figure 4.16.

Figure 4.20: Test in a total darkness. Rotation errors during rendezvous with PMD sensor in the closed loop for the test 1.

Figure 4.21: Test in a total darkness. Position errors during rendezvous with PMD sensor in the closed loop for the test 1.

With this test I have shown one of the important and positive features of the PMD sensor - its ability to perform in a darkness. The visual navigation system is totally independent on illumination conditions.

4.5 Comparison of Visual Navigation with PMD and CCD sensors

In this section, I am going to compare the open-loop performances of the PMD sensor with a charge coupled device (CCD) sensor. I have chosen the same rendezvous trajectory as in the previous section. For the comparison of the results, the coordinate system of the chaser satellite is chosen. The chaser coordinate system is presented in blue in Figure 1.12. A Prosilica Gigabit Ethernet GC-655 vision camera with CCD sensor is used to measure the pose of the target during the rendezvous phase. This sensor is also mounted on the breadboard of the Robot 1 at EPOS laboratory, see Section 2.3. Some technical characteristics of the CCD camera are presented in Table 4.10. Noticeably, the resolution of the sensor chip is almost twice as much as the PMD sensor. The FOV of the CCD camera is also bigger.

Table 4.10: Technical data of Prosilica GC-655 camera

Resolution of the chip	640×480 pixels
focal length	$[604, 604]$ pixels
pixel size	9.9×10^{-6} m

The setup for the open-loop performance comparison is as follows: I run in parallel two visual navigation techniques with PMD and CCD sesnors. The first one is the navigation technqiue presented in this thesis and uses PMD sensor (see Section 3.3), and the second one has been developed by the colleagues of the OOS group of DLR and uses aforementioned CCD sensor [163]. Before I discuss the numerical results after the executed test, it is interesting to compare the images from both visual sensors.

In order to have an idea how two gray-scaled images differ - an image from CCD camera and an amplitude image from PMD sensor, I plot both of them taken in the same moment in Figure 4.22. Since the CCD camera does not provide a depth image, the depth image from PMD sensor is omitted. Since the image from PMD sensor has the lower size, I have resized it up to the size of CCD image for better presentation. Because of the different FOVs, the target in amplitude images appears bigger than in the 2D CCD image. High contrast of the CCD gray-scaled images and higher resolution of the CCD chip makes the target more clear and sharp than in the PMD gray-scaled image.

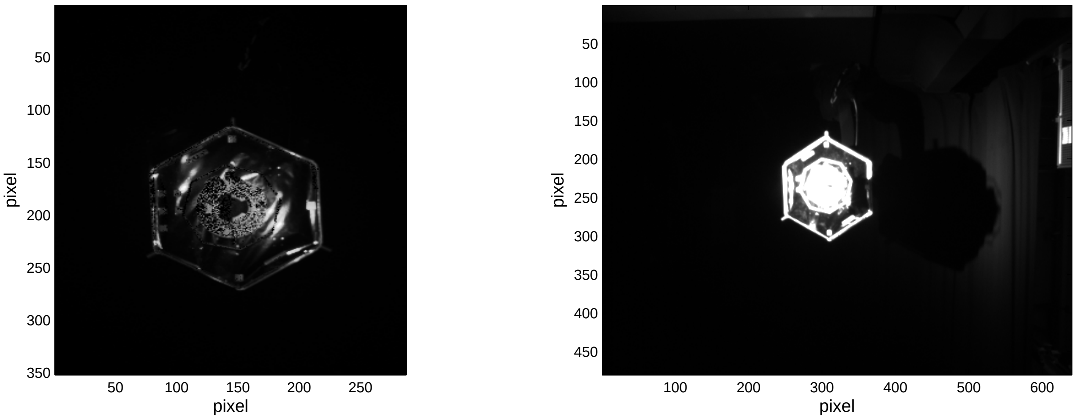


Figure 4.22: (a): PMD gray-scaled image. (b) CCD gray-scaled image.

I simulate the rendezvous approach from 8 meters to 5.5 meters, with a safe approach velocity of 0.01 m/sec. This rendezvous scenario was repeated 10 times. In Figure 4.23 and Figure 4.24, the errors in position and orientation with PMD and CCD sensors are presented for the test number 6.

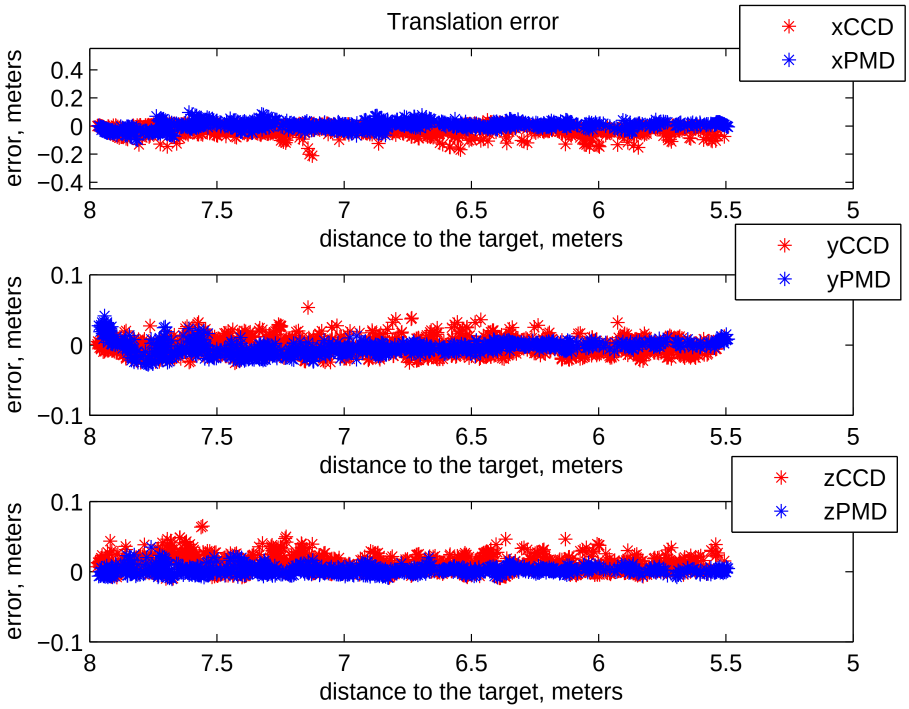


Figure 4.23: Position errors for the PMD and the CCD sensors.

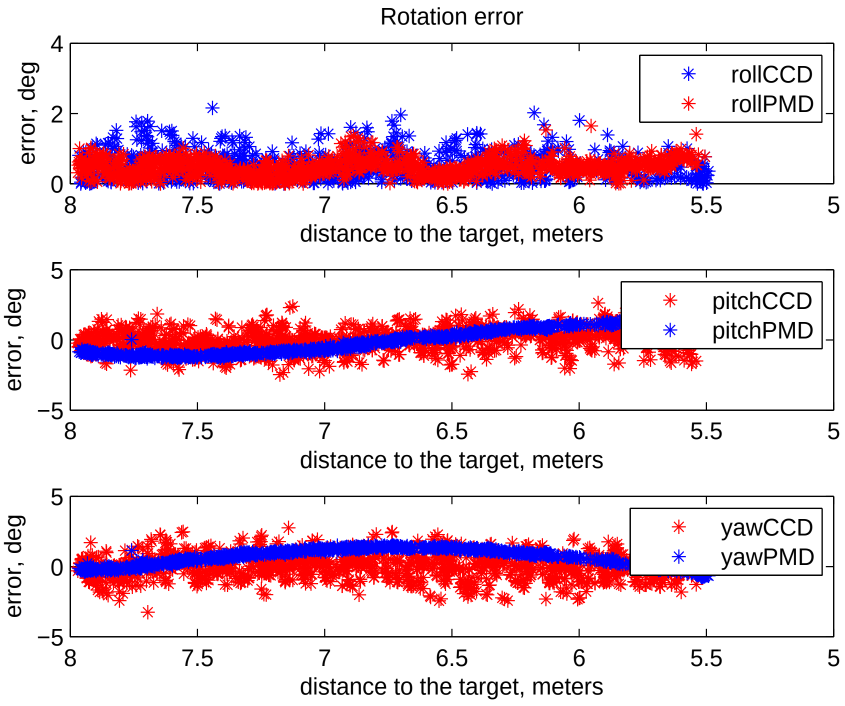


Figure 4.24: Rotation errors for the PMD and the CCD sensors.

In Table 4.11 and Table 4.12, I show the root mean square (RMS) errors for the position and orientation using the PMD sensor and the CCD camera. The statistical characteristics are presented for the 10 approach trajectories.

Table 4.11: Root mean square errors for position and orientation using the PMD sensor.

RMS	1	2	3	4	5	6	7	8	9	10
$roll_{rms}$, deg	0.6360	0.6713	0.6795	0.6921	0.6722	0.6474	0.6339	0.6164	0.6561	0.6495
$pitch_{rms}$, deg	0.7342	0.6217	0.7075	0.8047	0.8274	0.8896	0.9421	1.0000	0.9850	1.0394
yaw_{rms}, deg	0.9520	0.8790	0.9088	0.9192	0.9082	0.9087	0.9645	0.9509	0.9650	0.9809
x_{rms}, meters	0.0310	0.0235	0.0244	0.0275	0.0226	0.0286	0.0260	0.0265	0.0275	0.0289
y_{rms}, meters	0.0191	0.0085	0.0096	0.0128	0.0121	0.0104	0.0086	0.0115	0.0173	0.0183
z_{rms}, meters	0.0088	0.0062	0.0055	0.0080	0.0053	0.0057	0.0074	0.0075	0.0060	0.0069

Table 4.12: Root mean square errors for position and orientation using the CCD sensor.

RMS	1	2	3	4	5	6	7	8	9	10
$roll_{rms}$, deg	0.5022	0.5235	0.5249	0.5348	0.5248	0.5167	0.4972	0.5304	0.5332	0.5145
$pitch_{rms}$, deg	0.8552	0.7884	0.8075	0.8422	0.8313	0.7948	0.8305	0.8568	0.8307	0.8859
yaw_{rms}, deg	0.8297	0.8077	0.8396	0.9247	0.8452	0.8496	0.8746	0.7628	0.8295	0.8346
x_{rms}, meters	0.0420	0.0393	0.0422	0.0408	0.0422	0.0393	0.0383	0.0382	0.0375	0.0360
y_{rms}, meters	0.0135	0.0110	0.0106	0.0133	0.0120	0.0115	0.0104	0.0117	0.0126	0.0124
z_{rms}, meters	0.0193	0.0168	0.0109	0.0198	0.0165	0.0154	0.0115	0.0119	0.0096	0.0106

For the better comparison of the final results I present Table 4.13, where the average of all RMS errors for both sensors are included.

Table 4.13: The average result of 10 tests.

average	PMD sensor	CCD sensor
$roll_{rms_{avg}}$, deg	0.6554	0.5202
$pitch_{rms_{avg}}$, deg	0.8552	0.8323
$yaw_{rms_{avg}}$, deg	0.9337	0.8398
$x_{rms_{avg}}$, meters	0.0267	0.0396
$y_{rms_{avg}}$, meters	0.0128	0.0119
$z_{rms_{avg}}$, meters	0.0067	0.0142

The obtained results of both pose estimation techniques with the CCD and PMD sensors in an open-loop showed a good, stable tracking. From Table 4.13, one can see that the averages of the RMS errors for the rotational components estimated with PMD and CCD sensors over 10 tests are slightly different. The average measured attitude is a bit more accurate with the CCD sensor. The average of the RMS errors of the estimated distance (along X axis) with PMD sensor (0.0267 meters) is better than of the CCD sensor (0.0396 meters). When one has a look at Tables 4.11-4.12, we can notice that there is not any test, where the distance to the target is measured more accurate with CCD rather than with PMD camera. Moreover, the PMD sensor was able to estimate with a better precision the position along the Z axis.

For the current work it was sufficiently to show the general behavior of the PMD and CCD sensors. I think, the identical tendency for the estimated state vector will be obtained if the same pose estimation algorithm processes images of PMD and CCD sensor.

4.6 Computational Complexity

It is very important to evaluate the computational complexity of the algorithm during the developing phase (even if it could be only estimated roughly) in order to have an idea, if the algorithm is able to guarantee a good performance with real on-board computers (OBC) used for the spacecrafts. The demonstration of the time complexity assists not only for the further choice of the OBC, but also could push us to make thoughts how to optimize the algorithms and reduce the time complexity.

Below I show experimental analyses of the computational complexity of the proposed pipelines for the pose acquisition and tracking. The computer, which I used for execution, has a Debian GNU/Linux 9.5 (stretch) operating system. The characteristics of the CPU are presented in Table 4.14.

4.6.1 Computational Complexity of Pose Initialization Technique

I run pose initialization technique for different images in the range from 8 to 6 meters. The main factor which strongly influences the time complexity of the pose estimation procedure is the number of scene points, since the number of model points are defined as a constant throughout all simulations. In Figure 4.25, I depict the CPU computational

Table 4.14: CPU characteristics of the processor used for tests.

Architecture	x86_64
Order	Little Endian
CPU(s)	6
On-line CPU(s) list	0-5
Vendor ID	GenuineIntel
CPU family	6
Model	62
Model name	Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz
BogoMIPS	6982.36

times, which were spent to estimate the initial pose with different sizes of the scene point cloud. After evaluation of the data I have found out that the computational complexity suits to the polynomial regression $T(n) = O(n^k)$, where n denotes the number of scene points. The degree of the regression k is found out to be 5. I estimate the standard error and plot the 95% prediction interval of CPU computational time $t_{CPU\_comp} \pm 2\triangle$.

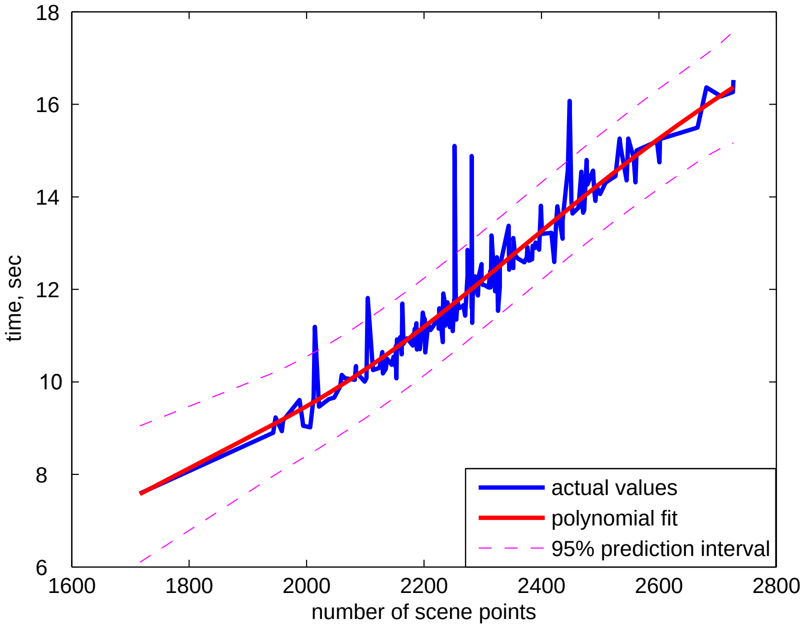


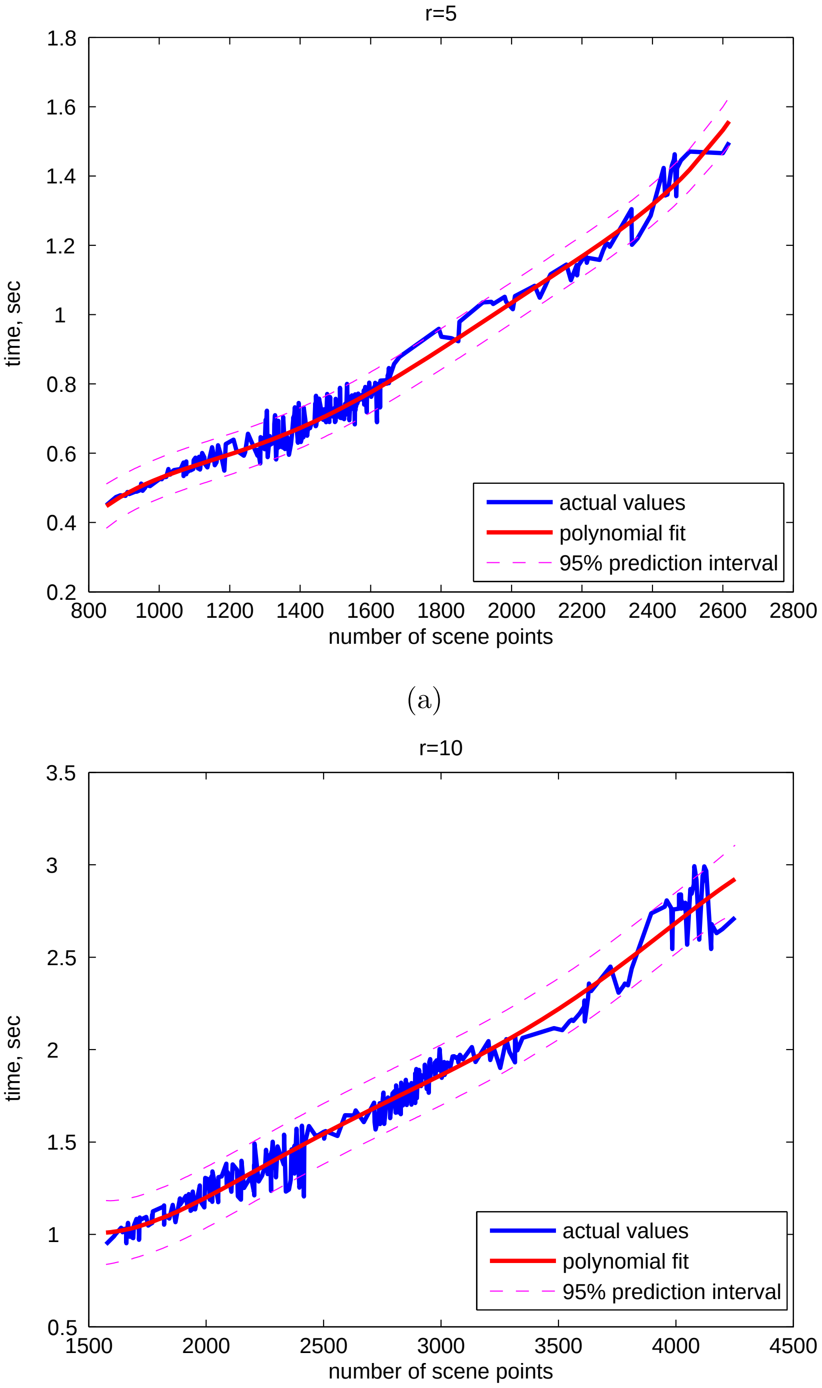
Figure 4.25: Computational complexity of the pose acquisition technique. Polynomial fit of data with 95% prediction interval.

The increase of the scene point cloud usually occurs when the distance between the space target and the camera decreases. It means, the closer chaser is, the bigger the point cloud of the scene and longer the CPU computational time of the pose acquisition. From Figure 4.25, one can observe the minimum processing time for one pose execution corresponds to $t_{min} = 7.58$ seconds and the maximum $t_{max} = 16.52$ seconds.

4.6.2 Computational Complexity of the Tracking Technique

Let us evaluate the performance of the tracking technique. I estimate the CPU processing time to estimate one pose using the fusion technique, since I determined this approach as very robust during the tracking with the current PMD sensor. The main factor, which plays role for the evaluation of the time complexity, is the same as in the previous case - a number of taken scene points. For the tracking technique the major parameter, which changes the amount of points, is mainly the chosen radius for the neighbor search. Check Section 3.3.1 for the description of the radius. Shortly, the larger the radius, the bigger scene point cloud. Furthermore, the quality of the depth image regulates the number of the scene points, but not so drastically as the radius of search. I present the following Figures 4.26-4.27, where the spent CPU times depending on the number of scene points are introduced. Every plot is for the specific radius for the neighbor search.

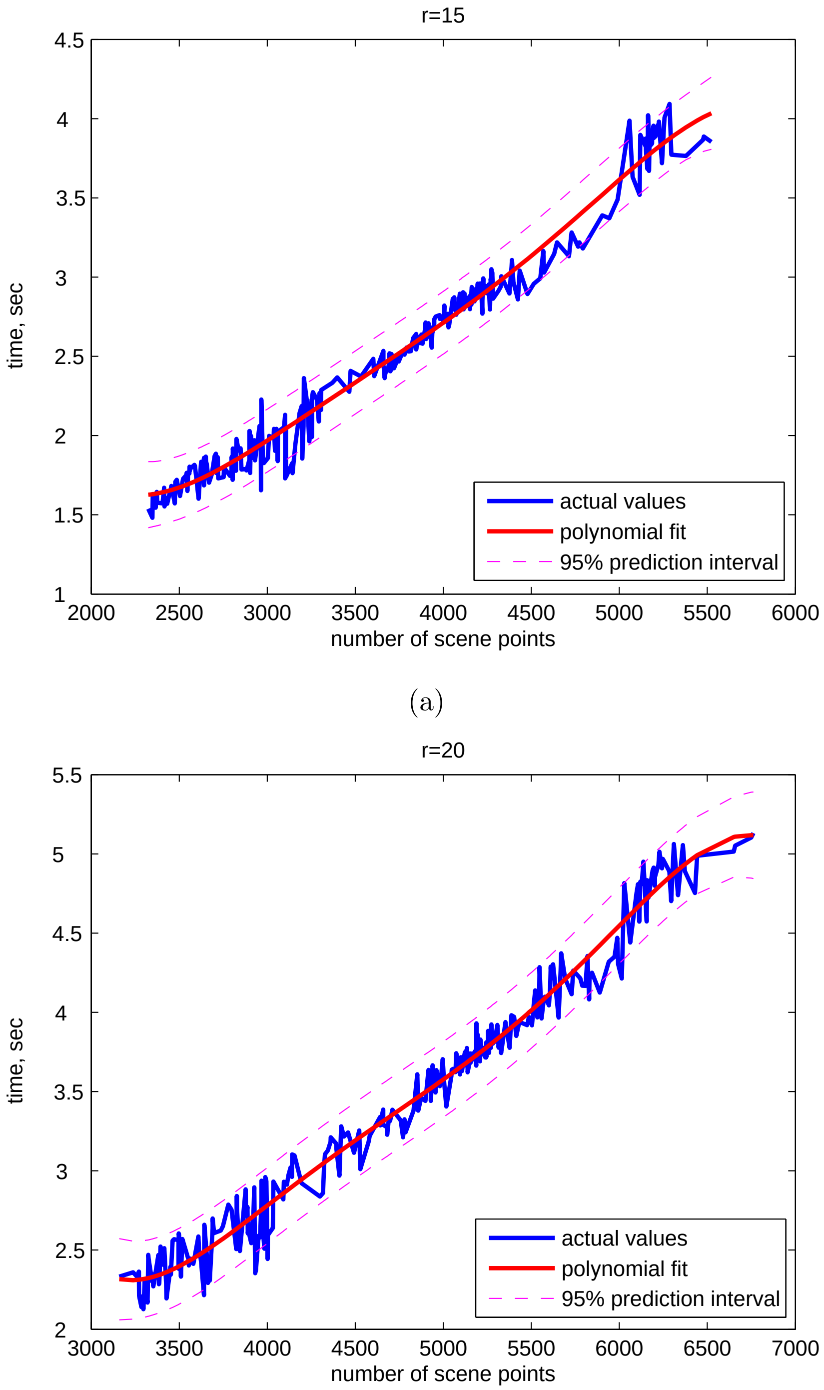
After some experimental tests I can make a statement that the computational complexity of the hybrid navigation technique for the tracking is shown to be polynomial. The plots in Figure 4.26 show the time complexity depending on the size of scene point cloud by radius of neighbor search r=5 and r=10. The plots in Figure 4.27 show the same with the bigger radii equal to r=15 and r=20. With the calculated and depicted prediction intervals (see Figure 4.26 and Figure 4.27) in 95% one can suppose that having the certain number of scene points the computational time will be within determined limits.



(a)

(b)

Figure 4.26: (a)-(b) Computational complexity of the tracking technique. Polynomial fit of data with 95% prediction interval, radii for the neighbor search are r=5 and r=10.



(a)

(b)

Figure 4.27: (a)-(b) Computational complexity of the tracking technique. Polynomial fit of data with 95% prediction interval, radii for the neighbor search are r=15 and r=20.

CPU time spent for the calculation of one pose depends on a software application, more specific, on number, types and frequency of executed instructions. In generally, an instruction is an order given to processor by a computer program. Nowdays, it is easy to calculate the CPU execution time for a process, but it is difficult to find out the instruction count or number of clock cycles per instruction (CPI). The simple equation for the CPI time looks like [170, 171]:

$$t_{CPU\_comp} = Instruction\ count - \frac{CPI}{Clock\ rate}. \tag{4.1}$$

Since the CPU time depends on three different factors, I provide a brief parameter analysis in order to decrease the CPU time. Clock rate depends on the hardware technologies of the processor. CPI depends on organization of the cycles, instruction set architecture (ISA) and compiler. Instruction count can be changed by varying the ISA and compiler as well. One can notice that by improving one of these components it will have an impact on the others.

To sum up, for the choice of the chaser's OBC for the future rendezvous scenarios to non-cooperative targets it is necessary to pay attention on detailed description of scenario, hardware setup (e.g. visual sensors), and complexity of the software for the rendezvous. For sure, it is difficult task to find the trade-off between many parts of the complex system.

Chapter 5

Conclusion

From the scientific point of view, it is interesting to sum up the results of the research presented in this thesis and to propose some ideas for the further researchers in a field of pose estimation with the PMD sensor.

This PhD thesis represents a theoretical and practical research in the branch of proximity operations for On-Orbit Servicing and Active Debris Removal missions. It is very important to extend the existing knowledge about the new hardware and software possibilities and share them with the scientific communities. The more we try to simulate the missions scenarios on the ground, the faster we can start a real mission for the space debris mitigation. The primary goal of the thesis was to develop pipelines for pose estimation (position and orientation) of the target spacecraft using a PMD sensor. For the pose estimation the 3D target model was available. In particular, the challenge of my thesis is to create a stable tracking technique, which allows to chaser approach a target with PMD images.

The first contribution of the thesis to the state-of-the-art is a creation of a Hybrid Navigation Technique (HNT), where amplitude and depth images of the PMD sensor are used independently for pose estimation. This method processes the gray-scaled amplitude image in parallel with the depth image and gives to the system the redundant state measurement. The HNT is able to cope with two of the strong drawbacks of the sensor: its inability to work under the strong ambient light and with high reflective or absorbing materials. When the depth measurements are (partly) absent, the tracking will not fail. The chaser continues its approach only with amplitude images. In case when both of them have a good quality, the amplitude image serves as an additional data source for the pose improvement. I was able to fuse both estimates and use a fused state as a single pose. The weights for the fusion technique were calculated in a post-processing phase, when each of the algorithms operated solely. It was proved that the data fusion has its best performance than depth or amplitude pose estimation techniques separately.

The proposed feature of the amplitude image made also the second contribution for pose initialization. At the first stage, the pose was acquired with the depth image and a known 3D model. The initial assessment of the rotation component with a point cloud of the PMD image is a difficult task and lead to some errors. For the follow-up correction of the initial estimated pose, the pose estimation with an amplitude image was applied. The author demonstrated evolution of accuracy of the rotational components with a small resolution gray-scaled image.

The third contribution is integration of the PMD visual navigation in a GNC system of the OOS group at DLR. It gave us a great possibility to test the developed techniques in a closed-loop with the EPOS simulator. The evaluation of the performances with the DLR-Argos3D sensor totally satisfied the assigned tasks with an acceptable accuracy (position errors less than 10 cm and rotation errors less than 5 deg). I also compared in an open-loop tests the performance of the CCD and PMD sensors for the same tracking scenario. As it was expected, the depth PMD sensor showed better results for measurements of the relative distance component throughout 10 tests. The CCD sensor provided slightly better results of the estimated attitude. Moreover, the PMD sensor has one great advantage in compare to CCD sensor: it can be used in a completely darkness without any additional light source, whereas a flash light is necessary for the CCD sensor in umbra phase.

It is interesting and important to know, if the developed techniques can be used with other time-of-flight sensors, which have different characteristics. I am trying to answer this question in the next paragraph.

5.1 Can the Presented Pose Estimation Algorithms be Applied with other Active Sensors?

Let us denote that here active sensors are considered, which use the time-of-flight/phase shift depth measurement principle in order to get a raster image. As I already defined, there are two main pose estimation methods - the first one for the pose initialization and the second one for the pose tracking. In case of the pose acquisition the initial position and attitude are calculated with the depth image. The point cloud of the DLR-Argos3D Camera and the pose estimation technique with the Point Feature Vectors described in Section 3.2 are applied. Can one use the same pose acquisition technique and the current mockup but another active sensor? The answer will be yes, if another active sensor is able to provide the 3D point clouds, which reflect the features of the observable object. It means, the resolution and the FOV of the depth sensor must be big enough in order to fit the mockup into the image. The FOV depends on the focal length and the chip size (length and width).

Let us imagine a case, when the resolution of the sensor is even bigger than I have at the moment. Evidently, it will be considered as an advantage, since the details of the object are better visible. On the contrary, if the resolution of any other PMD sensor is noticeably smaller, e.g. one of the previous PMD sensors had the resolution 64 × 48, there is no guarantee to have a good quality of the point cloud and, respectively, the great accurate output of the acquired pose.

Working with the PMD sensor one should not forget that it is an active sensor, and the depth measurements are dependent on the power of the illumination unit integrated in the camera kit. This fact is in close correlation between the FOV of the sensor and the good reliable depth measurements. In other words, by increasing the FOV of the camera, the sensor is able to provide images of the full object even in the close vicinity between given mockup and the sensor. But then the weakness will arise when one is going to work in a long distance range. There will be a need for more high-power LEDs units and robust suppression of the ambient illumination.

Now the pose estimation algorithm is considered, which is used during the tracking phase for the depth images and described in Section 3.3.1. Can one replace the existent PMD sensor with another one without any consequences for the operation of the algorithm? The main prerequisite to use the ICP algorithm with the reverse calibration is the raster depth image. It means that for the group of sensors defined in this section the algorithm can be applied. The main factors which can influence the performance of the algorithm are the same as with the previous algorithm: FOV and resolution of the sensor. For the rendezvous with non-cooperative target the trade-off must be found concerning the approach scenario and the space object.

What about the pose estimation technique for the amplitude image described in Section 3.3.2?[1] I propose this technique for the correction of the initial pose, as well as for the hybrid navigation technique during frame-to-frame tracking. Can one expect a good performance of the algorithm with another PMD sensor and the same mockup? Interestingly that here the main dependencies are same as for the two previous mentioned pose estimation techniques with the depth image - the resolution of the senor and FOV. It is commonly known that the accuracy of the estimated pose with monocular vision is more precise if the resolution of the image sensor is higher, because of smaller pixel noise. Especially it concerns the distance measurement. Therefore, it is preferably to choose a higher resolution of the PMD sensor for the proper work of the algorithm with the amplitude images. The small FOV limits the observation of the mockup in the close vicinity.

To sum up, all presented algorithms for the pose estimation of the non-cooperative target can be processed with other active sensors with a raster image, when the resolution of the sensor chip and FOV are big enough in order to provide full information of the chosen features of the space object. In general, the higher resolution, the better received data from the sensor.

5.2 Future work

In Sections 1.3-1.5, I defined the objectives and tasks of the thesis. All these research questions, such as development and validation of the pose initialization and tracking techniques, as well as calibration of the PMD sensor have been successfully completed. The more deeply immersed in the topic, the more new questions arise. From my point of view, it would be interesting extending the length of the close rendezvous phase, and considering distances less than 5 meters. In this case it will be required to choose other features, e.g. octagon, since the hexagon, the outer contours of the target mockup, will be no longer in the FOV of the camera. For this purpose, the current pose estimation methods must be improved and adjusted, or a new pose estimation technique for the tracking phase must be developed.

Currently the DLR-Argos3D - P320 camera has one IR illumination unit, which emits the signal. As far as I know, there is a possibility to extend the illumination unit with some additional ones. This brings us more illumination power and, as consequently, it will allow to increase the measurement range of the sensor. For example, when using the

---

[1] The algorithm for the amplitude image can be applied also for other cameras (like CCD, CMOS) which deliver a gray-scaled image and have a sufficient FOV and resolution.

modulation frequency of 5 MHz the sensor should provide depth measurements in the range up to 30 meters. Nowadays, using DLR-Argos3D - P320 camera it is quite hard to get the reliable depth images with the modulation frequency of 5 MHz, because due to insufficient power of the IR illumination unit.

What probably can be also interesting to consider is the use of the second PMD sensor. The configuration of two PMD sensors can be considered as a stereo system by using the gray-scaled amplitude images. It means, there will be three point clouds (two from depth images of each sensor and third one can be obtained via stereo vision, i.e. the triangulation principle for two amplitude images) of the imaging scene, three sources of depth information. The depth data from stereo configuration can be examined to substitute the lost or incorrect depth data in the PMD depth images, when this case occurs.

Future work in the field of software part, namely, a development of other suitable techniques for pose estimation with the PMD sensor can be implemented. Nowadays, the branch of computer vision is enormous. It would be interesting to try deep learning algorithms with the data from the PMD sensor and compare its performance with the state-of-the-art computer vision techniques.

Moreover, an integration of the provided software into any available prototype of an on-board computer is an invaluable contribution for the research. Tests of the developed methods with the real on-board computers allow evaluate their future applicability in the real OOS missions. It will also help to optimize and improve any of the proposed software components on the ground. The colleagues from the OOS group already work on integrating the GNC system on the ScOSA on-board computer [172]. The next step would be also to integrate the PMD algorithms.

The other investigation can be done in the following direction - an implementation of the current image processing algorithms in hardware (if it is possible), e.g. with a Field Programmable Gate Array (FPGA) [173, 174]. The outstanding features of the FPGAs for image processing are optimization of imaging operations, high computational capacity and low cost. Working with FPGA, one has the possibility to execute independently the instructions of the algorithm in parallel, each of them on a separate hardware module, thereby increasing the speed of the execution. Having numerical results after such implementations one can compare performance results of the algorithms executed in software and in hardware.

Bibliography

[1] International Space University. DOCTOR: Developing On-Orbit Servicing Concepts, Technology Options, and Roadmap. *Final Report,Summer Session Program 2007*, 2007.

[2] OECD. *Space 2030 Exploring the Future of Space Applications: Exploring the Future of Space Applications*. OECD Publishing, 2004.

[3] *Torn of Solar Panels*. Available at `https://www.space.com/4586-space-station-surgery-astronauts-mend-torn-solar-wing.html`. Last accessed: February, 18, 2019.

[4] *NASA*. Available at `http://www.nasa.gov/mission_pages/shuttle/shuttlemissions/sts120/multimedia/fd8/fd8_gallery.html`. Last accessed: February, 18, 2019.

[5] *NASA*. Available at `http://spaceflight.nasa.gov/gallery/images/shuttle/sts-120/html/iss016e008937.html`. Last accessed: February, 18, 2019.

[6] *ESA: Space Debris*. Available at `http://www.esa.int/Our_Activities/Operations/Space_Debris/About_space_debris`. Last accessed: February, 18, 2019.

[7] W. Fehse. *Automated Rendezvous and Docking of Spacecraft*. Cambridge Aerospace Series. Cambridge University Press, 2003.

[8] J. W. Wertz and R. Bell. Autonomous Rendezvous and Docking Technologies Status and Prospects. *Proceeding SPIE 5088, Space Systems Technology and Operations*, 2003.

[9] *flickr*. Available at `https://www.flickr.com/photos/nasa2explore/9452428826/in/album-72157634958743610/`. Last accessed: February, 18, 2019.

[10] *Pochta Polevaya*. Available at `https://pochta-polevaya.ru/aboutarmy/calendar/date_history/13541.html`. Last accessed: February, 18, 2019.

[11] G. Arantes Jr. and L. S. Martins-Filho. Guidance and Control of Position and Attitude for Rendezvous and Dock/Berthing with a Noncooperative/Target Spacecraft. *Mathematical Problems in Engineering*, 2014.

[12] B. W. Barbee, J. R. Carpenter, Heatwole F. S., L. Markley, M. Moreau, B. J. Naasz, and J. V. Eepoel. Guidance and Navigation for Rendezvous and Proximity Operations with a Non-Cooperative Spacecraft at Geosynchronous Orbit. *George H. Born Symposium*, 2010.

[13] O. Montenbruck, M. Kirschner, S. D'Amico, and S. Bettadpur. E/I-vector separation for safe switching of the GRACE formation. *Aerospace Science and Technology*, 2006.

[14] S. D'Amico, J. S. Ardeans, G. Gaias, H. Benninghoff, B. Schlepp, and J. Jorgensen.

Noncooperative Rendezvous Using Angles-Only Optical Navigation: System Design and Flight Results. *Journal of Guidance, Control, and Dynamics, 36 (6)*, 2013.

[15] T. Ringbeck and B. Hagebeuker. A 3D Time of Flight Camera for Object Detection. *Oprtical 3-D Measurement Techniques, ETH Zürich*, 2007.

[16] *Bluetechnix*. Available at https://www.bluetechnix.com. Last accessed: February, 18, 2019.

[17] ESA. "Jules Verne" Automated Transfer Vehicle (ATV). *Information Kit*, 2008.

[18] B. Cavrois, A. Vergnol, A. Donnard, P. Casiez, and O. Mongrard. LIRIS Demonstrator on ATV5: a Step beyond for European Non-cooperative Navigation System. *AIAA Guidance, Navigation, and Control Conference*, 2015.

[19] *JAXA*. Available at http://global.jaxa.jp/projects/sat/ets7/index.html. Last accessed: February, 18, 2019.

[20] *NASA*. Available at http://spaceflight.nasa.gov/gallery/images/station/crew-16/html/iss016e034176.html. Last accessed: February, 18, 2019.

[21] *Overview of the DART Mishap Investigation Results*. Avaliable at https://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf. Last accessed: February, 18, 2019.

[22] R. T. Howard, A.(N.) Johnston, T. C. Bryana, and M. L. Book. Advanced Video Guidance Sensor (AVGS) Development Testing. *Spacebome Sensors*, 5418:50–60, 2004.

[23] *NASA*. Available at https://en.wikipedia.org/wiki/DART_(satellite). Last accessed: February, 18, 2019.

[24] *Wikipedia*. Available at https://en.wikipedia.org/wiki/Orbital_Express. Last accessed: February, 18, 2019.

[25] *PRISMA: DLR Experiment on Autonomous Satellite Formation Flight Begins*. Available at http://www.dlr.de/rd/en/desktopdefault.aspx/tabid-6238/10300_read-26824/. Last accessed: February, 18, 2019.

[26] *AVANTI - Paving the Way to Space Debris Removal*. Available at http://www.dlr.de/rb/desktopdefault.aspx/tabid-11685/. Last accessed: February, 18, 2019.

[27] *Researchgate*. Available at https://www.researchgate.net/figure/PRISMA-satellites-Mango-right-and-Tango_fig1_235718850. Last accessed: February, 18, 2019.

[28] *DLR*. Available at https://www.dlr.de/dlr/en/desktopdefault.aspx/tabid-10460/685_read-19916/year-all/#/gallery/24880. Last accessed: February, 18, 2019.

[29] *MEV-1*. Available at https://www.aiaa.org/news/industry-news/2020/04/20/northrop-grumman-s-mev-1-spacecraft-successfully-docks-with-intelsat-is-901-sate
Last accessed: June, 21, 2020.

[30] *Restore-L*. Available at https://nexis.gsfc.nasa.gov/OSAM-1.html. Last accessed: June, 21, 2020.

[31] *Clean Space: e.Deorbit*. Available at http://www.esa.int/Our_Activities/Space_Engineering_Technology/Clean_Space/e.Deorbit. Last accessed: February, 18, 2019.

[32] *Orbital ATK*. Available at https://spacenews.com/orbital-atk-unveils-new-version-of-satellite-servicing-vehicle/.

Last accessed: February, 18, 2019.

[33] *DARPA-RSGS*. Available at `https://www.darpa.mil/program/robotic-servicing-of-geosynchronous-satellites`. Last accessed: February, 18, 2019.

[34] S. Se, P. Jasiobedzki, T. Pan, M. Umasuthan, and M. Greenspan. Autonomous Satellite Rendezvous and Docking using LIDAR and Model Based Vision. *In Proceedings SPIE Spaceborne Sensors II*, 2005.

[35] F. Rems, S. Fritz, and T. Boge. Breadboard Model of a 3D LIDAR Sensor for Real-Time Pose Estimation of Spacecraft. *9th International ESA Conference on Guidance, Navigation and Control Systems, Porto, Portugal*, 2014.

[36] J. A. Christian, H. Hinkel, C. N. D'Souza, S. Maguireand, and M. Patangan. The Sensor Test for Orion RelNav Risk Mitigation (STORRM) Development Test Objective. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, 2011.

[37] F. Amzajerdian, V. E. Roback, P. F. Brewster, G. D. Hines, and A. Bulyshev. Imaging Flash Lidar for Autonomous Safe Landing and Spacecraft Proximity Operation. *AIAA Space 2016 Conference, Long Beach, CA, United States*, 2016.

[38] Y. Gao. *Contemporary Planetary Robotics: An Approach Toward Autonomous Systems*. Wiley-VCH; 1 edition, 2016.

[39] *ELIB*. Available at `https://elib.dlr.de/89757/1/GNC_2014_Paper_F_Rems.pdf`. Last accessed: February, 18, 2019.

[40] *ESA*. Available at `http://blogs.esa.int/orion/2014/03/19/liris-laser-infra-red-imaging-sensors-demonstrator-on-atv-5/`. Last accessed: February, 18, 2019.

[41] *Advanced Scientific Concepts*. Available at `http://www.advancedscientificconcepts.com/index.html`. Last accessed: February, 18, 2019.

[42] *NASA*. Available at `https://www.spaceflight.nasa.gov/gallery/images/shuttle/sts-128/html/iss020e035665.html`. Last accessed: February, 18, 2019.

[43] E. Gill, O. Montenbruck, S. D'Amico, and S. Persson. Autonomous Satellite Formation Flying for the PRISMA Technology Demonstration Mission. *16th AAS/AIAA Space Flight Mechanics Conference,Tampa, Florida*, 2006.

[44] J. L. Jorgensen, T. Denver, M. Betto, and P. S. Jorgensen. Microasc a Miniature Star Tracker. *Technical University of Denmark, Dept. of Measurement and Instrumentation*, 2001.

[45] J. L. Jorgensen and M. Benn. VBS - The Optical Rendezvous and Docking Sensor for Prisma. *NordicSpace*, pages 16–19.

[46] *Earth Observation Portal*. Available at `https://directory.eoportal.org/web/eoportal/satellite-missions/p/prisma-prototype`. Last accessed: February, 18, 2019.

[47] *Earth Observation Portal*. Available at `https://directory.eoportal.org/web/eoportal/satellite-missions/content/-/article/iss-utilisation-storrm`. Last accessed: February, 18, 2019.

[48] *NEPTEC*. Available at `http://neptecuk.com/`. Last accessed: February, 18, 2019.

[49] K. Schilling and L. Regoli. The PMD 3D-camera for Rendezvous and Docking with Passive Objects. *8th International ESA Conference on Guidence, Navigation and*

Control Systems, 2011.

[50] J. Scharnagl, L. Regoli, and K. Schilling. On-line Collision Detection in Space Using Photonic Mixer Devices. *19th IFAC Symposium on Automatic Control in Aerospace, Würzburg, Germany*, 2013.

[51] T. 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. *Dissertation, University Würzburg*, 2014.

[52] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka. Rigid 3D Geometry Matching for Grasping of Known Objects in Cluttered Scenes. *IJRR*, 31(4), 2012.

[53] I. Lenz, L. Honglak, and A. Saxena. Deep Learning for Detecting Robotic Grasps. *The International Journal of Robotics Research*, 34, 2015.

[54] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. 3D object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11), 2014.

[55] T. Rabbani and F. van den Heuvel. Efficient Hough Transform for Automatic Detection of Cylinders in Point Clouds. *ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005", Enschede, the Netherlands*, 2005.

[56] R. Schnabel, R. Wahl, and Klein R. Efficient RANSAC for Point Cloud Shape Detection. *The Eurographics Association and Blackwell Publishing*, 2007.

[57] F. Rems, J. A. Moreno Gonzalez, T. Boge, S. Tuttas, and U. Stilla. Fast Initial Pose Estimation of Spacecraft from LIDAR Point Cloud Data. *13th Symposium on Advanced Space Technologies in Robotics and Automation*, 2015.

[58] C. Long, G. Bao-long, and S. Wei. Relative Pose Measurement Algorithm of Non-cooperative Target based on Stereo Vision and RANSAC. *International Journal of Soft Computing And Software Engineering (JSCSE)*, 2(4), 2012.

[59] R. B. Rusu, G. Bradski, R. Thibaux, J. Hsu, and W. Garage. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. *The IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan*, 2010.

[60] J. O. Woods and J. A. Christian. LIDAR-based Relative Navigation with Respect to Noncooperative Objects. *Acta Astronautica*, 126, 2016.

[61] A. S. Mian, M. Bennamoun, and R. Owens. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 2006.

[62] M. Hebert and A. Johnson. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scene. *In Proceedings SPIE Spaceborne Sensors II*, 21(5), 1999.

[63] L. Alexandre. 3D descriptors for object and category recognition: a comparative evaluation. *IROS Workshop on Color-Depth Camera Fusion in Robotics*, 2012.

[64] S. Ruel, T. Luu, M. Anctil, and S. Gagnon. Target Localization from 3D Data for On-Orbit Autonomous Rendezvous and Docking. *In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA*, 2008.

[65] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. *IEEE Conference*, 2010.

[66] O. Sorkine. Least-Squares Rigid Motion Using SVD. *Technical notes*, 120, 2009.

[67] P. Jasiobedzki, M. Greenspan, G. Roth, H. Ng, and N. Witcomb. Video-Based System for Satellite Proximity Operations. *7th ESA Workshop on Advanced Space*

Technologies for Robotics and Automation, 2002.

[68] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 1992.

[69] L. Liu, G. Zhao, and Y. Bo. Point Cloud Based Relative Pose Estimation of a Satellite in Close Range. *Sensors*, 16, 2016.

[70] S. Gemme, D. Gingras, A. Salerno, E. Dupuis, F. Pomerleau, and F. Michaud. Pose Refinement Using ICP Applied to 3D LIDAR Data for Exploration Rovers. *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2012.

[71] T. Birdal and S. Ilic. Point Pair Features Based Object Detection and Pose Estimation Revisited. *Proceedings of the 2015 International Conference on 3D Vision*, 2015.

[72] S. H. Or, W. S. Luk, K. H. Wong, and I. King. An Efficient Iterative Pose Estimation Algorithm. *Image Vis. Comput.*, 16:353362, 1988.

[73] F. Pomerlau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP Variants on Real World Data Sets. *Auton. Robot*, 34:133–148, 2013.

[74] J. Ventura. Autonomous Proximity Operations for Noncooperative Space Target. *Dissertation, Technical University of Munich*, 2016.

[75] S. I. Nishida, S. Kawamoto, Y. Okawa, F. Terui, and S. Kitamura. Space Debris Removal System using a Small Satellite. *Acta Astronautica*, 65:95–102, 2009.

[76] M. Greenspan, L. Shang, and P. Jasiobedzki. Efficient Tracking with the Bounded Hough Transform. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[77] T. W. Lim, P. F. Ramos, and M. C. O'Dowd. Edge Detection Using Point Cloud Data for Noncooperative Pose Estimation. *Journal of Spacecraft and Rockets*, 54, 2017.

[78] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi. A Review of Cooperative and Uncooperative Spacecraft Pose Determination Techniques for Close-proximity Operations. *Progress in Aerospace Sciences*, 2017.

[79] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-DOF model based tracking via object coordinate regression. *Computer Vision – ACCV 2014. Lecture Notes in Computer Science, Springer, Cham*, 9006, 2015.

[80] D. J. Tan, F. Tombari, S. Ilic, and N. Navab. A Versatile Learning-based 3D Temporal Tracker: Scalable, Robust, Online. *IEEE International Conference on Computer Vision*, 2015.

[81] S. Thrun. Simultaneous localization and mapping. *Robotics and cognitive approaches to spatial mapping*, 2007.

[82] A. Sonnenburg, M. Tkocz, and K. Janschek. EKF-SLAM based Approach for Spacecraft Rendezvous Navigation with Unknown Target Spacecraft. *IFAC Proceedings Volumes*, 2007.

[83] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual Tracking: An Experimental Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 2014.

[84] V. Lepetit and P. Fua. Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1), 2005).

[85] C. Harris and C. Stennett. Rapid a Video Rate Object Tracker. *In Proceedings of*

the British Machine Vision Conference, 1990.

[86] T. Drummond and R. Cipolla. Real-Time Visual Tracking of Complex Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 2002.

[87] A. Petit, E. Marchand, and K. Kanani. Tracking Complex Targets for Space Rendezvous and Debris Removal Applications. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[88] A. Cropp and P. Palmer. Pose Estimation and Relative Orbit Determination of Nearby Target Microsatellite using Passive Imagery. *In Proceedings of the 5th Cranfield Conference on Dynamics and Control of Systems and Structures in Space, Cambridge, UK*, 2002.

[89] S. D'Amico, M. Benn, and J. L. Jorgensen. Pose Estimation of an Uncooperative Spacecraft from Actual Space Imagery. *International Journal of Space Science and Engineering*, 2, 2014.

[90] D. Patel and S. Upadhyay. Optical Flow Measurement using Lucas Kanade Method. *International Journal of Computer Applications*, 61(10), 2013.

[91] S. Basu, I. Essa, and A. Pentland. Motion Regularization for Model-Based Head Tracking. *In Proceedings of the IEEE International Conference on Pattern Recognition*, 1996.

[92] M. Pressigout, E. Marchand, and E. Memin. Hybrid Tracking Approach using Optical Flow and Pose Estimation. *In Proceedings of 15th IEEE International Conference on Image Processing*, 2008.

[93] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10), 1998.

[94] K. Briechle and U. D. Hanebeck. Template Matching using Fast Normalized Cross Correlation. *in Procedeengs SPIE*, 4387, 2001.

[95] D. Comaniciu, V. Ramesh, and P. Meer. Real-time Tracking of Non-Rigid Objects using Mean Shift. *In Proceedings IEEE CVPR, Hilton Head Island, SC, USA*, 2000.

[96] N. A. Thacker, F. J. Aherne, and P. I. Rockett. The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data. *Kybernetika*, 34, 11 1997.

[97] Y. Yoon, A. Kosaka, and A. C. Kak. A New Kalman-Filter-Based Framework for Fast and Accurate Visual Tracking of Rigid Objects. *IEEE Transactions on Robotic*, 24(5), 2008.

[98] N. W. Oumer and G. Panin. Tracking and Pose Estimation of Non-cooperative Satellite for On-Orbit Servicing. *In Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2012.

[99] L. Zhengzhou, G. Fengzeng, C. Wenhao, S. Wanxing, L. Bing, and C. Bei. Particle Filter-Based Relative Rolling Estimation Algorithm for Non-cooperative Infrared Spacecraft. *Infrared Physics and Technology*, 78, 2016.

[100] S. Augenstein and S. M. Rockyz. Simultaneous Estimation of Target Pose and 3-D Shape using the FastSLAM Algorithm. *Proceedings of AIAA guidance, navigation, and control conference (GNC), Chicago, IL, USA*, 2009.

[101] A. Doucett, N. de Freitast, K. Murphy, and S. Russent. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. *Uncertanty in artificial intelligence*, 2000.

[102] D. Marimon, Y. Maret, Y. Abdeljaoued, and T. Ebrahimi. Particle Filter-Based

Camera Tracker Fusing Marker and Feature Point Cues. *Proceedings SPIE 6508, Visual Communications and Image Processing*, 2007.

[103] G. Panin, E. Roth, and A. Knoll. Robust Contour-Based Object Tracking Integrating Color and Edge Likelihoods. *VMV*, 2008.

[104] A. Yol, E. Marchand, F. Chaumette, K. Kanani, and T. Chabot. Vision-Based Navigation in Low Earth Orbit. *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2016.

[105] L. Pasqualetto Cassinis, R. Fonod, E. Gill, I. Ahrns, and J. I. Fernandez. CNN-Based Pose Estimation System for Close-Proximity Operations Around Uncooperative Spacecraft. *AIAA Scitech 2020 Forum*, 2020.

[106] S. Sharma, C. Beierle, and S. D'Amico. Pose Estimation for Non-Cooperative SpacecraftRendezvous Using Convolutional Neural Networks. *IEEE Aerospace Conference*, 2018.

[107] J. Padial, M. Hammond, S. Augenstein, and S. M. Rock. Tumbling Target Reconstruction and Pose Estimation through Fusion of Monocular Vision and Sparse-Pattern Range Data. *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2012.

[108] M. Liu, Y. Wang, H. Leung, and J. Yu. A Novel Feature-Level Data Fusion Method for Indoor Autonomous Localization. *Mathematical Problems in Engineering*, 2013.

[109] H. Cho, Y. W. Seo, B. V. K. V. Kumar, and R. Rajkumar. A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments. *2014 IEEE International Conference on Robotics and Automation*, 2014.

[110] M. B. Alatise and G. P. Hancke. Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter. *Sensors*, 2017.

[111] A. Tobergte, M. Pomarlan, and G. Hirzinger. Robust Multi Sensor Pose Estimation for Medical Applications. *Intelligent Robots and Systems*, 2009.

[112] T. Dieterle, F. Particke, L. Patino-Studencki, and J. Thielecke. Sensor Data Fusion of LIDAR with Stereo RGB-D Camera for Object Tracking. *IEEE Sensors*, 2017.

[113] T. Boge, T. Wimmer, O. Ma, and M. Zebenay. EPOS - a robotics based hardware-in-the-loop simulator for simulating satellite rvd operations. *10th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Sapporo, Japan*, 2010.

[114] H. Benninghoff, F. Rems, and E.-A. Risse. European Proximity Operations Simulator 2.0 (EPOS) - A Robotic-Based Rendezvous and Docking Simulator. *Journal of large-scale research facilities*, 3, 2017.

[115] M. Zebenay. Development of a Robotics-based Satellites Docking Simulator. *Dissertation, TU Delft*, 2014.

[116] T. Rupp, T. Boge, R. Kiehling, and F. Sellmaier. Flight Dynamics Challenges of the German On-Orbit Servicing Mission DEOS. *21st International Symposium on Space Flight Dynamics, Toulouse, Frankreich*, 2009.

[117] T. Wolf, D. Reintsema, B. Sommer, P. Rank, and J. Sommer. Mission DEOS Proofing the Capabilities of German's Space Robotic Technologies. *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2012.

[118] *SpaceTech GmbH.* Available at `http://www.spacetech-i.com/products/missions-satellites/deos`. Last accessed: February, 18, 2019.

[119] D. Piatti and F. Rinaudo. SR-4000 and CamCube3.0 time of flight (ToF) cameras: Tests and comparison. *Remote Sensing*, 4, 2012.

[120] B. Hagebeuker. Mehrdimensionale Objekterfassung mittels PMD-Sensorik. *Optik and Photonik*, (1), 2008.

[121] *3D Image Sensor REAL.* Avaliable at `https://www.infineon.com/cms/en/product/sensor/3d-image-sensor-real3/`. Last accessed: February, 18, 2019.

[122] D. Lefloch, R. Nair, F. Lenzen, H. Schäfer, L. Streeter, M. J. Cree, R. Koch, and A. Kolb. Technical Foundation and Calibration Methods for Time-of-Flight Cameras. Time-of-Flight and Depth Imaging. *Sensors, Algorithms, and Applications*, 2013.

[123] S. Fuchs and G. Hirzinger. Extrinsic and Depth Calibration of ToF-Cameras. *In Proceedings of 22nd IEEE Conference on Computer Vision Pattern Recognition*, 1-12, 2008.

[124] X. Luan. Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology. *Dissertation, University Siegen*, 2011.

[125] S. Theiss. Analysis of a Pulse-Based ToF Camera for Automotive Application. *Master Thesis, University of Siegen*, 2015.

[126] B. Langmann. Wide Area 2D/3D Imaging. Development, Analysis and Applications. *Springer Vieweg*, 2014.

[127] P. Fürsattel, S. Placht, M. Balda, C. Schaller, H. Hofmann, A. Maier, and C. Riess. A Comparative Error Analysis of Current Time-of-Flight Sensors. *IEEE Transactions on Computtional Imaging*, 2(1), 2016.

[128] M. Frank, M. Plaue, and F. Hamprect. Denoising of Continuous-Wave Time-Of-Flight Depth Images Using Confidence Measures. *Optical Engineering*, 48, 077003, 2009. 1.

[129] M. Frank, M. Plaue, H. Rapp, U. Koethe, B. Jaehne, and F. A. Hamprecht. Theoretical and Experimental Error Analysis of Continuous-Wave Time-Of-Flight Range Cameras. 48, 01 2009.

[130] F. Mufti and R. Mahony. Statistical Analysis of Measurement Processes for Time-of-Flight Cameras. *Proceedings SPIE 7447, Videometrics, Range Imaging, and Applications X*, 2009.

[131] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo. *Time-of-Flight Cameras and Microsoft Kinect.* Springer Science and Business Media, 2012.

[132] G. Marin, P. Zanuttigh1, and S. Mattoccia. Reliable Fusion of ToF and Stereo Depth Driven by Confdence Measures. *In Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands*, 2016.

[133] S. Hussmann, T. Ringbeck, and B. Hagebeuker. A Performance Review of 3D TOF Vision Systems in Comparison to Stereo Vision Systems. *Stereo Vision, Asim Bhatti (Ed.), InTech, DOI: 10.5772/5898.*, 2008.

[134] *GoldenEye 3D Flash LIDAR Space Camera.* Available at `http://www.advancedscientificconcepts.com/products/portable.html`. Last accessed: February, 18, 2019.

[135] F. Amzajerdian, M. Vanek, L. Petway, D. Pierrottet, G. Busch, and A. Bulyshev. Utilization of 3-D Imaging Flash Lidar Technology for Autonomous Safe Landing on Planetary Bodies. *In Proceedings SPIE 7608, Quantum Sensing and Nanophotonic*

Devices VII, 2010.

[136] *Camera Calibration with OpenCV.* Available at `http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html`. Last accessed: February, 18, 2019.

[137] J. Y. Bouguet. Available at `http://www.vision.caltech.edu/bouguetj/`. Last accessed: February, 18, 2019.

[138] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, M. Smisek, and K. Arbter. *DLR CalDe and DLR CalLab.* Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany.

[139] Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 2000.

[140] P. F. Sturm and S. J. Maybank. On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recogition*, 1999.

[141] R. Y. Tsai and R. K. Lenz. A New Technique for Fully Autonomous and Efficient 3D Robotics Hand-eye Calibration. *IEEE Transactions on Robotics and Automation*, 5(3), 1989.

[142] K. H. Strobl and G. Hirzinger. More Accurate Camera and Hand-Eye Calibrations with Unknown Grid Pattern Dimensions. *Proceedings of the IEEE International Conference on Robotics and Automation,Pasadena, California*, 2008.

[143] K. H. Strobl. A Flexible Approach to Close-Range 3-D Modeling. *Dissertation, Institute for Data Processing, Fakultät für Elektrotechnik und Informationstechnik, Technische Universität München, Munich, Germany*, 2014.

[144] M. Lindner. Calibration and Real-Time Processing of Time-of-Flight Range Data. *Dissertation, University of Siegen. Germany*, 2010.

[145] K. H. Strobl and G. Hirzinger. Optimal Hand-Eye Calibration. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, 2006.

[146] C. Papazov and D. Burschka. An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes. *Proceedings of the 10th Asian conference on Computer vision - Volume Part I*, 2010.

[147] *Calculation of the Normal in a Point.* Available at `http://docs.pointclouds.org/trunk/group__features.html`. Last accessed: February, 18, 2019.

[148] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. *Asian conference on computer vision*, 2012.

[149] F. Castanedo. A Review of Data Fusion Techniques. *The Scientific World Journal*, 2013.

[150] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. *Third International Conference on 3D Digital Imaging and Modeling*, 2001.

[151] J. A. Stipes, J. G. P. Cole, and J. Humphreys. 4D scan registration with the SR-3000 LIDAR. *IEEE International Conference on Robotics and Automation Pasadena, CA,USA*, 2008.

[152] G. Blais and M. D. Levine. Registering Multiview Range Data to Create 3-D Computer Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*

CA, USA, 17(8), 1995.

[153] B. K. P. Horn. Closed-form Solution of Absolute Orientation using Unit Quaternions. *J. Opt. Soc. Am. A*, 4(4), 1987.

[154] K. L. Low and A. Lastra. Reliable and Rapidly-Converging ICP Algorithm Using Multiresolution Smoothing. 2003.

[155] F. Pomerlau, F. Colas, and R. Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics, Now Publishers*, 4, 2015.

[156] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 1986.

[157] D. Antolovic. Review of the Hough Transform Method, With an Implementation of the Fast Hough Variant for Line Detection. *Department of Computer Science, Indiana University: Bloomington, IN, USA*, 2008.

[158] J. Matas, C. Galambos, and J. Kittler. Robust Detection of Lines using Progressive Probablistic Hough Transform. *Computer Vision and Image Understanding*, 78(1), 2000.

[159] *ImageProc.* Available at `http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/HoughTrans_lines_09.pdf`. Last accessed: February, 18, 2019.

[160] J. Grossman, M. Grossman, and R. Katz. The First Systems Of Weighted Differential And Integral Calculus. *Archimedes Foundation*, 1980.

[161] A. Schörgendorfer and W. Elmenreich. Extended Confidence-Weighted Averaging in Sensor Fusion. *Proceedings of the Junior Scientist Conference*, 2006.

[162] W. Elmenreich. Fusion of Continuous-Valued Sensor Measurements using Confidence-Weighted Averaging. *Journal of Vibration and Control*, 2007.

[163] H. Benninghoff, F. Rems, E.-A. Risse, B. Brunner, M. Stelzer, R. Krenn, M. Reiner, C. Stangl, and M. Gnat. End-to-end Simulation and Verification of GNC and Robotic Systems Considering both Space Segment and Ground Segment. *CEAS Space Journal*, 2018.

[164] F. Rems, E.-A. Risse, and H. Benninghoff. Rendezvous GNC-System for Autonomous Orbital Servicing of Uncooperative Targets. *Proceedings of the 10th International ESA Conference on Guidance, Navigation and Control Systems, Salzburg, Austria*, 2017.

[165] K. Klionovska and H. Benninghoff. Initial Pose Estimation using PMD Sensor during the Rendezvous Phase in On-Orbit Servicing Missions. *27th AAS/AIAA Space Flight Mechanics Meeting, Texas, USA*, 2017.

[166] R. Bridson. Fast Poisson Disk Sampling in Arbitrary Dimensions. *ACM SIGGRAPH 2007 Sketches*, 2007.

[167] K. Klionovska, J. Ventura, H. Benninghoff, and F. Huber. Close Range Tracking of an Uncooperative Target in a Sequence of Photonic Mixer Device (PMD) Images. *Robotics*, 7, 2018.

[168] K. Klionovska, H. Benninghoff, E.-A. Risse, and F. Huber. Experimental Analysis of Measurements Fusion for Pose Estimation Using PMD Sensor. *23nd Iberoamerican Congress, CIARP 2018, Madrid, Proceedings*, 2018.

[169] M. Ganet, I. Quinquis, J. Bourdon, and P. Delpy. ATV GNC during Rendezvous with ISS. *In Proceedings of the 5th Cranfield DCSSS Conference*, 2002.

[170] J. L. Hennessey and D. A. Patterson. Computer Organization and Design: The

Hardware/Software Interface. *Elsevier Applied Science*, 2007.

[171] *Performance of Computer Systems.* Available at `http://web.cse.ohio-state.edu/~teodorescu.1/download/teaching/cse675.au08/CSE675.02.Performance.pdf`. Last accessed: February, 18, 2019.

[172] C. Johann Treudler, H. Benninghoff, K. Borchers, B. Brunner, J. Cremer, M. Dumke, T. Grtner, K. Johann Hflinger, J. Langwald, D. Ldtke, T. Peng, E.-A. Risse, K. Schwenk, M. Stelzer, M. Ulmer, S. Vellas, and K. Westerdorff. Scosa - scalable on-board computing for space avionics. *69th International Astronautical Congress 2018, At Bremen, Germany*, 10 2018.

[173] K. Schwenk and F. Huber. Connected Component Labeling algorithm for very complex and high-resolution images on an FPGA platform. *Proc. SPIE 9646, High-Performance Computing in Remote Sensing V*, 2015.

[174] K. Schwenk. Component Labeling auf einem FPGA für eine zukünftige On-Board Bilddatenverarbeitung. *Dissertation*, 2015.