PhD-FSTM-2022-021
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 14/03/2022 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Philippe LUDIVIG
Born on 18th July 1987 in Luxembourg, (Luxembourg)

FIGURING OUT WHERE YOU ARE ON THE MOON: THE SELECTION AND VALIDATION OF DIFFERENT POSE-ESTIMATION TECHNIQUES FOR LUNAR SURFACE ROBOTICS.

Dissertation defence committee

Dr Holger Voos, dissertation supervisor
Professor, Université du Luxembourg

Dr Leon van der Torre. Vice Chair
Professor, Université du Luxembourg

Dr Miguel A. Olivares-Mendez, Chair
Professor, Université du Luxembourg

Dr Nico Hochgeschwender
Professor, Hochschule Bonn-Rhein-Sieg

Dr Julien-Alexandre Lamamy
Managing Director, ispace Europe

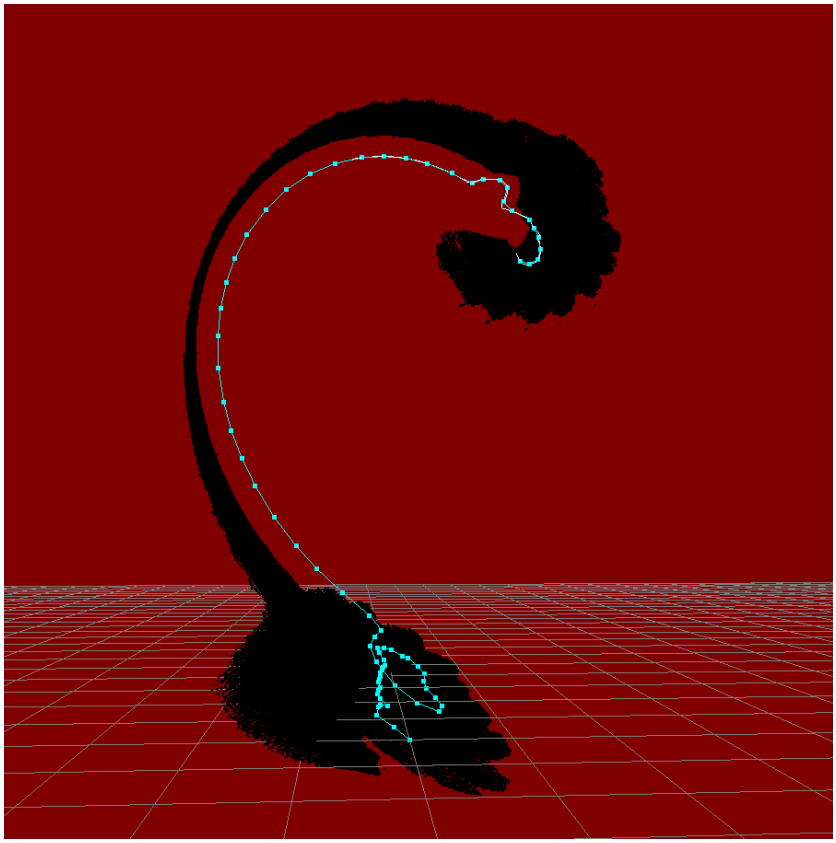

Figure 1: Failure is inevitable, part of the process and sometimes beautiful. (Localisation estimate of a rover driving in a straight line.)

Thesis advisor: Professor Holger Voos Philippe Ludivig

Figuring out where you are on the Moon: The selection and validation of different pose-estimation techniques for lunar surface robotics.

Abstract

With many New Space companies aiming to return to the surface of the Moon in the coming years, novel missions are being considered that can only last up to a single lunar day (14 earth days). In combination with the communication time delay (2.4 seconds), this leads to an increased incentive for enabling more autonomy to maximise operations for any mobile surface robot.

Since the primary component for this type of autonomy is good localisation estimates, both relative and absolute, are the main focus of this work. While this has been demonstrated before for terrestrial applications, the challenge here is to propose systems (software and hardware) which function within the limitations of what is physically possible on the lunar surface, and more importantly, what is financially viable for private companies, leading to more conservative mass and power requirements.

In terms of relative localisation, we address which sensor hardware should be considered for such applications. Additionally, we propose a novel software approach to improve localisation around lunar landers. The resulting localisation estimates can then be used to either support and accelerate the operator's decision making, or to allow for the rover to perform some of its driving independently.

On the absolute localisation side, we have turned towards machine learning to propose two novel methods to speed up the absolute localisation process through orbital and surface perspective imagery. Through this approach, we can more rapidly determine a rover's position in orbital imagery, which in turn, can be used to quickly commence surface operations after landing, as well as correct the localisation drift on longer traverses.

Because none of these methods could be validated on the Moon, we also considered the how to effectively configure testing environments to achieve the required confidence for private investors to support this technology. As such, two lunar analogue facilities were built, multiple virtual simulators were configured and an extensive field test was also conducted for the completion of this work.

Contents

Listing of figures

Nomenclature

COTS Commercial Of The Shelve

DEM Digital Elevation Model

DGPS Differential GPS

DoF Degrees of Freedom

FoV Field of View

FPGA Field-Programmable Gate Array

GNSS Global Navigation Satellite System

GPS Global Positioning System

ICP Iterative Closest Point

IMU Inertial Measurement Unit

LIDAR LIght Detection And Ranging

LOD Level Of Detail

LOLA Lunar Orbiter Laser Altimeter

LRO Lunar Reconnaissance Orbiter

m/px Metres per PiXel

MER Mars Exploration Rovers

NAC Narrow Angle Camera

NAVCAM Navigation camera

OF Optical Flow

PANCAM Panorama Camera

PPK Post Processed Kinematic

PSR Permanently Shadowed Regions

RGBD Red Green Blue Depth

ROS Robotics Operating System

RTK-GPS Real Time Kinematic - Global Positioning System

SFM Structure From Motion

SLAM Simultaneous Localisation And Mapping

ToF Time of Flight

TRL Technology Readiness Level

UAV Unmaned Aerial Vehicle

VIO Visual Inertial Odometry

VO Visual Odometry

WO Wheel Odometry

Acknowledgements

To my parents, who fostered my curiosity and always supported me regardless of what I wanted to study and explore. Villmools Merci Mam a Papp!!!! To Margarita Reyes, my better half, without you this would not have been possible. You stood beside me, come rain or shine, especially when is was a pain to be around while I was trying to finish that one paper. Thank you for reading trough [sic] the gobbledygook that is my thesis. To my siblings Anne and Michele, who were always there for me.

To the NASA Frontier Development Lab community, for jump-starting my research career by introducing me to a whole bunch of incredibly smart people, where I am still not quite sure if I am clever enough to fit in.
To Frank Soboczenski, for being involved in any Machine Learning project ever. Live long and prosper.
Sara Jennings, for being the best person at organising anything, thank you for making things happen. Also, Alpacas!
To Paul Wright, without our regular calls, the pandemic would have been a bleak place. Without you also efficiently picking a pose, one chapter of my thesis would also have been a bleak place.
To Mathieu Labbe, for sharing his know-how on SLAM development and patiently answering questions regarding Rtab-Map to the one guy who actually showed up in Sherbrooke.
To Ben Wu, Andrew Chung, Ross Potter and Timothy Seabrook, being on the only team aiming for a B at the NASA Frontier Development Lab. With a second mention to Ben Wu, for being crazy enough to attempt to repeat this feat with me.
To Will Facett, for introducing me to the truly amazing and spectacular world of particle physics. And Factorio.
To the whole MoonrAIders team, who embarked on a two-year roller coaster with me to build a fully autonomous multi-robot system, and patiently listened through all the meetings I *tried* to organise.
To Zahi Kakish, for doing his best to debug a state machine, while changing diapers halfway around the world!

To Daniel Medina, the only man to have successfully managed to make me get out of bed on a Sunday morning to explain the intricacies of D-GPS and IMU systems to sleepy Phil.
To Lukas Meyer, who builds controllers that just work, and datasets that are complete!
To Sanjeev Narayanaswamy, who responded to a lonely email, and subsequently introduced me to the world of orbital debris remediation. I'm still trying to catch up!
To Louis Burtz, for always being cheerful, and for being an honourable competitor during the NASA SRC2 competition.
To Richard Galvez, for being the guy who put ML into beer. Or was it beer into ML?
To Hriday Bavle, for bothering to read this, and for supporting multiple field tests!
To Claudio Cimarelli, for different insights on localisation systems!
To Michele Faragalli, for taking the time to discuss lunar analogue testing environments.
To Manuel Castillio-Lopez, for sharing complex control systems with me, crazy Manolo.
To Maciej Zurad, for taking on machine learning challenges, just for fun.
To Jose Luis Sanchez-Lopez, for his continued support throughout my research.
To Abigail Calzada Diaz, for all her knowledge on rocks, you rock!
To Karthik Venkataramani, who will just randomly pick up anything, as long as it's space-related!
To Dimitry Ivanov, the go-to guy when things get extraordinarily hot or extremely cold, for radiating joy, and talking about radioisotopes to absolutely anyone who will listen.
To Alessandro Ercolani, Marrio Merri and the whole team at ESOC OPS-GD, for putting me on the right track towards this research.
To the NVIDIA Corporation for supporting this work with the NVIDIA Applied Research Accelerator Program.
To the ESA ISEB program, for introducing me to researchers in the space sector from all over the world.
To ispace and Julien Lamamy for providing me with the best use case for this research.
To Prof. Miguel Olivares Mendez and the SpaceR group, who happily listened when I ordered to paint absolutely everything black.
To Prof. Holger Voos and the whole Automation Robotics Research Group at SnT, for supporting me throughout my research, and especially during the last couple of months of this work.
To the Luxembourg National Research Fund, without whom this research would not have been possible.
A big thank you to those I have not mentioned here.

Disclaimer

This research was funded through the AFR-PPP industrial fellowship research grant offered by the Luxembourg national National Research Fund (FNR). It aims to foster research collaborations with the private sector trough public private partnerships. The project resulted out of a proposal between myself, the University of Luxembourg and ispace Europe, with the latter being the host institution. (FNR grant number: 11824057).

1

Introduction

When we talk about pose estimation in the field of robotics, we generally are interested in the position and orientation of an object with regards to a reference frame. The term *pose* is used as a combined term for position and orientation ([179], p98). The term *estimation* is used because there is always some degree of uncertainty or error in the calculated pose. In the context of mobile robotics, we are interested specifically in the pose of our robot. Here, the

term *localisation* often refers to the pose of a robot with respect to its initial starting point ([196], p157). Other arbitrarily selected reference frames can also be considered as the centre of a robot's world, such as the centre of the lunar lander which deploys the rover. Global reference frames, such as the latitude and longitude coordinates of a global geodetic system can also be used.

Because the meaning of the following terms can sometimes be ambiguous, we define them as follows for the purpose of this thesis. We refer to the term *absolute localisation* when dealing with the pose of a robot in a global frame, such as a map or orbital imagery, *relative localisation* when discussion the pose of a robot relative to it's starting point and *pose estimation* when talking about the pose of an object in an image with regards to the camera reference frame.

Localisation for planetary rover missions has been a concern since the first Mars rover landed on the surface of Mars in 1997. All existing planetary rovers have had some internal localisation system to support surface operations. Even the most basic Lunokhod 1 and 2 which roamed the Moon back in the seventies, had some simple form of odometry/speedometer in order to support the ground operators during transverses [180]. While the first rovers had very basic localisation systems [143], we have seen constant improvement with new rovers and more complicated systems [91]. The current generation of these robots, such as Perseverance and Exomars, are complex systems developed by large teams of engineers from different space agencies. While expensive to develop and operate, some of these systems have managed to display high degree of reliability with more than 15 years of continuous opera-

tion [3]. Given that these are solitary rover missions, they also present a single point of failure where one problem on a rover system can lead to the end of a mission [213]. Nowadays, we are starting to see much smaller exploration systems which are instead developed by private companies. These smaller robotic systems rely on Commercial Off The Shelf (COTS) components [9] to save costs. This approach is spearheaded by so-called "New Space" companies which aim to bring innovative and affordable solution to space [147]. The small size of the proposed rovers also makes multi-robot systems a possibility, offering up new mission concepts with heterogeneous robot teams [174]. With localisation being a crucial element to rover operations, new approaches are being developed to accurately estimate a rover's pose during traverses [56] on these smaller systems.

In addition to the decreasing rover size, a new class of rover missions is being considered, where extremely short missions of less than a lunar day* [57, 124] are acceptable for commercial operations on the lunar surface to achieve the same primary mission goal, but at a fraction of the cost. The primary reason for this short life-time mission is the extreme temperature difference between the lunar day and the lunar night, where the temperature drops from +120 degrees Celsius during the day to -205 degrees Celsius during the lunar night [154]. In combination with the lunar night lasting 14 earth-days, and the fact that most current commercial lunar rovers solely rely on solar energy to power their systems, night-time survival is a costly feature[85]. Due to the limited duration, rover operations should commence almost immediately after touchdown to maximise the time for exploration. Additionally, an increased level of autonomy of the rovers is needed to be less dependent on teleoperation with

*From sunrise to sunset, this equates to approximately 14 earth days.

its large time delay.

This poses several challenges that we aim to address, including absolute localisation immediately after touchdown, improved relative localisation to enable semi- and fully-autonomous operations, as well as facilities and environments to validate any of these proposed methods.

1.1 Problem Statement

A. Testing and Validation

Problem: The first issue we encountered when we started working on lunar surface robotics was the lack of available testing environments to validate our results. Given the added risk involved with any autonomy reliant on new localisation systems, the validation of these systems also needs to be addressed. If a risk cannot be mitigated, some level of risk may be acceptable, but it should always be properly understood. For terrestrial localisation applications, like autonomous cars, this translates to collecting thousands of hours of data from localisation systems operating in real scenarios in order to find edge cases which the engineers have not previously considered [76]. Based on such datasets, the reliability of a system can then be estimated. For lunar surface applications however, this is not practical because there is no access to the actual operational environment on the Moon before each mission. Furthermore, we only have limited knowledge of the surface environment which our rovers will encounter. While high resolution orbital imagery covering almost all parts of the Moon are available, this imagery is only accurate to about 0.5 metres per pixel [164] for nearly all locations. Knowledge of anything smaller than that, such as smaller rocks or the composition of the surface material, are simply estimates based on other landing sites or other available remote sensing

data. As a consequence, testing requires a wider range of scenarios to be covered to account for the environment uncertainty. On its own, this is already a complex challenge and many space agencies worldwide are currently trying to address this issue. For commercial operations however, the question of testing and validation is more nuanced. To them, reliability is important, but so is the cost of testing. This issue is further complicated by the fact that there are currently only a handful of relevant lunar testing facilities available in the world. We will therefore, address how new facilities can be built, specifically with testing of localisation systems in mind. While taking into consideration the concerns of New Space companies, we will also examine which environments satisfy the trade-off between acceptable accuracy, and cost-effective testing strategies.

Brief State of the art: For the purpose of this work, we have divided the testing environments into three different categories: virtual simulator, indoor testing facilities and outdoor field tests.

There are numerous virtual simulators used in robotics research, with the most commonly used ones being Gazebo, Vrep and Webots (based on [28]). However, only a handful of simulators have been configured [7] or built [43] specifically to mimic the lunar or martian environment. One major disadvantage is that some of these solutions are not publicly available or expensive to procure. Alternatively, simpler lunar environments are available, which are easy to interface with robotic systems such as the Gazebo environment in the NASA Space Robotics Challenge II* as well as more realistic simulators based on computer games engines [177, 40]. Lastly, we can also look towards employing 3D rendering tools such as Blender to generate images offline. Notably, Blender has also been extended to generated more accurate

*http://www.spaceroboticschallenge.com/

shading of the lunar surface [5].

In terms of physical lab facilities, their construction depends heavily on the intended use case. These facilities normally have to focus on a number of priorities, such as the use of accurate soil properties with lunar regolith simulant [111, 51], accurate environmental properties with simulated pressure and temperature values [45], lighting conditions [51], complex terrain [29] or simply size [187, 89, 50].

For the final validation, field tests should always be considered for autonomous systems as indoor facilities are often limited in variation due to their small size. Often, the more practical solutions are nearby quarries [212] or sand beaches [80], which are excellent choices because of their accessibility. Ideally, volcanic locations such as mount Etna should also be used [202] as they are highly suitable. The fresh basalt rock that covers the landscape mimics the landscape of the Moon accurately because the fine sharp particles are very similar to the lunar regolith. Alternatively, desert locations, which are often used for Mars analogues, present good solutions as they provide large unstructured environments. Some favourable desert locations can be found for example in Morocco [142, 108], the Atacama desert [142] or Devon Island [61].

That leads to the first research objective of this thesis: How can we best configure an array of testing environments which can help use validate localisation systems without going to the Moon.

B. Relative localisation

Problem: The next issue we need to address is the time delay of at least 2.4 seconds for round trip communications between the Earth and the Moon [32, 140]. This delay considerably

slows down any remote operations because the controller needs more time to intervene in case a problem arises. Often, this leads to short distance driving commands or very slow driving speeds. As the mission lifetime is limited, this leads to a strong incentive to increase the autonomy of surface exploration systems, even if this comes at the expense of some additional risk. In this case, we consider autonomous systems which can reduce the wait times while a robot is waiting for human input. This could be as simple as an operator providing waypoints [132] rather than direct wheel commands. On the other end of the spectrum, fully autonomous planning, navigation and obstacle avoidance would provide full autonomy. Between these two solutions, there is a sizeable difference in risk. Regardless of the complexity, both solutions rely on a robust relative localisation system which can inform the robot where it has started and how far it has since moved. This is the basis of any complex autonomous system such as mapping, obstacle avoidance and path planning. For New Space missions, the question which arises from this issue is how to select a localisation system for small-sized rovers while relying on Commercial Of The Shelf (COTS) components. This question deals with issues which are both software and hardware related. On the hardware side, we need to evaluate which new sensors are available and suitable for this application whereas on the software side, we can leverage more computing power of newer COTS components.

Brief State of the art: For relative localisation, we look at existing solutions employed by current planetary rovers. First promising localisation results were shown by Spirit and Opportunity, where visual odometry was used to correct short stretches of way-point driving [132]. To date, all planetary rovers still rely on stereo camera systems as primary localisation sensor (Yutu[121] and Yutu-2[131], Exomars[199], MSL Curiosity[91], Perseverance [133]). To achieve localisation out of these stereo camera systems, the rovers rely on what is

known as *Stereo Visual Odometry*, where the difference between the left and the right cameras is used to estimate depth, while the difference between two consecutive images is used to estimate travel distance. The resulting estimate is also often used with other available information. In relation to hardware, the question which emerges is why these robots do not use LIDAR, as this technology has been proven in terrestrial rover experiments[67]. The main reason is due to the lack of availability of a small reliable LIDAR sensors that are space qualified. On the other hand, digital cameras have a longstanding history of being used in space. As an additional option, localisation can also be achieved with a monocular camera system. However, stereo systems provide more accurate and thus more reliable odometry. On the software side, the next big leap for camera based localisation systems is to run Simultaneous Localisation And Mapping (SLAM) on-board planetary rovers. Some missions have performed additional processing similar to SLAM in post processing on the ground (Yutu2[131], MER [38]). However, this technology has not yet found its way onto the rovers themselves because it requires considerably more processing power, which is simply not available on today's radiation hardened computing platforms. In terrestrial experiments, they have however shown promising results: [174].

That leads to the second objective of this thesis, which focuses on evaluating different localisation systems in order to recommend a relative localisation system for lunar surface robots.

C. Absolute localisation

Problem: The first problem is the uncertainty of the actual touchdown location within the landing ellipse of the lunar lander. For missions aiming to perform a traverse to a specific anomaly detected from space [27], or missions with a detailed mission plan based on orbital

data [74], this requires an absolute localisation system with the ability to accurately estimate the position of the landing site in relation to a global lunar reference frame [130]. Given that most lunar missions have a limited lifetime, speed is the essential component here to maximise exploration time, thereby providing increased opportunity for science return.

Brief State of the art: For terrestrially applications, the solution to this problem is to rely on Global navigation satellite systems (GNSS)[142], but these will not be available around the Moon for at least another decade[156]. The next best solution is to search for the robot in an orbital image taken after landing [121]. This relies on the availability of a high resolution orbiter to fly over the expected location in favourable lighting conditions, which can take weeks [34]. On Mars, this has also been solved through a time-consuming process of triangulating a rover's location through horizon features [25], before manually matching the ground around a rover to orbital imagery. In recent findings, an example from a lunar mission has successfully shown that this process can be accelerated [206]. For this approach, one needs to rely on lander descent images, which may not be available when flying a rover on a commercial lander. However, this method would also not be helpful on traverses where we want to sporadically update the rover's position. Instead, we can look towards terrestrial methods, where machine learning has been used successfully to match surface perspective and orbital imagery [197, 13]

That leads to a third objective of the thesis, which deals with the question of how to accelerate absolute localisation immediately after landing.

1.2 Objectives

The *Main objective* is to increase autonomy of small lunar rovers through improved localisation. This will be achieved through the following *sub-objectives*:

A. Configure an array of testing environments to validate lunar localisation systems on Earth.

B. Evaluate different localisation systems in order to recommend a relative localisation system for the lunar surface covering both hardware and software.

C. Build an absolute localisation system to speed up rover operations immediately after landing.

1.3 Thesis Contributions

The work in this thesis will contribute to advancing the autonomy of small lunar exploration rovers while specifically considering commercial New Space applications. Since one primary technology to enable autonomous mobile robots is localisation, we will therefore focus on localisation, as well as the testing of localisation systems.

- When dealing with computer vision systems, guaranteeing a flawless implementation is impossible, even for terrestrial applications. For lunar rovers, this is even more challenging as we cannot test on location. While there are specialised facilities to replicate the designated environment on the Moon, only a few of them exist, and they are expensive to build. On the other hand, simulation environments are accessible and easy

to set up, but have low degree of fidelity. In chapter 2, we address the different solutions which exists in order to make a qualitative assessment on how to build, configure and plan a testing campaign. This is achieved especially by considering cost, in order to keeping the proposed solutions accessible to New Space actors. This includes the following contributions:

  - We provide examples of existing facilities and testing environments to provide a baseline for the configuration of any new environments.
  - Construction of two indoor lunar lab facilities.
  - Configuration of virtual simulation testing environments that can interface with existing robotic systems, such as ROS (Robotics Operating System).
  - Organisation and performance of a field test in an outdoor lunar analogue testing environment.

- When developing a relative localisation system for a lunar rover, many factors must be taken into account. In chapter 3, we will explore the different hardware and software solutions which are available for this purpose. We will provide an overview of possible solutions, and address which solutions are preferable for small rover applications. With respect to hardware, this addresses which sensors to use and how to configure them. For software options, this concerns examining existing terrestrial localisation systems and extending them for planetary rover missions.

  - Evaluation of the state of the art of relative localisation systems.
  - Qualitative assessment of different localisation sensors, and their suitability for small rovers operating in the harsh environment of the lunar surface.

  - Stereo camera considerations for the lunar surface robotics
  - Proposal of a localisation system that makes use of the continuous close proximity of a lunar lander in order to improve a rover's localisation estimate.

- In chapter 4 we look at different approaches available for performing absolute localisation on other planetary bodies with regards to a global reference frame. We then assess how machine learning can be used to improve the state of the art by evaluating two different methods to compare orbital imagery and surface perspective imagery. Here, we provide the following contributions:
  - Evaluation of the state of the art of absolute localisation systems.
  - Generation of a virtual lunar dataset composed of orbital and surface perspective imagery to perform absolute localisation benchmarking
  - Training and evaluation of a machine learning approach to perform absolute localisation between orbital imagery and surface perspective imagery
  - Evaluation of a second absolute localisation method, address the shortcomings of the initial approach to include horizon features in the matching process.

1.4 Scope

To focus on the development of more flexible New Space approaches, the following limitations have been observed:

Environment: Planetary probes have been sent to the surface of Moon [122], Mars [143], Venus, Titan [112] and a number of asteroids and comets, with many more missions

being planned. For the main body of work in this thesis, we will focus on the specific problems posed by operating rovers on the surface of the Moon. We will also address some of the challenges of operating on the surface of Mars. While this is not the main aim of this thesis, it is important to mention these rover missions, because the majority of planetary rovers to date have been sent to the red planet.

Processing Hardware: We are not making use of radiation hardened or even flight-proven space hardware to validate our results due to the added development complexity. Instead, we are using Commercial Off The Shelf (COTS) components which are within reasonable assumptions of power and mass budget. While this technology has not yet been proven in space, we assume that similar hardware will be flown to the Moon and Mars in the near future, especially considering the much shorter lifetime of upcoming lunar missions. A good example of this is the cellphone grade processing unit flown on the Mars helicopter Ingenuity [12].

Bench-marking limitations Concerning testing and validation of localisation systems, we purely evaluate the localisation performance of systems, for which we benchmark the estimated pose of a system and compare it to the pose provided by our ground truth. We do not address any hardware requirements, such as structural, thermal or any other engineering challenges as they are only indirectly related to the localisation quality. Additionally, we assume software validation and verification to be outside of the scope of this thesis because this step is only required after a valid localisation system has been selected to fly.

Risk: Some of the presented work applies a level of risk-taking currently not being applied

by space agencies for planetary exploration. We do however, expect such risks to be acceptable in the near future for multiple reasons. Commercial operators are joining and even financing space exploration of the Moon, and are eager to be successful on a smaller budget [9]. At the same time, they are not required to justify potential failures to the general public which can lead to a very conservative approach to space exploration. Secondly, multi-robot missions will no longer result in the end of mission when a single robot fails, making smaller less reliable rovers an acceptable compromise. Lastly, the mission lifetime which is being considered, is less than 10 earth-days on the lunar surface. This is due to the fact that some commercial robots are not slated to survive the lunar night with temperatures of -205 degrees Celsius. This short lifespan provides added incentive to take additional risk, especially towards the end of a mission. This is in stark contrast to some Mars exploration missions such as Opportunity, lasting more than a decade and where there is very little incentive to take major risks[3].

Mission concepts: When developing a localisation system, the intended use case is just as important as the available payload mass or size. Especially for resource constrained systems, we have to ensure that the final solution is tested in similar scenarios to those expected during normal operations. In this work, we are considering a set of possible use cases towards which we benchmark our pose estimation. For relative localisation, we will investigate long traverses (> 100 metres), ground mapping (zigzag shapes that cover a maximum of the surface area), and collaborative robot systems (multiple robots operating around a landing structure). For absolute localisation, we are considering localisation immediately after landing, as well as missions requiring single absolute position updates when taking scientific measurements. When dealing with localisation

systems, the intended use is just as important as the available payload mass or size.

1.5 Interdisciplinary Aspects

Planetary robotics has its roots deeply embedded in terrestrial applications as the majority of the work presented here can be applied on earth. But it is difficult to compete with autonomous cars with their highly accurate LIDAR systems, which are not available for planetary applications. Many applications however, need to keep costs low and therefore still make use of similar vision-based localisation systems found on today's Mars rovers. Even more interestingly, unmanned aerial vehicles (UAV), have very similar limitation than what is observed in space. As UAV's also suffer from mass constraints, their payload is also guided by mass, power suitable compute units. This is very similar to what we would expect for space applications, even if the radiation issues lead to more stringent restrictions in terms of available compute power.

In addition to possible applications, the thesis is also drawing on the expertise of a number of different fields. The biggest contribution, besides the robotics component, comes from machine learning. In recent years, it has found many more applications in robotics, especially when dealing with complex vision systems. In order to accurately benchmark such systems, a good understanding of computer graphics as well as geographic information systems was also necessary. A basic understanding of the engineering challenges in the target environment should also not be discarded. There are many interesting methods that can be explored, but at the end of the day, solutions will only really be successful, if they are based on reasonable

engineering assumptions. This is especially important in commercial environments where this can also mean that a technically feasible solution might not be economically viable. For New Space, 'good enough' can sometimes be more interesting than the best, most accurate solution.

1.6 Methodology

For the presented body of work and its respective topics, the thesis follows the following high-level approach to arrive at our conclusion, moving from the initial field to an identified problem, to a proposed solution with its evaluation. As a first step, we provide the reader with a detailed introduction of why and how the chapter topic is relevant to pose estimation on the lunar surface. For each designated topic, we have evaluated and tested different existing solutions, to gain an understanding of the underlying issues as well as the proposed solutions that deal with them. After this initial step to familiarise ourselves with the topic, it is essential to cover the state of the art in detail to be able to build on top of existing work. Aside from examining work from planetary rover missions, terrestrial applications must also be evaluated. This is integral to bringing broader, more compelling and original ideas to this research, as the body of work from terrestrial applications is much wider. After gaining an understanding of the problems in the field, we identify a gap in the state of the art which can be addressed in order to support future generations of lunar exploration rovers. Once this gap has been identified, development of different solutions begins, which is the most time-consuming aspect to this work. This is where new methods are being implemented and tested in order to bring novel solutions to the proposed problem. The most critical part

when implementing new approaches is a strong evaluation of the resulting solutions. Ideally, this includes a quantitative comparison against ground truth measurements, but also testing in representative environments. For this reason, a significant part of this body of work has been devoted to recreating the characteristic environment of the lunar surface with different testing approaches.

1.7 Thesis Outline

The primary topics covered in this thesis are **relative localisation**, **absolute localisation** and **testing & validation of localisation systems**. As each of these topics constitutes a single body of work, the thesis is separated into three separate chapters, each chapter covering their own state of the art, before elaborating on their the different experiments and the resulting outcomes:

Chapter 1 provides an overview of the work covered in this thesis, including objectives and methodology.

In *Chapter 2*, we present the different testing methods and facilities that are needed to evaluate localisation techniques with a certain degree of reliability. This is subdivided into the following major sections.

- In the first section, we cover virtual simulation environments. This includes evaluating the state of the art as well as the configuration and use of four different virtual environments.

- In the second section, we introduce existing indoor lab facilities, which mimic the surface of the Moon. Based on these facilities, we then explain how two additional such facilities were constructed as part of this research.

- In the third section, we describe the use of outdoor analogue field tests as final data point in a series of test to validate a localisation for the lunar surface.

In *Chapter 3*, we consider different relative localisation systems and sensors.

- In the first section, we address the state of the art, considering different types of odometry as well as simultaneous localisation and mapping (SLAM).

- In the second section, we investigate different sensor technologies used for localisation, and we give a qualitative assessment on how they can be used for planetary robotics.

- In the third section, we evaluate which factors need to be considered when building a stereo localisation systems for the lunar surface.

- In the fourth section, we apply machine learning based pose estimation technique to improve localisation systems of surface rovers in proximity of a lunar lander.

In *Chapter 4*, we look at absolute localisation system.

- In the first section, we introduce different absolute localisation systems that have been used for space applications.

- In the second section, we present a novel machine learning approach to localise by comparing surface perspective imagery and orbital imagery.

- In the third section, we extend previous methods to localise by comparing surface perspective imagery with reprojected orbital imagery.

Chapter 5 concludes the thesis and provides a prospective outlook to where the localisation domain is heading.

2

Lunar Testing and Validation

2.1 Background

In order to reliably compare different localisation techniques, we need to consider how to evaluate their qualities. This is especially problematic for our case due to the inability to test in the actual operational environment. In this chapter, we therefore discuss the best methods to evaluate and validate vision-based localisation systems for planetary surface robots. The focus on vision-based systems stems from the considerations shown in chapter 3, where we establish that camera-based localisation is currently the most likely candidate for localisation in our use case. As presented in chapter 1, our work is geared towards the private industry, where cost-effective testing methods are of particular interest. This chapter serves as a guide on how to best evaluate the accuracy of localisation systems by advising the reader of the strengths and weaknesses within each approach. This chapter is divided into three separate sections where we will be detailing testing in virtual simulation environments, indoor testing facilities and outdoor testing facilities. For each of these sections, we will first cover the state of the art and the facilities which were constructed or configured for this research before discussing the strengths and weaknesses of each environment.

Some of this work in this chapter has been previously published and presented in [128], [127] and [129].

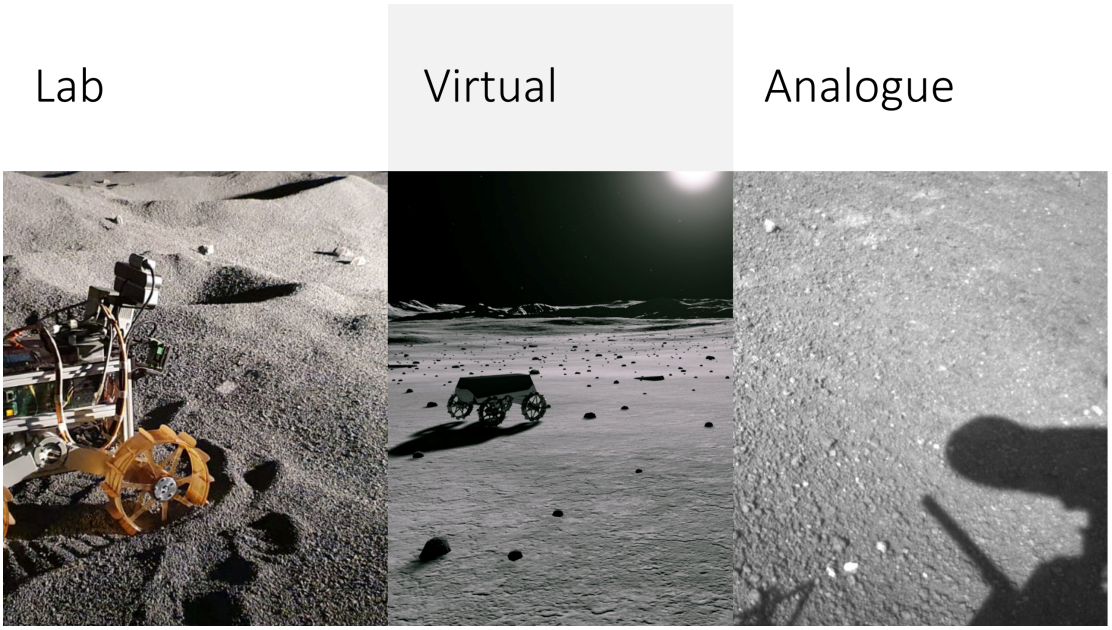

Figure 2.1: Different types of testing environments for a lunar surface rover

2.2 Virtual Simulation Environments

The first environment to address is that of digital simulations as they require no actual rover hardware or physical testing grounds. As such, they should be used right from the start, when changes to the design and the requirements are still relatively inexpensive.

Virtual environments are commonly used in robotics. As shown in figure2.3 from [28], there are numerous simulators, however, only a handful of them are regularly mentioned in research papers. The most commonly mentioned simulators are Gazebo, CoppeliaSim (previously known as Vrep) and Webots. While MuJoCo and PyBullet are also popular, we will avoid these as they primarily focus on accurate physics rather than camera images and are thus not relevant to this work

Figure 2.2: Virtual lunar environment in Unreal Engine

In addition to dedicated simulators, 3D computer graphics tools such as Blender[*], Maya[†] or Houdini[‡] also need to be mentioned, as not all applications require direct a connection with a robotics system, as long as they can create the required imagery and metadata required for experiments.

GameEngine based solutions such as Airsim[§] and ROS-Sharp[¶] also need to be mentioned. They are both built on top of current computer games engine Unreal Engine and Unity. While not being full robotics simulation environments, they are able to produce real-time imagery and metadata, often providing a more realistic rendering quality, rivalling offline rendering tools. They sit in the middle of complete robotics simulators and offline rendering tools.

[*] https://www.blender.org/
[†] https://www.autodesk.com/
[‡] https://www.sidefx.com/
[§] https://microsoft.github.io/AirSim/
[¶] https://github.com/siemens/ros-sharp

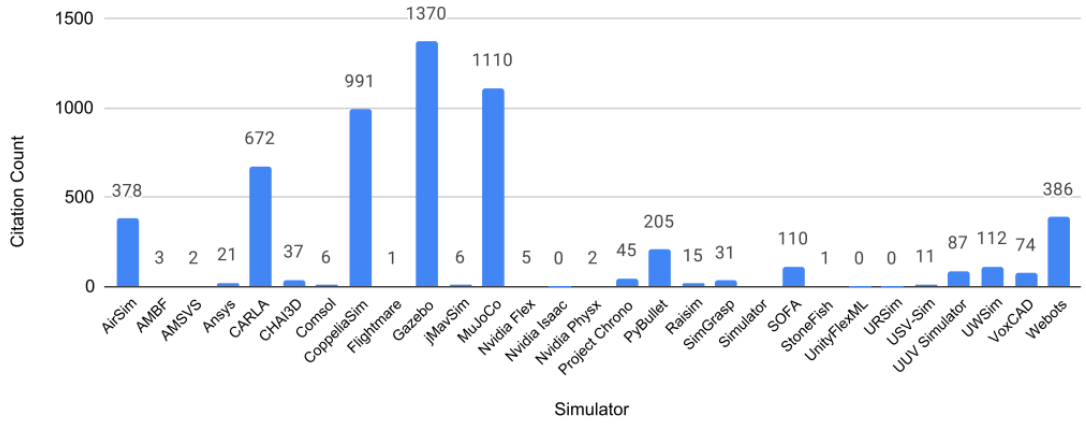

Figure 2.3: Citation count from 2016 to 2020 for reviewed simulators. Citations were gathered from Google Scholar using either one or more of a simulators' research paper, reference manual or other citation type and then filtered for robotics keyword. (graph taken from [28])

2.2.1 Existing Lunar Simulators

Only a handful of simulators have been specifically built or modified for planetary surface operation testing, such as Pangu [43, 136], Gazebo [7, 23], IRIS [4], and 3Drov [68]. In this section, we will examine Gazebo and Pangu in more detail.

2.2.1.1 RP Simulator - Gazebo [7]

Gazebo is one of the primary simulators used in the research community, because it is an open-source project closely tied to ROS. It provides a range of localisation sensors such as cameras, wheel encoders, inertial measurement units and LIDAR's, for which it also has noise models to simulate more realistic sensor data. However, it is lacking capabilities to render complex environment and lighting conditions. As a result, the camera images are somewhat simpler. It is also limited in the scale of it's surface geometry as it provides no Level Of Detail (LOD) system that can change the number of polygons of distant objects automatically.

The cited references [7] and [23] show a modified version of this simulator providing more realistic rendering of the lunar environment. Most importantly, it introduces a Level Of Detail (LOD) system which can automatically manage the size and resolution of a subdivided surface terrain model based on the distance from a robot. This is essential for dealing with large scale high resolution surface geometry and textures. The presented work describes methods for adding synthetic centimetre level resolution to the best Digital Elevation Models available for the lunar surface with the help of rock and crater distributions estimates. The work also implements an approximation to provide real-time Hapke physically-based rendering (based on the Hapke Bidirectional Reflectance Distribution Function (BRDF) [75]). This is a photometric model often used to accurately render planetary surfaces. To improve shadows, Gazebo's default shadow maps are replaced with light-space perspective maps, which reduce their aliasing artefacts. Lastly, it also adds rover tracks in the soft soil through the help of an existing Gazebo plugin. Overall, the simulation is a very complete package, and probably the closest, most realistic Lunar simulator currently available, which can provide real-time feedback and a good integration with existing robotic systems.

However, the upgrades presented in this work still are limited by the use of Gazebo and the Org3D render engine. The updated shadow maps still present artefacts. The self-shadowing of the surface geometry is baked into the surface textures, and is only updated once every hour to account for the slow movement of the sun. The physics model of the rover is also approximated to reduce the computational cost. Most importantly the simulator is not publicly available, which makes some of the claims hard to verify. The features described in this section are therefore solely based on available publications and presentation given by members of the NASA AMES research centre [7, 23].

2.2.1.2 Pangu [43]

Pangu*, also known as Planet and Asteroid Natural scene Generation Utility, is a tool developed by the European Space Agency and the University of Dundee to produce synthetic environments of entire planets or asteroids, as well as for limited high resolution surface sections. In order to achieve this, it can generate environments from scratch, or by building on existing data, such as Digital Elevation Models (DEM). Craters and rocks can be generated based on statistical models in order to produce realistic environments [43, 134]. It can simulate a range of different sensors, such as cameras, LIDAR or RADAR. In [152], it has been used to simulate Guidance and Navigation (GNC) operations for planetary lander and rendezvous operations. It was also utilized to simulate surface environments, as described in [136] to generate data for the validation of the navigation algorithms of ExoMars. Similarly, [162] mentions how Pangu data was applied in combination with other approaches to validate a monocular SLAM system. One notable limitation that was mentioned in this paper is that boulder textures produced by Pangu can be repetitive, leading to *identical visual features being observed at different locations*. While we considered testing this simulator, it was not used due to the software licensing costs involved.

2.2.2 General Simulator Considerations

When building a simulation environment, several considerations and design choices must be dealt with. The following is a list of parameters which were considered in the simulation environments that were tested throughout this work (Gazebo, Unreal Engine/Airsim, Blender, ISAAC Sim).

*https://pangu.software/

2.2.2.1 Real-time Vs Offline Rendering

If data is not required in real-time, there are more options for data generation available. We see this especially in machine-learning applications, where recorded datasets are needed, rather than real-time interaction with a simulation environment. From a simulation perspective, this leaves more processing time for higher quality rendering, or the use of more suitable tools such as Blender or Maya with render engines such as Cycles, Vray or Renderman. A good example can be seen in [5], where the limits of physically accurate shading of the lunar surface are being explored, but at costly rendering times of $\sim$ 40 seconds per frame.

On the other hand, some applications require real-time feedback to test a complete system. Autonomous navigation systems for example, generate control commands based on the incoming sensor data. As the new rover positions are not known beforehand, sensor data has to be generated in real-time.

Our localisation problem is situated somewhere in the middle. To benchmark a localisation system, real-time feedback is not required if we are only interested in the pure localisation accuracy of an existing traverse. The downside of this approach is that researchers often only rely on a small number of datasets, while edge cases are more likely to appear as additional testing is completed. For rover systems with autonomy, this is also problematic, because the type of driving can influence the localisation accuracy, which in turn, impacts the driving commands of the autonomous system. As such, when testing a navigation with a full autonomy, real-time simulators are preferable.

2.2.2.2 Dedicated Robotics Simulator vs Game Engine

In recent years, robotics simulators such as AirSim[177] or Carla[40], which are based on computer games engines, have gained more popularity. Other notable examples are also machine learning approaches which have been trained directly on an existing computer game as shown in [163]. The discussion here is mainly choosing between more realistic graphics, and a higher overhead on integration with other robotics systems. In recent years, these types of simulation environments have shown up more frequently in machine learning papers, where no direct interface is needed, as long as the required data can be extracted.

2.2.2.3 Dynamics - Physics

To generate camera images, physics might not seem of great importance, but when benchmarking localisation systems, the type of locomotion a vehicle presents may have a significant impact on the quality of the localisation. Shocks caused by rocks add noise to the IMU measurements, and potentially cause motion blur in the camera systems. All of the tested simulators provide a physics engine, which is used to estimate a rovers movement. As this is calculated in real-time, the physics accuracy can face some limitations. Often, simulators rely on existing physics engines such as ODE*, Bullet†, Simbody‡, or DART§ to produce these calculations.

If physics needs to be estimated at a higher accuracy, discreet Element Method analysis can more accurately estimate the wheel-soil-interactions as shown in [28]. These methods, while

*http://opende.sourceforge.net/
†https://pybullet.org
‡https://simtk.org/projects/simbody/
§http://dartsim.github.io/

highly precise, are however slow to compute and put additional limitations on our simulation environment, which is why we have decided to discard them.

Real motion data from a motion capture system is also another option for providing accurate motion. In this case, a 3D mesh of the surface should be captured to match the virtual environment with the motion. While this method limits the surface area to the lab size where the data is captured, it does allow for virtual environments that extend over the boundary of the lab environment, while providing real dynamics. We did not further investigate this approach, as significant vibrations were not observed during testing in our lab environment. However, during some outdoor testing on harder surfaces, vibrations from the mobility were more noticeable and impacted the quality of the camera and IMU data. With Mars missions, this could also be a more predominant issue, as Mars does not possess a similar uniform dust coverage as on the Moon.

2.2.2.4 Scanline Vs Raytracing / Path-Tracing

Since the beginning of computer graphics, abstractions have been used to generate images accurately. Simulating photons bouncing around the scene is a computationally expensive process, which is why most render engines use a form of scanline rendering, where polygons are sorted by the proximity to the camera, and then shaded based on the orientation towards the camera and the available light sources. This greatly reduces the complexity of the problem and is ideal for applications that require real-time feedback, as seen in computer games. Ray-tracing works by firing rays into the scene through each pixel of the image. The colour of each pixel is then determined by the average of colour of all the photons that are being fired through this pixel, as each photon can be reflected into slightly different directions. This

essentially reverses the way photons travel, but is also much more efficient as most photons emitted by a light-source never hit the camera. This process was generally used for single images or animations that can be rendered offline. Only in recent years, has this approach made its way into real-time rendering of computer games. To achieve this, they often use hybrid-methods where different rendering passes are rendered with different methods, in order to find the best compromise between realism and rendering speed.

2.2.2.5 Integration With Other Systems

When selecting a simulation environment, one should consider how to best interface it with external systems. Many simulators offer interfaces to tools such as Robotics Operating System (ROS) as shown in [7], or Matlab Simulink as shown in [134]. Here we differentiate between open-loop systems which are only able to export data and close-loop systems where the simulator reacts to commands coming from outside. Such systems even allow for testing with flight hardware, where the sensor data is generated by a simulator, but the processing is performed on the actual rover hardware. While it is always possible to build custom interfaces, the added work should however not be underestimated, as they often add a considerable amount of overhead in terms of development and maintenance.

2.2.2.6 Post-Processing - Improved Realism

As shown in figure 2.4 from [163] Generative Adversarial Networks (GANs) have successfully been used to improve the realism of simulator images based on a training dataset of real and synthetic imagery. While it remains to be seen how suitable such imagery will be for validating computer vision tasks, it is certainly an impressive achievement towards photo-realistic

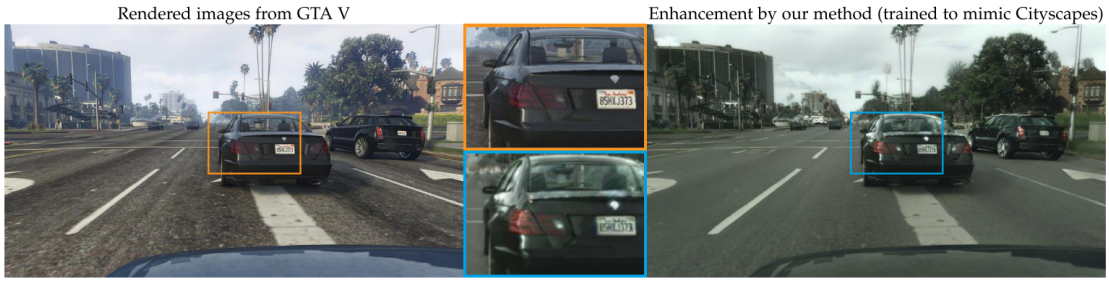

Figure 2.4: Example of post-processing performed on computer games images (GTA V) to improve the realism of the images. (image taken from [163])

computer generated imagery. Most importantly and unlike other recent examples [90], the presented approach is able to provide temporal consistency* between multiple frames, which is crucial for localisation tasks. It should also be noted that depending on the configuration, the inference of such a network needs to be operated within a fraction of a second to produce real-time results. The results shown in [163] are too intensive for this, as they require half a second of inference time on a high-end GPU. Such an approach was also considered for this thesis through a collaboration with another researcher. For the proposed project, has been assembled covering both real imagery and simulator imagery. 2500 real images from the lunar surface were taken and pre-processed from the ongoing Chang'e 4 [116] mission. For the simulator images, we used an existing dataset of 840.000 images from our Unreal Engine environment which is described in more detail in section 4.4. Due to lack of time, this project was not completed at the time of writing.

2.2.3 Lunar Environment Considerations

In this subsection we will explore the different trade-offs to consider specifically when building a lunar environment in a virtual simulator.

*Video of [163] showing the temporal consistency: https://www.youtube.com/watch?v=P1IcaBn3ejo

2.2.3.1 Real Lunar Surface Vs Synthetic Environment

Regardless of the simulation environment being used, the realism of the solution is heavily dependent on the environment that is being rendered. This is particularly interesting for the lunar surface, where we have only data of limited accuracy for most locations. From a scientific point of view, as much real data as possible should be used. As we can see in [7], this is not always possible, because we simply do not have enough data for detailed surface environments of the Moon*. The work does demonstrate that it is possible to take existing orbital data as a basis for synthetic environments. This is currently the best possible scenario, despite adding a significant amount of work to produce a more realistic simulation environment. In figure2.5 we can see an example of a 1 metre/pixel height map that was converted into a 3D mesh with the help of Open3d† and some custom code. Alternatively, it is also possible to

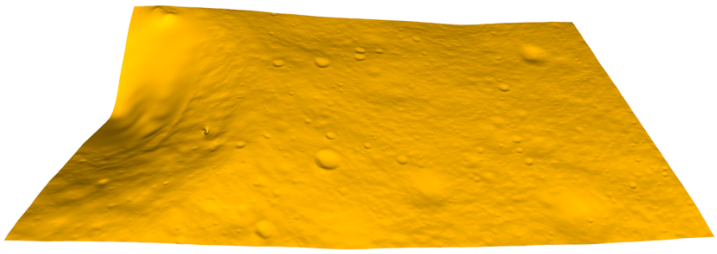

Figure 2.5: Example of a 1 metre/pixel lunar heightmap which was converted into a 3D mesh.

use existing lunar environments made for computer games. Although these environments are synthetic, they instantly provide a significant amount of realism at minimal cost. Some of these environments are also procedural, meaning that we can easily produce square kilo-

*with the exception of a handful of existing landing sites.
†http://www.open3d.org/

metres of terrain, which is useful for longer traverses. Procedural terrains can also be easily modified to generate multiple slightly different environments, which are helpful in machine learning, where over-fitting is a serious problem. Aside from the terrain, surface features such as rocks or textures can also be generated procedurally, as shown in 2.15.

The time saved from using an existing environment should also not be underestimated in the development cycles, as the simulation environment is only a single link in the chain of the development cycle. Ultimately, regardless of the simulation quality, any perception system still needs to be validated in lunar analogue environments and with real imagery, as simulators are still not able to perfectly reproduce image data.

2.2.3.2 Scale Of The Environment

When dealing with lunar or martian environments in simulation, one of the issues to deal with is how to cover large surface areas while providing high resolution detail. Additionally, we want to be able to spawn our rover in many locations, so that we can simulate many different scenarios.

This use case is problematic for simulators like Gazebo, which are simply not optimised to handle such large scale environments. As shown in [7], Gazebo can be extended to achieve this, but it requires additional modifications. Other tools like Unreal Engine, already have this type of optimisation built in with their landscape tool*, which automatically generates map subdivisions with different levels of detail, with a an area of up to 8×8 kilometres at a 1 metre per pixel resolution.

In section 4.4, we have addressed this issue inside Blender, by building our own tools to

*https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Landscape/TechnicalGuide/

generate surface geometry tiles with different textures and geometry resolutions. The different tiles are then replaced with higher or lower resolution tiles based on which part of the map was being used.

2.2.3.3 Ephemeris Model

In order to accurately simulate the conditions on the lunar surface, we need to take into account the position of other elements in our solar system. This includes the relative of the sun, which dictates the illumination conditions, but also any other objects that appear in the lunar sky, such as stars, Earth, or any other celestial bodies. In order to achieve this, an ephemeris model is needed. Thankfully, there are an existing and well maintained libraries that can be used for this purpose such as SPICE (Spacecraft Planet Instrument Camera-matrix Events)[1]. SPICE can provide us with the orientation vector of the sun with respect to our current surface position on the Moon. Additionally, Earth may also be correctly positioned in the sky based on this library. As time passes, we can then update these values iteratively. While this could also be used to correctly map the sky with stars, it is less of an issue in our use case, as a camera that is correctly exposed for the lunar surface would not be able to see any stars.

2.2.3.4 Directional Lighting

Recreating the correct scale and distances between the Sun, Earth and the Moon is possible in a virtual environment, but dealing with a centimetre level surface area and a correctly position point-light at a distance of an astronomical unit ($1au$) leads to numerical precision problems. Therefore, most render engines propose an abstraction known as a directional

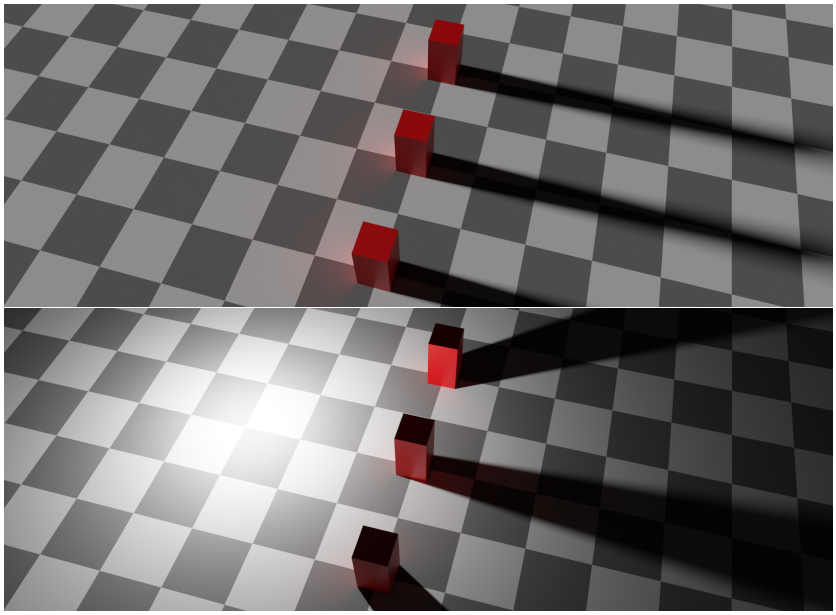

Figure 2.6: Example of shadows case by a **directional light source** (top) vs a **point light source** (bottom)

light. As shown in figure 2.6 the light rays from a point-light (and the shadows they cast) are not parallel. Given a large enough distance from the light source, they do however, appear to be so. In our case, we cannot place the point light at the correct distances, but the directional light solves this problem by simply illuminating the scene with parallel light rays from one direction. It also removes the quadratic falloff which would not be noticeable within our scene at this distance to the sun. This will then produce identical shadows to what we would expect from a point light positioned at 1*au*. This feature was available in all the simulation environments we have tested.

The brightness of the scene is less of a concern for our digital environments, as render engines do not require a set amount of photons to produce an image. Images are generally exposed correctly and can be over or under exposed if this is required.

2.2.4 Sensor Modelling

Determining how sensor data will be generated is an important factor when choosing a testing environment, as well as when evaluating results. In this subsection, we will cover how various sensor data can be simulated, and how accurate the modelling of the sensor data is.

2.2.4.1 Camera

Simulating realistic data from camera sensors is undoubtedly one of the most difficult parts of building a virtual simulation environment. Apart from the different rendering techniques described in the previous subsection, we also need to consider a couple of optical artefacts.

1. Motion-blur is a difficult effect to simulate because it is a temporal effect which requires our simulation environment to be sampled over a certain time period. One approach to solving this problem involves rendering an image in slightly varying positions. Those images are then fused to produce a final image taking into account the motion of the object. This does however lead to a considerable increase in rendering time. Today, we can also rely on a method that applies motion vectors, to blur pixels in certain directions. While this method is considerably faster, it is also lacking in accuracy, especially when separate objects move in different directions in the same frame.

   Luckily, this effect can and should be mitigated on our real rover cameras. It is the result of longer exposure times on a camera, which can be avoided by having more sensitive camera sensors, faster lenses, or simply filming in brighter environments. Luckily, the sun is even brighter on the lunar surface than on Earth due to the lack of atmosphere on the Moon. As a result, we consider this issue less problematic, with the no-

table exception of landing sites closer to the polar regions where the sun is consistently at a very shallow angle.

2. Lens distortion is an issue that affects every single camera. There are different types of distortion, as well as different models to represent them. In simulation though, we make use of perfect camera models which do not exhibit any type of distortion. This can however easily be added in post-processing. Ideally, the sampled distortion of a real camera is employed for added realism.

3. Vignetting is an effect which darkens the image the further we move away from the optical centre of the lens. This effect is a known issue present in any camera lens. This effect can also be sampled from a real camera lens, and applied in post-processing.

4. Film grain is an effect that occurs because of the differences in measurement noise experienced by every single pixel on a camera sensor. This noise is included on top of the data that the sensor collects from the environment when a picture is taken. As the noise also varies over time, it generates some small fluctuations in the image, even when the complete scene is stationary. While different in nature to noise collected by analogue film cameras, digital cameras also experience the same problem. Because simulator images do not experience this type of noise, it can either be sampled from a real camera, or generated digitally and added to an existing image. A good example of generated film grain can be found in ISAAC Sim, as shown in figure 2.7.

5. Depth of Field is a natural effect of parts of the image being out of focus. The strength of the effect primarily depends on the sensor size and the chosen lens aperture, with larger sensors and wider lens apertures seeing a more pronounced effect. In virtual im-

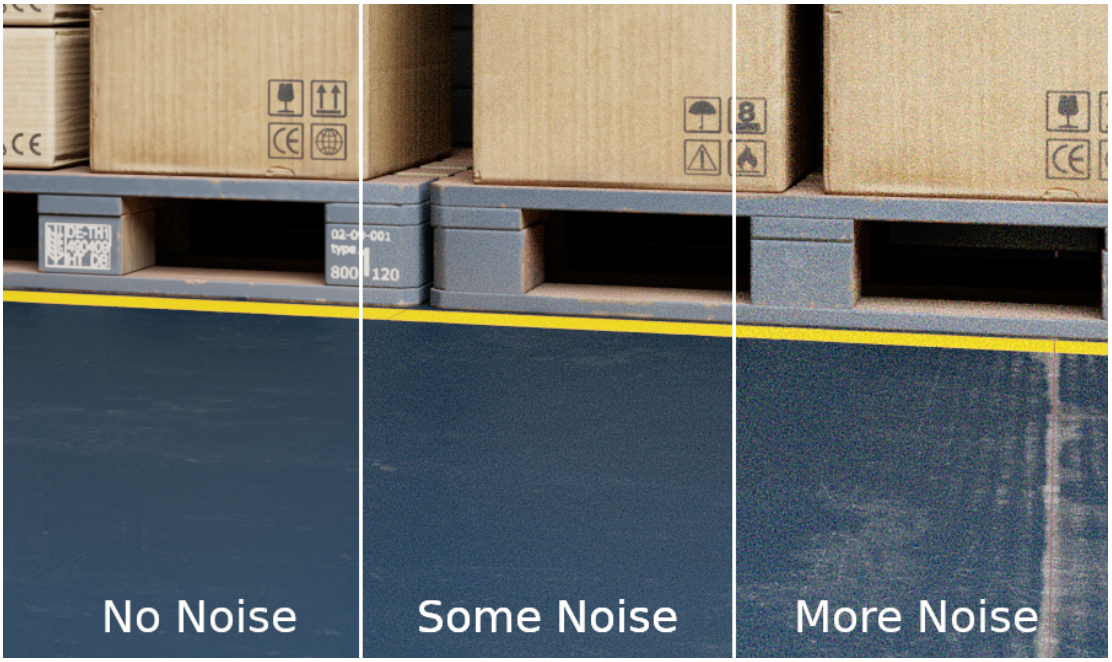

Figure 2.7: Examples of simulated film grain in camera images from ISAAC SIM

ages, this effect is not present because the used pinhole camera model has an infinitely small camera sensor. The out-of-focus effect can be simulated through the Z-Blur* method, where a depth image is produced, and each pixel is blurred based on its distance to focal plane. We see that more modern raytracing methods also have the ability to simulate this effect directly, resulting in a higher fidelity image. In our case, we decided to disregard this effect, because it is not very pronounced on cameras with small sensors typically used in our application.

*the Z in Z-Blur refers to the Z axis, which typically refers to the axis moving away from the camera in computer graphics. Sometimes it is also known as Depth-Blur.

2.2.4.2 Stereo Camera

Surprisingly, producing good stereo camera data in simulators is much more challenging than it appears. The issue here lies within the synchronisation between multiple cameras and the added complexity required to provide it. Additionally, this is a feature not often needed for simulations because many terrestrial localisation applications tend to rely on LIDAR (Autonomous Cars), RGBD or monocular camera (UAVs) configuration. Nonetheless, it is possible to achieve synchronisation, by freezing the simulation when images are rendered in sequence. Notably, this feature is not implemented in the Airsim/Unreal Engine setup. Even on dedicated robotics simulators such as Gazebo, we have encountered issues where stereo camera sensors have not always produced reliable synchronisation.

2.2.4.3 RGBD

Depth images are simple to deliver for render engines. The depth-buffer or Z-buffer is essential for render engines to determine which object is in front of which. Highly accurate depth images can thus be provided for no extra computational cost. The drawback is that in real-life, depth images are never perfect, and usually contain a lot of noise. This is because they are not measured directly, but through observing other factors, such as the distortion of a projected pattern, or the difference in perspective between two cameras. To produce more realistic depth images, it should be considered to simulate the entire sensor configuration that is used to produce depth images on the real rover hardware.

2.2.4.4 IMU

Given that simulation environments provide perfect position and acceleration measurements, accelerometer and gyroscope data can also be mimicked. Even so, for robotics applications, it is necessary to accurately simulate sensor noise. Gazebo provides a good implementation of such sensor noise, where the sensor noise is defined with *rate noise*, *rate bias*, *acceleration noise*, and *acceleration bias* [*]. Incidentally, these are the same parameters which are needed to accurately characterise an IMU for localisation purposes.

2.2.4.5 LIDAR

In comparison to camera sensors, LIDAR sensors have proven to be simpler to implement. Even if no direct implementation is available, most render engines provide ray-casting tools which can be used to cast a way into a specific direction, and measure the distance to the first object it hits. A noise distribution can then be added to mimic the ranging errors present in real LIDAR data. If the object reflectivity is available, maximum range estimates can also be computed. In order to more efficiently sample a large number of points, depth images are at times used to generate LIDAR data in simulators, because they already provide accurate range measurements while efficiently returning a complete point array simultaneously. In specific scenarios with edge cases like dust, fog or rain, realistic LIDAR measurements are more difficult to simulate. Since these effects rarely occur in lunar environments, we can safely avoid these.

[*] http://gazebosim.org/tutorials?tut=sensor_noise#IMUnoise

2.2.4.6 Star Trackers

It is technically feasible to fully simulate a star tracker by generating imagery of the sky taking into account an ephemeris model (as shown in [170]). For that reason, the added complexity, as well as the computational cost is not proportional to the gain in realism. As simulators already provide us with pinpoint accuracy ground truth, this ground truth data can also be used with noise to mimic the orientation estimates from a real star tracker, while adding minimal computational cost.

2.2.4.7 Ground Truth

Ground truth data is where virtual simulators excel. They can provide the pose of any object in the scene with pinpoint accuracy. This is especially interesting for localisation systems, where it is often challenging to correctly align the orientation of the estimated rover trajectory with the ground truth. Additionally, simulators can also provide automatically-labelled image data, which is advantageous for many machine learning applications. Additionally, simple object masks or pixel-wise semantic segmentation images can be calculated with almost no effort.

2.2.5 Evaluated Simulators

For the purpose of this work, we evaluated virtual lunar environments inside Gazebo, Airsim/Unreal Engine, Blender and ISAAC Sim:

2.2.5.1 Unreal Engine 4

Unreal Engine is a computer game engine which has been used for numerous computer game titles and as opposed to the other simulators presented here, its primary focus is on realistic imagery delivered with real-time performance. This simulator has also been used in sections 4.3 and 4.4. As the source code of this engine is publicly available, the code can also be compiled from source, and potential modifications can be applied directly.

In order to use it for our applications, we need to make the sensor data available for outside tools. Unreal provides two different interfaces to develop and automate data collection solutions. Its Blueprint interface allows for simple node-based programming. This interface lets us randomly position the camera and take screenshots. Through some simple extensions [*], it is also possible to easily write data to disk. Alternatively, it is also possible to use Unreal's direct C++ interface to automate data collection tasks. This method is preferable as it provides much more control then the Blueprint setup. Aside from its stellar real-time rendering

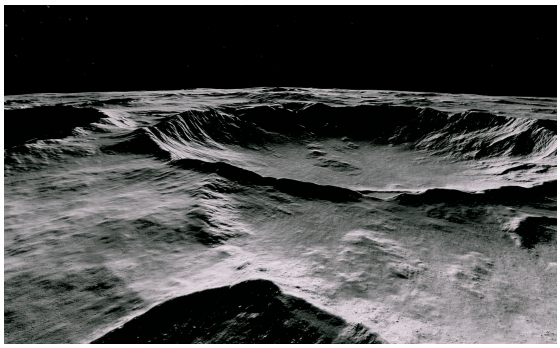

Figure 2.8: 64 km^2 lunar surface environment inside Unreal engine

capabilities, Unreal Engine also shines with its terrain optimisation tools. Unreal can provide

[*]https://forums.unrealengine.com/t/39-ramas-extra-blueprint-nodes-for-you-as-a-plugin-no-c-required/3448

large surface environments of up to 64 km^2 at resolution of 1 metre/pixel (in terms of surface geometry). With procedural shaders, additional detail can then be generated, including surface texture depending on the slope of the terrain, as well as random rock distributions. Additionally, these effects are deterministic, which is an important factor for our application. The lunar landscape used for Section 4.3, can be observed in figure 2.8. We have tested two existing ROS interfaces in order to connect our lunar environment with our robotics systems.

AirSim[177] * is the most commonly used tool to extract data from Unreal Engine. It was primarily developed for data generation in machine learning applications. The data-capturing process can be automated through Python and C++ APIs. For robotics applications, AirSim also provides a ROS wrapper † which can stream it's data to our other robotic applications. AirSim possess an active community, with regular software updates. In figure 2.9, we can observe a rover in a lunar environment, while the AirSim sensor data is displayed on the bottom, and the data stream arriving in ROS is shown in the window on the right. One notable issue we found with this configuration was that stereo cameras were not synchronised. **ROSIntegration**‡ was also tested and explored as an option, but was considered more limited, as it only provides the communication interface between Unreal Engine and ROS. While sample code is provided, the user still needs to connect the individual sensors of each rover to the provided communication interfaces. This requires an in-depth understanding of Unreal Engine, making this approach less user-friendly, and which was ultimately the reason why we did not pursue this route any further.

*https://microsoft.github.io/AirSim/
†https://microsoft.github.io/AirSim/airsim_ros_pkgs/
‡https://github.com/code-iai/ROSIntegration

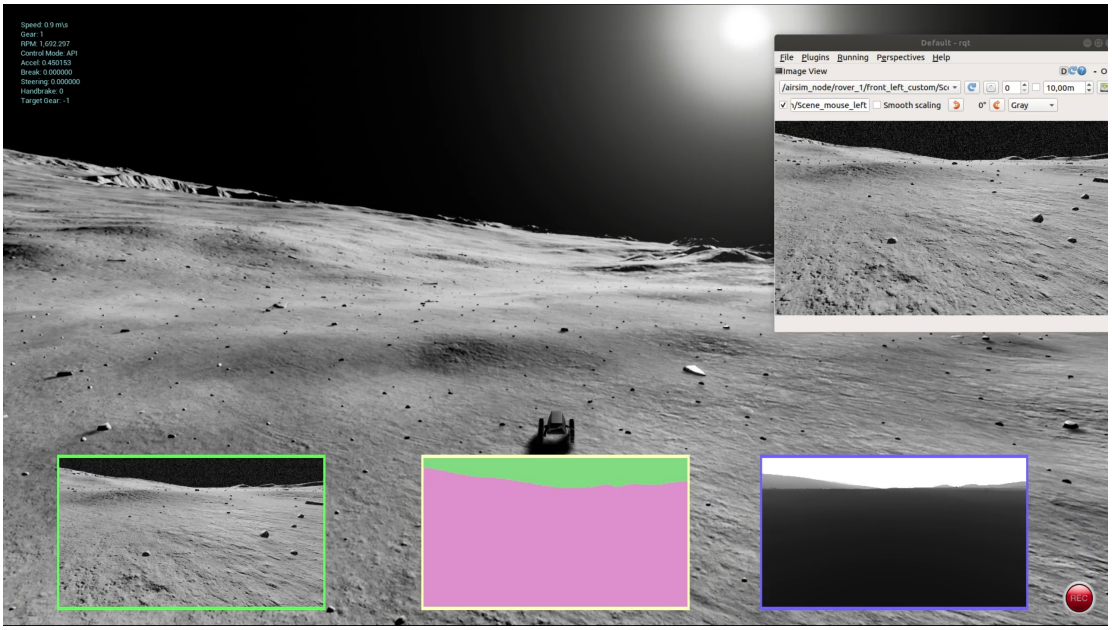

Figure 2.9: Unreal/AirSim with ROS wrapper showing a rover in a lunar environment. The AirSim sensor data (camera image, semantic segmentation image, and depth image) is shown at the bottom while the data stream arriving in ROS is shown on the right.

Another notable mention is **CARLA** ([40]) which is often used for automotive data generation. In the end, it was not tested for the purpose of this work due to limited time.

2.2.5.2 Gazebo

Gazebo [97] was used extensively throughout this work, mainly because the NASA Space Robotics Challenge* provided a lunar environment inside Gazebo (more about this in Annex A). It is by far, the most popular simulator in the research community. It maintains an excellent integration with ROS, which also makes it easy to use. It presents solutions for all commonly used robotics sensors, including some simulation of sensor noise. While it is easy to setup, it does have some limitations. It can be quite resource-intensive to run. Notably,

*http://www.spaceroboticschallenge.com/

it does provide a real-time factor, which means that if it can not simulate the environment rapidly enough, it will slow down the simulation time to still provide accurate results. We have observed that the simulator quickly slows down when the terrain size is large or if there are many objects in the scene, especially if these objects require interactive dynamics. Its 3D rendering qualities are not as realistic as the other solutions presented here. It relies on the Orge3d* render engine, which is a raster-based engine. However, due its practicality, the limitations in 3D rendering quality can be overlooked for some applications.

2.2.5.3 ISAAC Sim

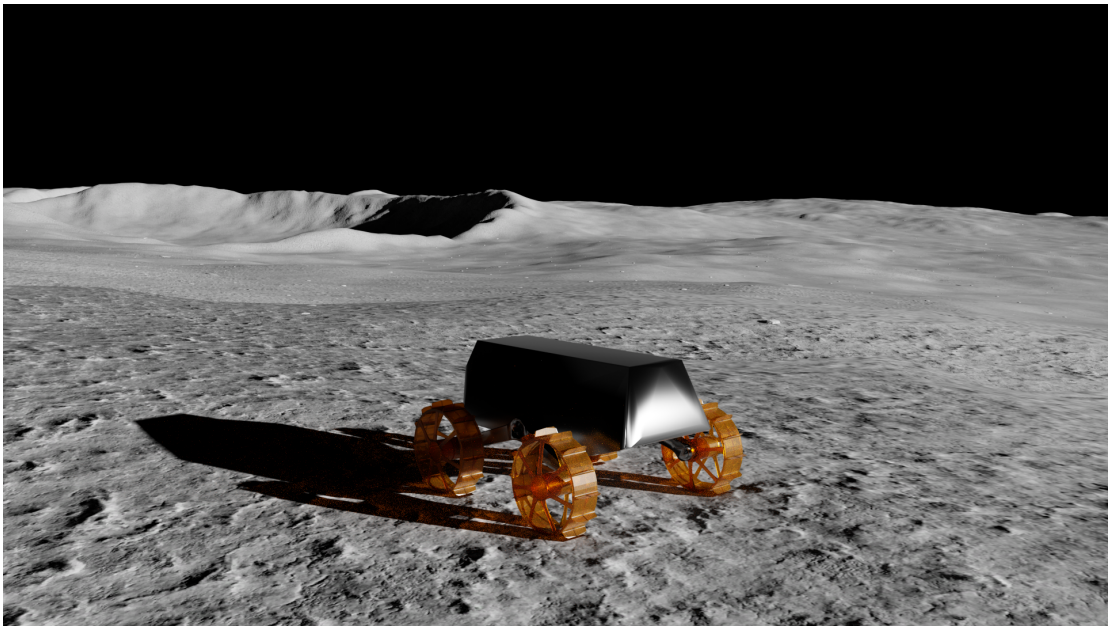

Figure 2.10: Image from ISAAC Sim using it's path-tracing render engine.

Nvidia ISAAC Sim is a relatively new simulator which was released in 2019, and is cur-

*https://www.ogre3d.org/

rently still in an Open-Beta and it is based on the NVIDIA Omniverse environment. For our work, it is interesting because it attempts to combine realistic 3D rendering with real-time robotics simulators. It provides both Real-time ray-tracing and path-tracing options to generate more physically accurate images. To produce these computationally heavy images in real-time, it offers to render images at lower sampling rates, which can then be denoised through machine-learning tools. While this process does soften the images and is not physically accurate, it still generates more realistic images than for example, Gazebo. The tool also provides a bidirectional ROS bridge, sending sensor data to ROS, while taking ROS command velocity control commands as an input. Since version *2021.1*, it also offers synchronised stereo camera images, which is an essential feature for simulating planetary robotics (as seen in 2.12). Given that it was designed with machine learning applications in mind, it provides additional features to automatically label data as shown in 2.11. It is currently still in a Beta version, which means that some features, such as sensor noise for IMU's are still missing.

This simulator looks promising for future robotics and machine learning research. Unfortunately, we were only able to test this tool briefly at the end of this thesis, as it does require access to an adequate high performance GPU.

2.2.5.4 Blender

Blender is not a robotics simulator, but can be used to generate datasets for machine learning for computer vision tasks. It is a flexible tool which can perform many tasks from modelling, animation, to rendering. It was not designed for real-time rendering, but instead offers high quality offline-rendering capabilities. With its path-tracing render engine *Cycles*, it

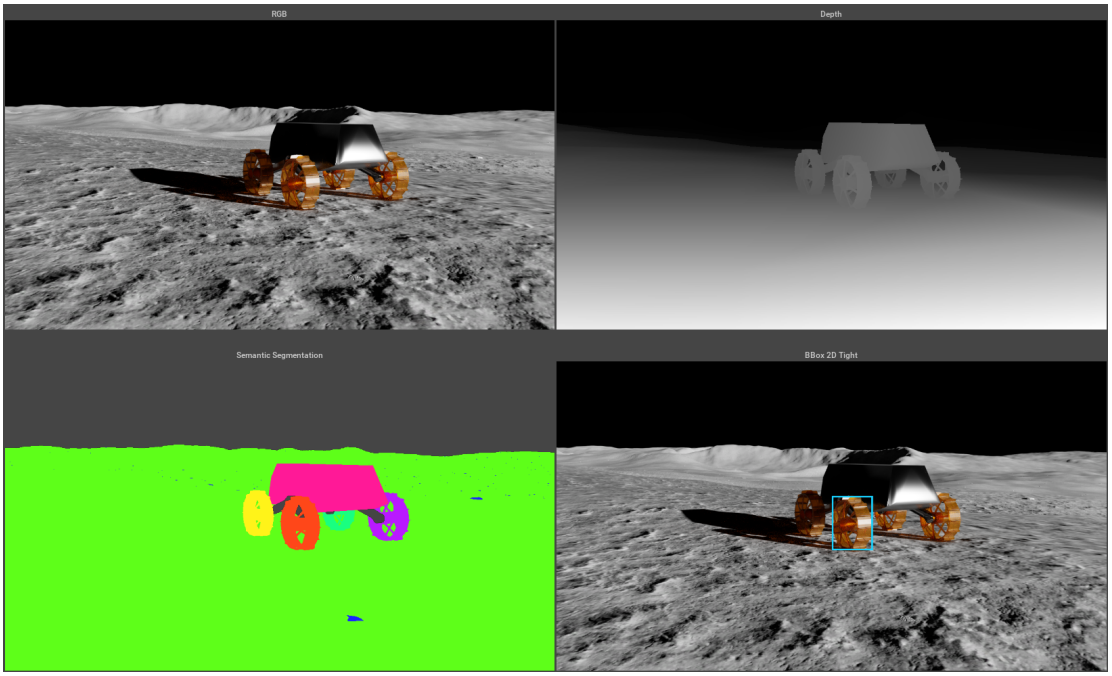

Figure 2.11: ISAAC sim producing RGB, Depth and Semantic images, as well as 2D image labels.

can produce realistic imagery. For our application, its most interesting component is the included Python programming interface. Through this interface, we can procedurally modify any parts of the 3D environment in order to automate tasks. An example of procedurally generated rock geometry can be seen in figure 2.15. This can be extended to complete scenes, the placement of virtual cameras, and automatic rendering. To render large amounts of images, this can also be performed remotely on connected computers. The work presented in Section 4.4 was produced following this approach where the camera was placed randomly based on a list of predetermined locations. Through this approach, 800.000 images were generated on a render farm. Through the Python interface, data from robotic systems such as ROS can also be loaded to produce animated visualisations, as shown in figure 2.14. This work was completed in collaboration with Manuel Castillo-Lopez for the following publication: [21].

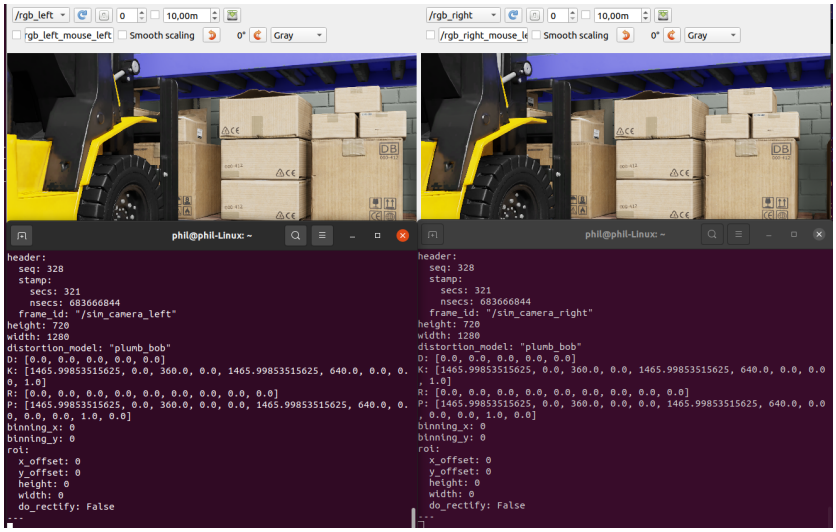

Figure 2.12: Synchronised left and right stereo images streamed from ISAAC into ROS. The top part shows the left and right image data inside ROS, while the bottom part shows the image headers, with the synchronised timestamps.

All of the previously described examples require a significant amount of experience with 3D software packages which not all researchers posses. In order to overcome this issue, tools like BlenderProc [36] or BlendTorch [79] are available to help generate different types of sensor data and metadata, while taking into account the domain randomisation required to generate datasets for many machine learning applications.

As shown in [5], Blender can also be extended to accurately render the lunar surface based on the Hapke BRDF lighting model [75]. In this example, one can also see the correctly rendered opposition effect that was observed by astronauts on the Moon.

Of all the examined solutions, Blender is the most promising option if real-time rendering is not a requirement. The realism of offline path-tracing is difficult to overcome with current compute hardware. Other offline rendering tools such as Autodesk Maya or SideFx Houdini, can also be considered for this, but Blender has the advantage of being Open-Source and free to use which makes it accessible to a broader research community.

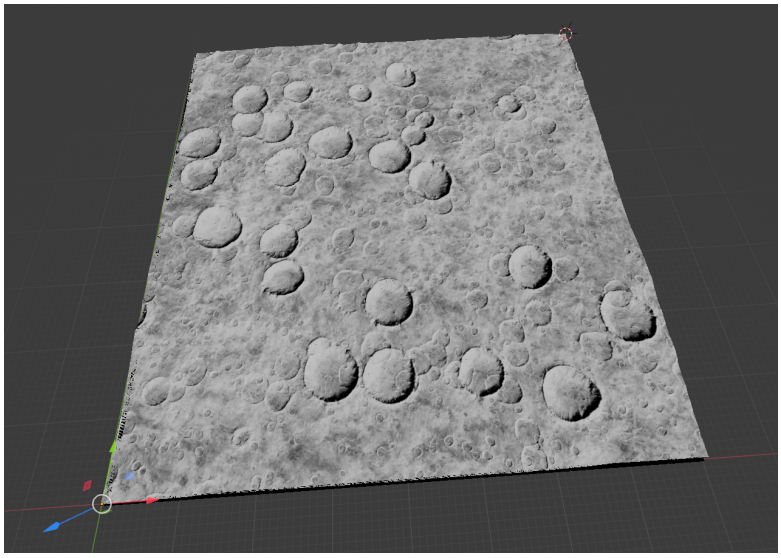

Figure 2.13: Lunar environment inside the 3D rendering software Blender.

2.2.5.5 Comparison

As visible in table 2.1 we compare some of the previously mentioned features between the four simulations environments which were used for this research. Of the evaluated tools, Gazebo and Blender stand out being that Gazebo provides the best integration with other robotics systems and Blender because of its high rendering quality. Unreal Engine and ISAAC Sim also present relevant solutions, but they do require more time to be configured for our application.

2.2.6 Advantages & Limitations Of Simulation Environments

The primary limitations of simulators are that they provide data which is too perfect. By default, surfaces are clean and present no defects. Rendered edges are often perfectly sharp and there is generally very little noise in simulated data, or it must be generated in addition

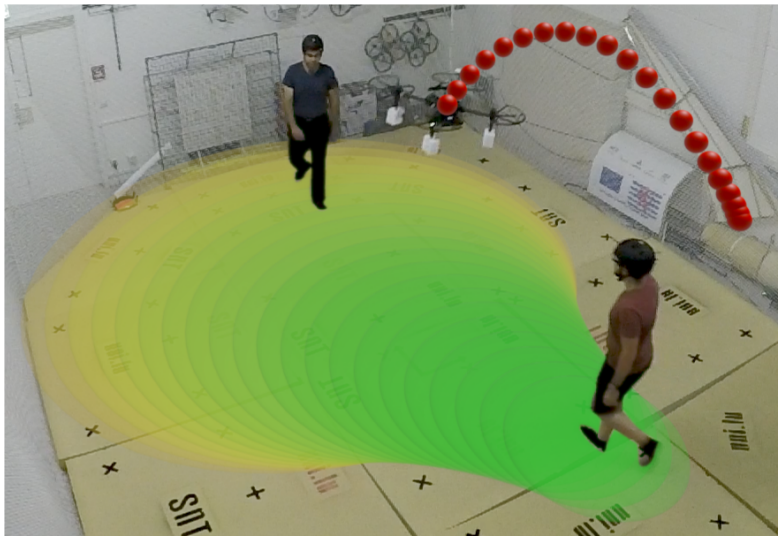

Figure 2.14: Visualisation of uncertainty in Human motion, produced with ROS and Blender.

to the produced data. The terrain resolution of virtual environments is also not without limits, especially when dealing with large-scale landscapes. When approaching objects up close, this can lead to sharp edges through the low polygon count on these objects. In terms of advantages, the simulators are able to create environments which are larger than most other testing facilities. Landscapes can also be modified easily and in a procedural way. The testing is deterministic and reproducible, which is a crucial element when trying to modify specific parameters in isolation, and is especially interesting for troubleshooting. Additionally, the perfect ground truth knowledge is invaluable for accurate benchmarking and likely the best feature of virtual environments.

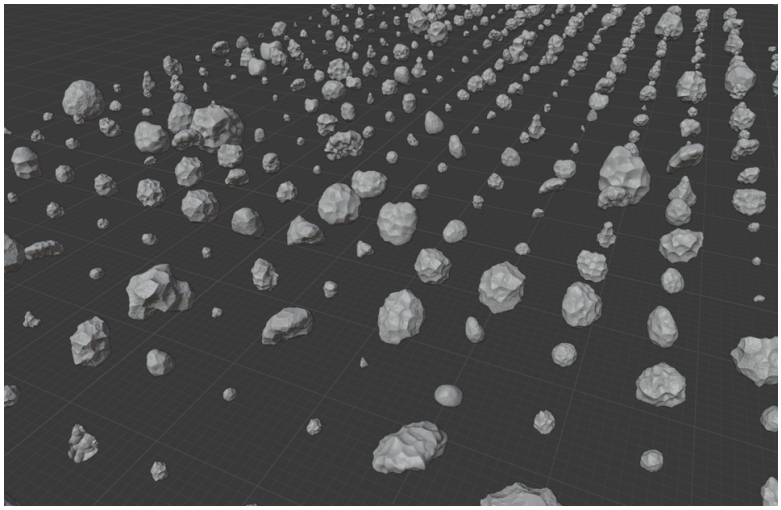

Figure 2.15: Procedurally generated Rock geometry inside Blender.

2.3 Lab Environments

2.3.1 Existing facilities

Before exploring the details on building an indoor testing facility, we list a number of existing facilities which were considered before the construction of our own environment.

2.3.1.1 NASA Ames - US

The NASA Ames Lunar Testbed [51] has a 4 by 4 metre footprint with a depth of 0.5 metres. In order to have a high material fidelity, it uses JSC-1A Regolith Simulant. In addition, the facility uses a setup of 12 lights to facilitate changes in illumination conditions for dataset generation. In the past, it has been used for rover excavation tests, as well as testing of sensor payloads for localisation purposes. The POLAR stereo dataset has been produced here with a stereo camera and a LIDAR as ground truth [215].

	Gazebo	Unreal-AirSim	ISAAC Sim	Blender
Real-time	✓	✓	✓	×
ROS-Bridge	✓	✓	✓	×
Ray-Tracing	×	✓	✓	✓
Machine Learning Data Generation	×	✓	✓	✓
Terrain LOD Optimisation	×	✓	×	×
Stereo Camera Synchronisation	✓	×	✓	✓
IMU	✓	✓	×	×
Masks/Semantic segmentation	×	✓	✓	✓
Physics Engine (default)	ODE	PhysX	PhysX	Bullet

Table 2.1: Table comparing the features of the different simulation environments that were evaluated.

2.3.1.2 DFKI - Germany

While this facility does not use stimulant or even sand as surface material, it has a large slope used to demonstrate the climbing capabilities of multi-legged robots to crawl up the steep slopes of craters [29].

2.3.1.3 KSC SwampWorks - US

Granular Mechanics and Regolith Operations (GMRO) laboratory is located in Florida, US. It is an 8m x 8m enclosed chamber filled with up to 1.5 metres of regolith. Its primary purpose is to test mechanical systems of regolith mining robots, making use of Black Point-1 lunar-regolith simulant [111].

2.3.1.4 Sagamihara - Japan

The space exploration expert building is owned by the Japanese Space Agency (JAXA). It has a 22.6m x 17.7m sandbox filled with 425 tons of coarse and fine silica sand. Because of the height of the facility and amount of sand, it has the potential to simulate rough terrain with steep slopes. Their use of of silica sand does however not mimicking the optical properties of lunar regolith.[89]

2.3.1.5 KICT Dirty Thermal Vacuuum chamber (DTVC)

The DTVC facility [45] in South Korea hosts up to 25 tons of soil with a surface area of 4.0 x 3.8 metres for roving operations. The environment can be held at vacuum and at temperature ranges between -190 and +150. This facility represents the closest one to the actual conditions on the lunar surface. However, this also limits the size of the testing ground due to the large equipment needed to produce a vacuum and the temperature changes.

2.3.1.6 MCSS - Canada

The Moon Room was built in 2020 by Mission Control Space Services (MCSS) in Canada. The primary purpose of the facility is to generate image datasets for machine learning purposes. While it does not rely on regolith simulant, its surface material consists of different pebbles, cobbles and boulders to create a variety of different surface areas. [187].

2.3.1.7 Future ESA Facility - Germany

The European Space Agency (ESA) is planning to build a 1000 m^2 indoor facility named Luna at the European Astronaut Centre (EAC) in Cologne, Germany [50].

2.3.1.8 Mars Analogue facilities

In addition to Lunar testing facilities, we have also looked at Mars analogue testing facilities, which face similar problems. Two notable facilities in this field are the Astrium Mars Yard in Stevenage [139], and the Mars Dome at UTIAS, Canada [198].

2.3.2 Building a testing Facility

During this research, two separate Lunar Lab environments were constructed. The ispace Europe Lunar Yard (figure2.17) is a 10x10 metre facility, while the University of Luxembourg LunaLab (figure2.16) measures 7x11 metres. The details below present a selection of parameters to consider when building such a facility which were explored through the existing state of the art, as well as trial and error experimentation.

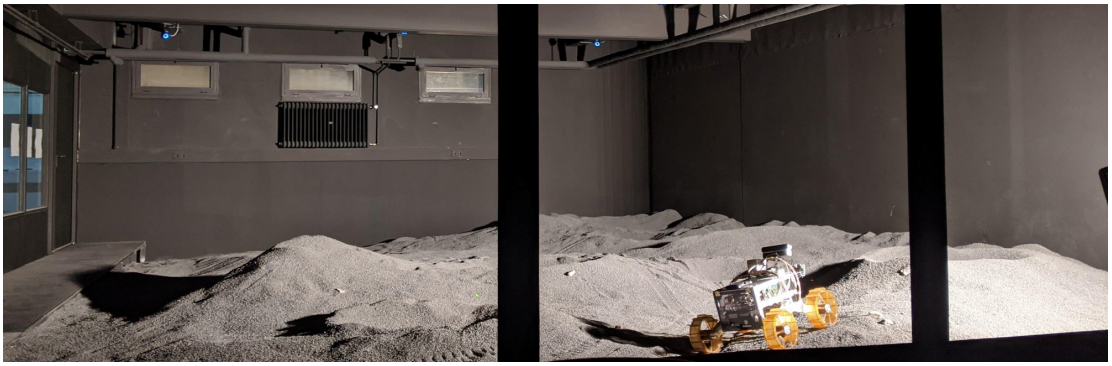

Figure 2.16: Lunalab at the university of Luxembourg. The blue lights on the ceiling are part of the motion capture system.

2.3.2.1 Material

Ideally, lunar regolith simulant is used for such a facility [214], because the material is the closest replication of the regolith on the lunar surface. It does, however, limit the operation

Figure 2.17: ispace Europe lunar yard in Luxembourg

of the facility, as regolith simulant is made from fine ground sharp particles of basalt rock which is a known carcinogenic. Additionally, it is expensive to produce, which limits the size of the facility. For daily testing, one can use rocks, gravel and sand of volcanic origins, which is sufficiently accurate, and easier to work with. It is still recommended to install a filtered ventilation system, as sand of any type tends to spread and infiltrate everywhere, including the mechanical components of robotic systems. For our tests, we have used a combination of basalt sand (0.1-2mm) and basalt gravel (2-5mm). As seen in [187], other materials, such as limestone are also sometimes used.

2.3.2.2 Surface Shape

The design of the facility surface will depend on the area of the Moon considered for testing. The lunar surface varies widely between smoother areas within maria regions, with low density of craters (approx. 73 craters per 1 million km^2; e.g. Apollo landing sites) to the rough and rocky areas with high density of craters (approx. 442 craters per 1 million km^2) [165]. Since direct surface observations are lacking for most areas of the Moon, we need to extrapo-

Figure 2.18: Opposition effect with washed out terrain features observed on lunar surface. Credit: NASA

late data to build an assumption of what a specific area could look like. When it comes to the crater and rock distributions, we look at the distribution of larger craters and boulders visible in high-resolution remote sensing images and make assumptions for the smaller craters and rocks. In figure 2.19 from [55], we can see an estimated frequency of different rock sizes for different locations on the Moon.

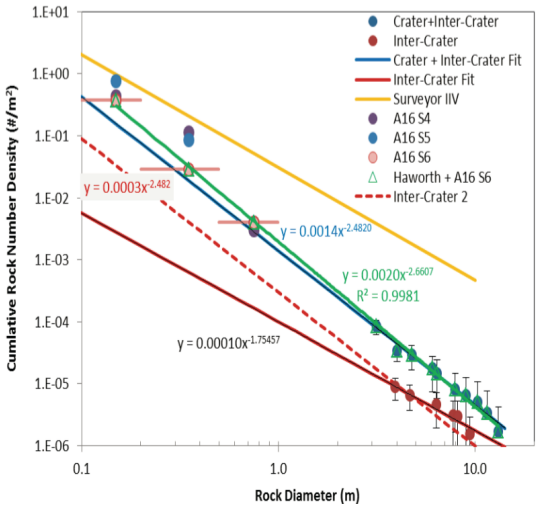

Figure 2.19: Estimated frequency of different rock sizes for different locations on the Moon. Taken from [55].

When manually crafting craters, they should be bowl-shaped and perfectly circular as described in [73] (p58). When considering the amount of surface material, at least half a metre of depth is recommended to increase the flexibility when designing craters or more mountainous regions with small hills which can cast longer shadows. This also allows for testing more edge cases with regards to localisation, but also the overall mobility of robots with steeper slopes. Another interesting approach that we can see in [215], is to rely on computer simulations to generated random surface and rock distributions, which are then implemented in a lab environment. This is useful to guarantee randomness of a facility setup. In this specific example, such a tool was used to produce many different surface configurations for an image dataset.

2.3.2.3 Illumination

To reproduce the optical properties of the lunar surface as accurately as possible, the illumination and the cast shadows play a significant role. Therefore, a single lighting source should be used to illuminate the complete environment. If multiple light sources are used, this causes each light to cast its own shadow, and the resulting shadows are also not as dark as they would be otherwise. Lights used in the film industry are suitable for this use case, as they provide enough power to illuminate the complete environment from a single light source. Even more cost effective tungsten light-bulbs are perfectly suitable to illuminate a lunar yard. While they cannot exactly match the spectrum of our sun, they cover the visible spectrum and the IR spectrum well, while being somewhat weaker in the UV part of the spectrum. When considering the strength of a light-bulb, commercial lights are given in lux output at a specific distance. With the quadratic falloff of light, we may then calculate (figure2.21) the distance

at which the light source can match the solar constant of 135000 *lux*.

Figure 2.20: 2000W Lamp used to illuminate the lunar yard.

Another element to examine is the directional light one can find on the Moon. Due to the large distance from the Sun to the Moon, it appears that the Sun's rays and the cast shadows are parallel (described in more detail in section 2.2.3.4 and figure 2.6). In order to replicate this effect, the light source should be positioned as far from the scene as possible. However, this is not very practical as it requires additional space and a more powerful light source due to the quadratic falloff of the light intensity. For practical reason and because this is less of a concern for most computer vision applications, we consider this factor as secondary.

The solar illumination is fairly consistent throughout the year, due to the near circular orbit of Earth, resulting in a distance between 1.01671033 AU and 0.935338 AU. For our calculations, we use a solar illumination constant for the Moon which is equal to 135.000

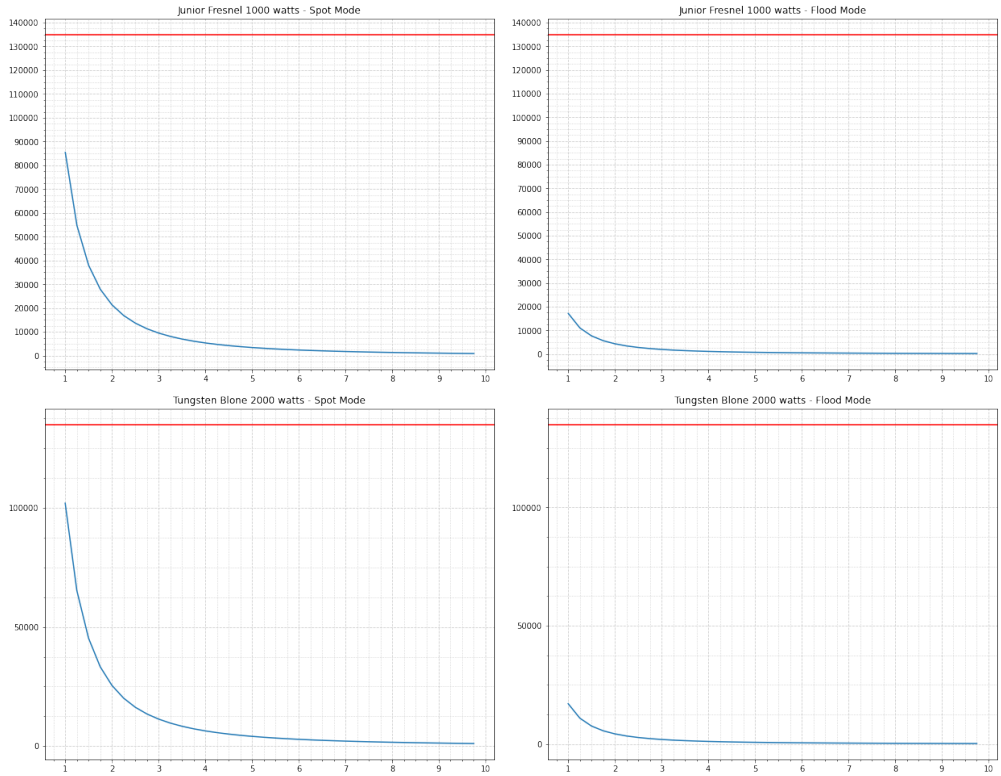

Figure 2.21: Illumination calculations for both lights used in each facility, showing the distance to the lightsource in metres on the X axis and the illumination intensity in lux on the Y axis. The red line indicates the strength of the sun on the Moon.

lux [33]. This is the approximate illumination intensity found on the lunar surface, before the sunlight is reflected of the ground. The combined illumination can be derived from the direct and indirect lighting in the target environment. For both types of illumination, we need to consider the sun inclination throughout the lunar day. For the indirect illumination, we also need to estimate the reflective properties of the surface material at the target location. We can estimate the landing site material properties from the albedo* values as measured by the Diviner lunar radiometer experiment [211]. If the sun angle, the sun intensity and the reflectively properties of the surface are modelled correctly, we can reproduce the illumination conditions to a high degree of fidelity.

For most computer vision problems, the illumination intensity is not directly an impacting factor as most cameras can adjust the image brightness through a combination of aperture, shutter speed, sensor sensitivity and ND filters. One impacting factor to consider is if images are too dark, as it can lead to longer exposure times, resulting in motion-blur. Therefore, the intensity of the light should be adapted to the size of the facility, to ensure a well lit scene.

While cameras can adapt to the illumination strength, some active illumination sensor, such as projection-pattern based systems, time-of-flight cameras or LIDARs, can be affected if the lightsource is too bright. These sensors are only able to operate if the signal they send out to measure distances is not overpowered by the Sun [201]. When performing field testing with such sensors outdoors, one should keep in mind that Earth's atmosphere not only diminishes the overall power of the sunlight, but that it also filters out certain wave-lengths at a higher rate than others. Sensor manufacturers for outdoor systems specifically pick these wavelengths because they work favourably within Earth's atmosphere and often extend the

*measure of solar radiation reflectively

range of the sensor. When testing such sensors in lab environments, light sources are thus needed, which more accurately mirror the spectrum and the illumination strength of the sun on the Moon.

2.3.2.4 Facility walls

On the lunar surface, few light-rays are reflected by the environment into shaded areas. This is because the Moon does not have an atmosphere, nor any tall structures to reflect light from. In an indoor lunar yard, the biggest sources of reflecting light are the walls and the ceiling. In order to reduce the amount of light bouncing into the shaded areas, the walls and the ceiling should be painted matte black. Curtains should also be considered to achieve the same effect. This results in darker shadows, which can also be observed in figure 2.16 where the contrast between illuminated and non-illuminated areas is much stronger. In comparison, we refer to figure 2.17 where the shadows are brighter due to the lighter environment. This effect has an impact on the localisation system because darker areas are less suitable for localisation systems to detect features. It also influences the auto exposure system on cameras, which now have to deal with more extreme values in a single picture. Darker walls are therefore useful to increase the accuracy of the simulated environment.

The walls themselves should be featureless and not covered with patterns or horizon images, as any features will be picked up by the camera systems providing ideal points for stereo depth estimation. Whilst on the Moon, stereo cameras will also see the horizon, but these features are not within the stereo depth estimation range and can thus, only marginally contribute to better localisation accuracy.

2.3.2.5 Sun Position And Movement

The angle of the Sun should be positioned with respect to the expected latitude for operation on the Moon. Landing sites in the equatorial region will endure mostly top-down illumination with limited shadows through a large part of the day. In the polar regions, the Sun will remain on the horizon through-out the day, casting long moving shadows. The simulation of equatorial regions requires facilities with a higher ceiling to mount the light source sufficiently far away from the terrain. Due to the slow movement of the Sun's positions in the lunar sky, a moving Sun position depends on the scenario to be tested. The Sun's relative position moves approximately 28 times slower than on Earth. For most types of surface navigation, this means that the movement is negligible. Nonetheless, for SLAM systems, it is important to consider if loop closure algorithms will still function accurately, should a rover revisit a location during various time periods. To examine these results, the Sun's position can be moved by hand to simulate a different time of day. To assist with these changes in positioning, robotic light mounts may also be considered for repeatable configurations.

2.3.2.6 Localisation Ground-Truth

To test any type of localisation, an external system is needed to accurately measure the actual position of the testing platform. For this purpose, we rely on a motion capture system, because these systems can typically determine a rover's position with sub-millimetre precision. Such systems use reflective IR markers which are clearly visible with IR cameras. Next, we position multiple IR cameras around the testbed. Through triangulation, the system will then determine the position of each marker and consequentially, the position and orientation of the mobile platform with 6 degrees of freedom. In order to achieve this, the motion capture

system requires at least 3 cameras and 3 markers. By using more cameras and more markers, the precision of the measurement can be improved. This also helps in avoiding occluded markers, which can result in the loss of the position tracking. When placing the markers, they should be placed in an asymmetrical pattern, to guarantee that the system does not confuse left/right or front/back of the test platform. An example of such markers mounted on a sensor can be seen in figure 2.22. The sub-millimetre precision of the ground truth is required because of the small scale of such facilities. Since the roving space is limited, the localisation error must be measured at a smaller scale as well. One factor to note with such systems, is the illumination interference with other IR based systems, such as RGBD cameras or Time-Of-Flight cameras. In order to circumvent this problem, active IR markers can be used to limit the IR illumination of the scene to an absolute minimum, while still maintaining the accuracy of the system. An example of localisation markers can be found in figure 2.22

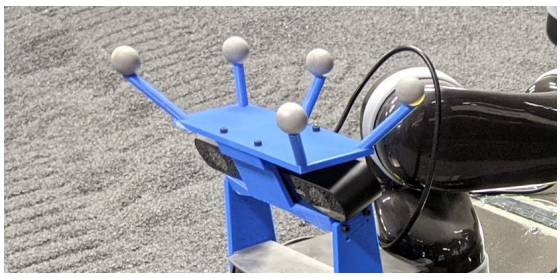

Figure 2.22: Motion-Capture tracking markers mounted on top of a rover, in order to retrieve accurate ground-truth data.

2.3.3 Advantages & Limitations Of Indoor Testing Grounds

Indoor testing facilities are ideal for regular testing where real sensor data can be collected. Since these tests are being held in a controlled environment, they are also suitable to configure specific scenarios and edge cases. On the other hand, the size of these facilities is problematic

for navigation purposes. The range of most sensors easily exceeds the size of most facilities, and often the walls are being detected by localisation systems, regardless of active or passive sensor approaches. Additionally, the small environment can lead to only a limited number of navigation scenarios being tested. Nevertheless, if testing in such locations yields favourable results, this is a good indication on the accuracy and the functionality of a navigation system on the lunar surface.

2.3.4 Existing Datasets

One notable dataset of an indoor facility is the POLAR dataset [214]. It has been recorded at the NASA Ames lunar yard, specifically to test the robustness of stereo camera systems with regards to the lunar lighting conditions, and the material properties of lunar regolith. The dataset also provides a LIDAR based ground truth, to validate the stereo depth estimation.

2.4 Outdoor Field Testing

Before flying any localisation system to the Moon, a long-range outdoor field test should be performed, ensuring that the developed system has no bias towards the simulation environment or the indoor testing environment. A good example for the necessity of such tests is the development of the Mars Exploration Rover's (MER) HAZCAM visual odometry system, where the system performed well during indoor tests, but unfortunately, turned out to be biased towards features on the walls of the testing facility [132]. Once presented on the surface of Mars, the system did not perform as anticipated. Therefore, it is crucial to test outside of lab facilities, in order to detect any potential bias towards the testing environment.

2.4.1 Field Test Examples & Datasets

Figure 2.23: Field test location at Mt. Etna, taken from the ROBEX dataset [202]

Due to the cost and time needed for field testing, initial research can also be conducted on existing field test datasets. While they probably do not match the the actual rover's size and sensor configuration, they are a good place to start testing algorithm and software implementations, before a testing platform is available. They can also be used as validation dataset, to test the software robustness in an additional environment. An example of such a lunar analogue dataset was recorded on mount Etna [202] (figure6). Mars analogues should also be considered, as the primary difficulty of lunar navigation systems lies in the repetitiveness of the environment and the limited number of visual features used for tracking and matching. One such Mars analogue is available for the Moroccan desert [108].

Figure 2.24: Test with SummitXL rover in a quarry during on our our tests.

2.4.2 Field Test Considerations

2.4.2.1 Ideal Locations

Ideal locations for testing are volcanic environments because the surface materials and rocks have similar optical properties to the ones on the Moon. There is also a limited amount of vegetation which is especially significant because vegetation moves with the wind, creating moving features which would not be present on the lunar surface. Buildings should also be avoided as they produce very angular shapes and shadows, which are not observed on the Moon. For example, volcanic areas such as Hawaii (USA), Cape Verde or Lanzarote (Spain) are good lunar analogue locations. Alternatively, desert-like unstructured environments with little vegetation may also be useful, such as the Atacama (Chile) or Mojave (USA) deserts. For the latter, the optical properties may not be quite as accurate, but the bare unstructured

environment is still useful, as it provides few visual features for navigation systems to properly localise.

2.4.2.2 Other Locations To Consider

Ultimately, it is not always a feasible option to travel to remote locations to perform the testing. When considering locations with easy access and/or closer in proximity, quarries are the most straightforward solution. To maximise the similarity between the analogue site and the lunar surface, it is advisable to look for an abandoned or out-of-hours basalt quarry with limited buildings and vegetation. The sand dunes of beaches and coastal areas can also be used for testing purposes if there is limited vegetation moving with the wind. For our research, most testing was performed in a sandstone quarry, as shown in figure 2.24.

2.4.2.3 Ground Truth

There are two types of ground truth sources which can be used for outdoor field testing: RealTime Kinematics - Global Positioning System (RTK-GPS) or optical surveying equipment [138].

The RTK-GPS solution provides a lot of flexibility because it puts few limations on the rover movement. It works in most places with good visibility of the sky and is capable of providing centimetre-level accuracy. To achieve this type of accuracy, a rover mounted receiver and a stationary receiver are needed. The stationary antenna measures the error in the signal which is then subtracted from the rovers receiver (figure2.25 right). If two GPS receivers are paired on the robotic platform, both position and orientation may also be estimated (figure2.25 left). However, the GPS signal can be subject to interference from other

Figure 2.25: RTK-GPS configuration with two GPS receivers mounted on top of the rover (left) as well as the ground station (right)

electronic systems onboard the testing platform. In such cases, shielding or receiver masts should be considered. The localisation estimates can be estimated in real-time, or through post-processing tools, such as RTKLIB [191].

Optical surveying equipment can also be used, to estimate the rover position position, but direct line of sight with a base station always needs to be guaranteed. Most total station surveying equipment can cover distances up to 1500 metres at an accuracy of up to 1.5 mm accuracy. Particularly with GPS systems, longer traverses should be considered to ensure that the constant ground truth error is smaller than the localisation error that we are trying to measure.

2.4.3 Advantages & Limitations Of Field Testing

The primary disadvantage of field testing remains in the time and effort required to organise a location, as well as the travel and delivery of the equipment. In addition, the illumination conditions of a field test cannot be fully controlled, as the Earth atmosphere scatters light,

particularly in the blue spectrum. Our atmosphere also blocks certain parts of the IR spectrum which affect testing of active IR sensors such as LIDAR. This can be mitigated to an extent by testing at night with a controlled light source, which, in turn, limits the range of the field test. The main advantage of a field test is the longer range of the tests which can not be replicated in a lab environment. The out-door tests are also necessary for verifying that tested systems are not bias towards our other testing environments.

2.5 Discussion

When addressing the different options for testing navigation systems, it is important to consider all available options to mimic the environment of the lunar surface. Planning different testing strategies is an integral part when it comes to developing systems for space to cover all possible edge cases. Particularly for lunar surface missions, this is difficult, because there is no single testing facility which can reliably reproduce all the environmental conditions encountered on the Moon. Of the approaches presented, all have strengths and weaknesses when it comes to the fidelity of the testing. Indoor facilities, while producing accurate lighting conditions and a certain degree of repeatability, suffer from the small surface area of the test setup. Virtual environments can be configured at a large scale and allow for perfect repeatability for testing different parameters. However, they still experience limitations when it comes to reproducing environment detail and sensor artefacts. Outdoor analogues help with large scale environment testing, while having downsides with control over environmental conditions as well as a significant organisational overhead. The outcome of this work provides a qualitative assessment of different testing approaches which can be used to test perceptions systems

for the lunar surface. While none of the described environments can perfectly recreate the lunar surface, each approach has its own specific advantages. We therefore, rely on the combination of all three methods to better assess the reliability of our systems. We also note that convenience is a major factor, as time consuming test setups can significantly slow down the development process. Accurate and frequent testing remains a priority in order to deliver reliable systems, especially when developing systems for the lunar surface.

2.6 Summary

The work in this chapter has been a journey through a wide variety of different fields and sources of knowledge ranging from the physical properties of rocks, to the illumination strength of the sun to efficient software implementations required for real-time simulation environments. With none of the described facilities described existing in Luxembourg at the beginning of this work, numerous literature examples were consulted. In the end however, several lessons were learned through trial and error. The first lunar yard was built in a matter of months providing invaluable insights into the different considerations for lab testing. The lessons learned from this were then addressed with the creation of the LunaLab at the University of Luxembourg, which resulted in a second lunar testing facility with a high level of fidelity. On the simulator side, game engines provide a great way to quickly generate realistic large scale environment. The integration with existing robotic systems is nevertheless not as straightforward. This is also what has led to the investigation of ISAAC, as it is a dedicated environment attempting to combine the qualities of both state of the art rendering and purpose build robotics interfaces. While this is a promising tool, it does currently not have all

the features one would expect. Given the complexity of developing a render engine, it seems unlikely that tools like Gazebo, which rely on open-source render engines, will be able to improve considerable over the current image quality in the near future.

When it comes to testing with real hardware, this is a tedious process, where more issues appear as more time is invested. The is especially true with field testing, where systems often experience new edge cases as for example different terrain types or the overheating of hardware. Additionally, said edge cases are difficult to address while being in these locations, without Internet access or even simple lab equipment, making the debugging a lengthy process.

3

Relative localisation

3.1 Background

Good localisation is an important part for all mobile robots that require any degree of autonomy. Even the most basic semi-autonomous systems require some form of localisation. While localisation is not the part that takes autonomous control, it is a necessary component to enable autonomy. A simple example of this is to command a robot to move forward by one metre. The rover will start driving, but in order to decide when to stop, it needs a metric to determine whether or not it has completed its task successfully and arrived at it's destination. In this case, simply counting the number of wheel revolutions could suffice, but for complex scenarios, more elaborate localisation systems are needed. For this type of robot-centric tasks, we are looking at relative localisation, as it does not matter where in the world the rover is, but only where it has started and where it will stop. Absolute measurements are also important, for example when trying to plan a trajectory through an existing map, or when trying to collaborate with other robots. However, this is a topic we will cover in chapter 4. Note that this work only covers online localisation methods. Structure From Motion (SFM) or photogrammetry, are also compelling tools for localisation and mapping. They are however very processing intensive, and often do not run in real-time, which makes them unsuitable for autonomous systems. Instead, they are sometimes used to support operators on the ground, as they have much more computational power available once images are down-linked.

In this chapter we will first cover the state of the art with regards to relative localisation. We will specifically address issues concerning planetary rovers while looking at both sensors and algorithms. Then we will cover considerations in developing a stereo visual-inertial localisation system for the lunar surface. Lastly we will cover the development of an vision based

localisation system which is automatically corrected through the pose estimation of known landmarks, such as the lander that deploys our rover.

3.2 State Of The Art

In this section, we will look at how relative localisation for surface robots can be performed in general, and how previous and current planetary rovers approach this problem.

3.2.1 Odometry

> *Odometry is the use of data from motion sensors to estimate change in position over time.* Cited from: [210]

In principal, we can estimate odometry measurements from any rover mounted sensors that let us infer the updated movement of a robot. This does not imply the consideration of a map, or any understanding of the environment. In terms of sensors, this can be something as simple as a wheel encoder counting wheel revolutions, to a LIDAR system estimates the transformation between two LIDAR scans at two different points in time. When working with robotic systems, odometry not only describes position updates, but also linear and angular velocity estimates. For example, this type of data is needed by systems sending drive commands to the rover's wheels to follow a path. The hardware component of the sensors used for this will be covered in section 3.3 with a more detailed look towards space applications.

3.2.2 Wheel Odometry (WO)

Wheel odometry is the simplest form of odometry. It only requires a sensor that can measure the number of wheel revolutions. Due to its simplicity, this method has been used on many planetary rovers [132, 121]. From the individual wheel revolutions we are able to estimate the overall movement of the rover, based on how the wheels are aligned to the rover. The main problem it suffers from is wheel slip. When modelling the motion of the robot in the environment, we must to take into account that the wheels will never provide perfect surface grip. In our case, the added issue is that we do not necessarily know what composition the surface has, making it even more difficult to predict the amount of wheel slip. This requires a good understanding of wheel-soil interaction. The type of motion affects this drift as well. For example, differential drive systems rely on wheels slipping sideways to control the rovers orientation. While such a system is mechanically simpler to construct, it also makes it more difficult to estimate the wheel slip. As described in the results from a field test at Mount Etna [202], wheel odometry should be tested in locations with similar soil properties, as the quality can vary significantly from lab testing.

The second issue we are facing is the complex surface shape which we are presented with. On a flat surface, it is fairly easy to calculate the robot's motion from the individual wheel rotations, but on a three dimensional (3D) surface, additional variables are introduced, which we are unable to estimate through wheel movement alone.

While WO is an important source of information, building an accurate model is a complex problem to deal with which requires a good understanding of the mechanical setup of a robot and the wheel soil interaction with the surface terrain. During the NASA SRC2

competition (Annex A), wheel odometry was considered and it worked well as on flat terrain traverses. However, once more complex surface shapes were involved, making some wheel lose contact with the ground, the pose estimation error grew much more rapidly. Because it had to compete with a visual odometry system which performed exceptionally well on simple simulator images, the wheel odometry was simply not accurate enough and discarded in the end.

3.2.3 Visual Odometry

Visual odometry covers any type of odometry generated through one or more camera sensors and image based methods. It relies on being able to estimate a pose update from the changes of a previous image to the current image. Once applied to a sequence of images, this allows us to estimate the complete motion of a camera through 2D or 3D space. In figure 3.1 we can see the different study fields in visual odometry, including the different existing sensor types and configurations, as well as the different approaches used to estimated the aforementioned pose updates. For most of the approaches we will mention, the principles will be applicable to all of the sensor types, even if they will differ in implementation.

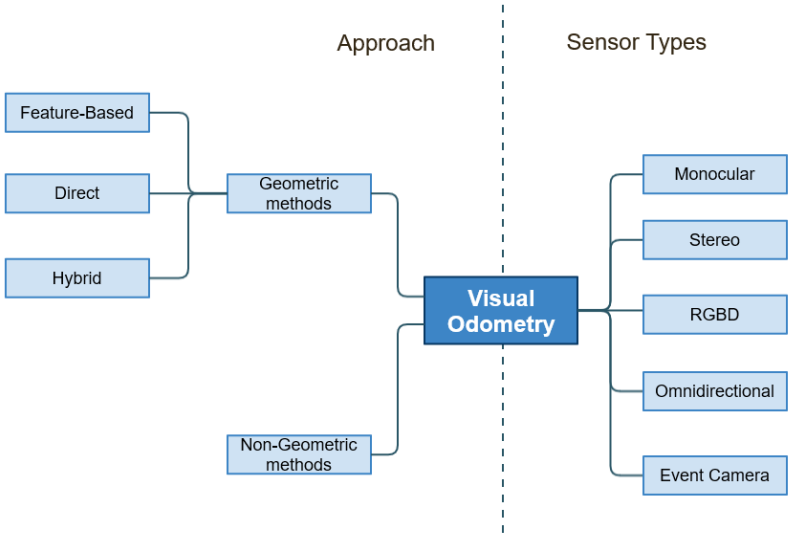

Figure 3.1: General classification of studies in the field of visual odometry

3.2.3.1 Geometric Approach

The concept of visual odometry is to rely on the slight difference in perspective between two images. For a stereo camera setup, the concept is the most simple to explain, and we will do so following the diagram shown in figure 3.2 with a 3D-to-3D estimation and the camera setup shown in figure 3.3.

In the first step, we use a *feature extractor* to search for individual features which we can detect in both the left and right image C_l and C_r. Because of the perspective change between the two images, we can then calculate the depth of each individual feature in the image based on the camera baseline (distance between left and right camera). This is done through the following formula from [87](p289):

$$depth = (baseline * focallength)/disparity) \tag{3.1}$$

Next, we repeat the same process for the previous images (P_l and P_r) and end up with two sets of features. We now proceed to the *feature matching* between the current features F_c and the previous features F_p. As we now have a relative 3D position of the same features from the current and the previous image, we can estimate the relative motion of the camera between both images C_l and P_l. Lastly, some *local optimisation*, can be performed to improve the trajectory estimation of multiple images.

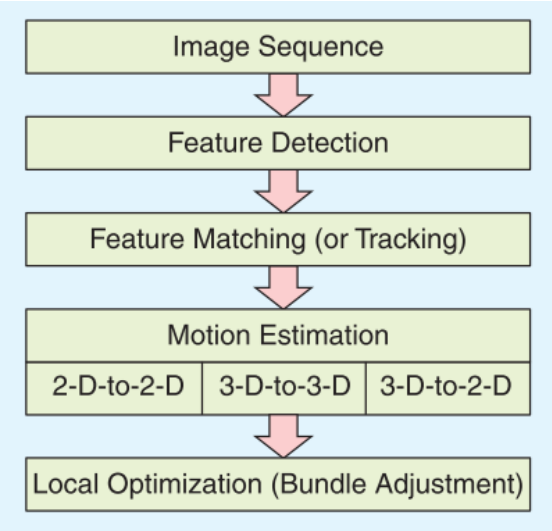

Figure 3.2: Block diagram showing the main components of a geometric VO system [172]

For a monocular camera setup, the process is similar, but the depth must be estimated through the current and previous image as well. As a result, monocular visual odometry has no way to accurately and consistently estimate the scale of the environment on its own. RGBD camera configurations are identical to the stereo camera, except that the depth estimation comes directly from the sensor, instead of estimating it through a left and right image. The 3D-to-3D correspondence seems to be the most obvious solution, but 2D-to-2D and 2D-to-3D approaches can also be beneficial. According to [148], these methods have the ad-

vantage in that they optimise the camera pose based on 2D features, which do not contain the added uncertainty from the depth estimation process.

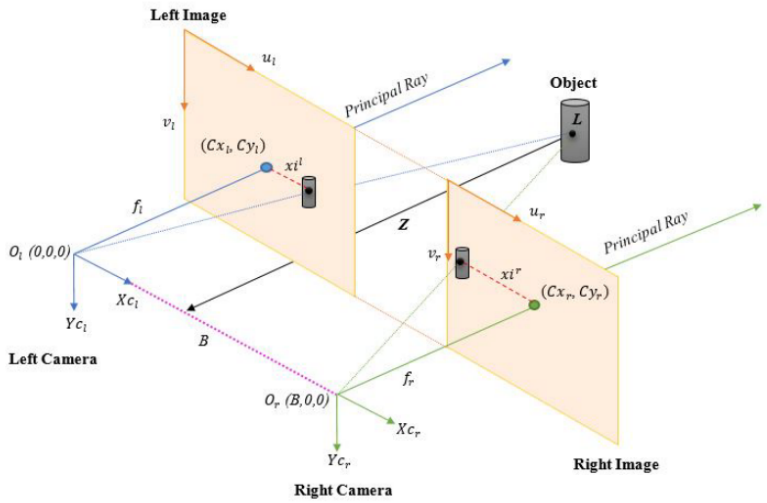

Figure 3.3: The operation principal of a stereo camera. Taken from [151]

3.2.3.2 Geometric - Direct

Early approaches to visual odometry have looked at estimating the movement of every pixel from one frame into the next frame. This method directly works on individual pixels and uses the updated position of pixels to estimate the pose update of the camera between two images. Given that this approach relies on processing every single pixel, it is computationally expensive. To simplify the process, sparse methods have also been developed. As opposed to the dense method, they apply the same principle, but first make a selection of pixels which seem more promising to track, such as pixels in high contrast areas. In general, direct methods are easy to implement but are not very robust against lighting changes or geometric distortions of objects and are rarely used today.

3.2.3.3 Geometric - Feature-based

Feature-based methods differ from direct approaches in that they first extract features from each image (feature detection), which they then try to match with features from previous images (feature matching). In principle, these features provide robust matching methods that can be invariant to specific types of transformations (rotation, scale, affine)[59] or changes in lighting conditions.

> *"One of the major advantages of the feature-based method is that it is robust against geometric distortions and brightness inconsistencies"* Cited from: [144], p6

The downside of feature-based methods is that they add computational complexity, which is directly related to the number of features selected. As a result, we can often only compare a limited number of them. Due to the added robustness and because not all parts of an image contribute usable features, this method is still preferred in most of toady's VO approaches. Examples of commonly used features include *Scale-Invariant Feature Transform* (SIFT)[194](shown in figure 3.4), *Speeded Up Robust Features* (SURF)[14] and *Oriented FAST and Rotated BRIEF* (ORB)[168]. Another issue presented by feature-based methods is that they do not work well in feature sparse environments such as smooth surfaces with no distinct texture.

Figure 3.4: Scale and orientation of SIFT features. Taken from [59]

As visible in figure3.4, feature detectors generally prefer locations with fine details. This can be problematic when some areas of the image are features spares, resulting in an uneven distribution throughout the image. Because this can impact the quality of the motion estimation, modern feature detectors divide the image into a grid, where they search for an equal amount of features in each cell of the grid.

3.2.3.4 Geometric - Hybrid

While feature matching approaches are preferred VO techniques today, direct methods can outperform them in some environments. Therefore, it can sometimes be useful to use a combination of both methods in order to produce a more robust setup, as shown in [53].

3.2.3.5 Non-Geometric

In recent years, there have been major advances in machine learning applications, especially in the field of computer vision. This can also be seen in the numerous machine-learning based

approaches and herein mainly neural networks for localisation as shown in the following survey [22]. Some of these methods can learn the motion estimation directly from consecutive images. Other methods use a hybrid approach, where the depth estimation part is taken over by a neural network, while the remaining VO components rely on traditional approaches. The benefit of using machine learning is that it does not necessarily require domain knowledge to produce a good model. The drawback is that large amounts of data are needed to train a good network, especially for modern deep neural networks. Ideally, this would be data from the lunar surface, which is not available in our case. Additionally, it is difficult to interpret the resulting models. Being a black box makes neural networks problematic when it comes to validating the reliability of a system.

3.2.3.6 Local Optimisation / Filtering

Besides simple motion estimation, more complex VO systems also apply additional post processing to the estimated trajectory. This is commonly achieved through filters such as the Extended Kalman Filter (EKF), the Unscented Kalman Filter(UKF) or the Multi-State-Constraint Kalman Filter (MSCKF). These filters are often chosen because of their low computational cost. Additionally, they also allow for easy integration with other sensor data, such as inertial measurements. As an alternative to this, optimisation strategies, such as bundle adjustment, can also be performed over several frames. This is also sometimes known as windowed bundle adjustment, as it is a local optimisation and not a global optimisation which would be performed for a SLAM system (section 3.2.7).

3.2.3.7 Application To The Moon

VO works best in static environments. The quality of the localisation depends on the quality of the images and requires that there is sufficient illumination which allows for lower sensors noise, and motion blur (through shorter shutter speeds). Additionally, a good amount of surface textures are essential to extract enough image features. The final requirement is that of sufficient overlap between consecutive images. On the lunar surface these requirements are manageable. The illumination conditions are good, with the sunlight being even stronger than back on earth (excluding polar regions). As opposed to some problematic terrestrial indoor environments, the lunar surface does present some texture, even if the landscape is mostly barren and not perfect for this type of application. The image overlap of consecutive images primarily depends on the camera field of view, frame rate and the turn rate of the rover. This is particularly important for rovers that rely on spot turns, the rotation speed needs to be chosen such that enough image overlap is guaranteed.

3.2.4 Inertial Odometry

Given an initial condition, we can estimate a pose based on a sequence of gyroscope and accelerometer data. The gyroscope provides use with angular velocity information which we can use to compute the attitude changes from the previous step to the current step. The accelerometer provides us with linear accelerations, which can be used to compute the updated positions based on the current attitude. On its own, inertial odometry present a high degree of drift. Due to its small size and because it has almost no mounting limitations, it is often used in combination with other sources of odometry. For the Mars Exploration Rovers

(MER), an initial pose estimate was calculated relying on data from wheel odometry and the IMU.[132]

3.2.5 Laser/LIDAR Odometry

LIDAR devices localisation by producing a 2D or 3D point-cloud of their environment. This point-cloud is then matched with a previous point-cloud, once the sensor has moved. Often an algorithm called Iterative Closest Point (ICP) is used for this type of odometry. Essentially it calculates a transformation matrix between two point-cloud, 'which minimises the distance between corresponding points'[144]. Other devices that can generate point-cloud can also use the same approach[169].

3.2.6 Odometry Fusion

Often, multiple types of odometry can be fused to provide better results. Adding inertial measurements to other sensor data is a popular choice because IMUs can be very small. However, this does require careful estimation of the quality of individual sensor inputs. A good example can be found in [141] and is shown in figure 3.5. Here, a VIO based SLAM system is directly compared to VO based SLAM with the same localisation algorithm (ORB-SLAM3). While VIO does not outperform VO in all the cases shown, the authors claim that the inertial part allowed the system to keep estimating the localisation accurately in some difficult edge cases. During the NASA SRC2 competition, we experimented with different Visual Inertial Odometry (VIO) algorithms (Annex A). The rover configuration was particularly interesting because it presented us with an exceptionally noisy IMU while the stereo camera was lacking a lot of the noise and image artefact which we would expect from real camera data. In this

particular case, we determined that the combination of both of these factors resulted in a worse result in comparison to the original visual odometry.

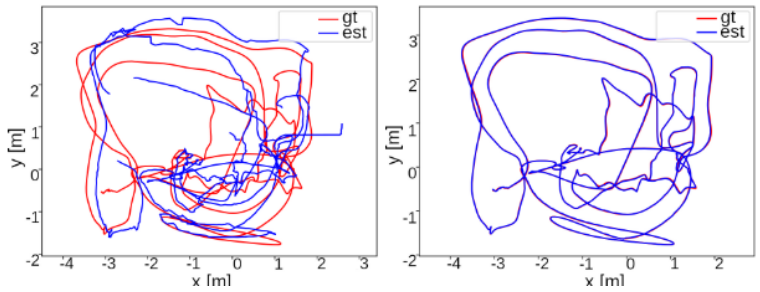

Figure 3.5: Stereo VO based SLAM on the left vs Stereo VIO based SLAM on the right. The graph shows an estimated trajectory(est) and the ground truth (gt). Taken from [141]

3.2.7 Simultaneous Localisation And Mapping

Simultaneous Localisation And Mapping (SLAM), as the name suggests, is an algorithm which builds both a map and determines the rover's pose in this map. This is useful for autonomous robotics, where a map is needed for continuous path planning and obstacle avoidance. However, even for tasks that do not require autonomy, SLAM can be interesting as the map the systems builds can be useful to improve the overall localisation accuracy. They can achieve this by detecting locations that the robot has already been to (loop closure), as well as by performing a global optimisation, rather than just optimising a portion of the map, as discussed in the previous section 3.2.3.

3.2.7.1 Loop Closure

Odometry systems continuously accumulate error over time, because they essentially look at the world as an infinite corridor (as shown in figure 3.6). SLAM systems try to address this by

looking previous data, to determine if the rover has already been to a location. Once a loop is detected, the map can then be corrected to take into account the drift between the beginning and the end of the loop, as shown in figure 3.7. This can also be performed between separate localisation sessions, as shown in [106], or for multi-robot systems as shown in [41].

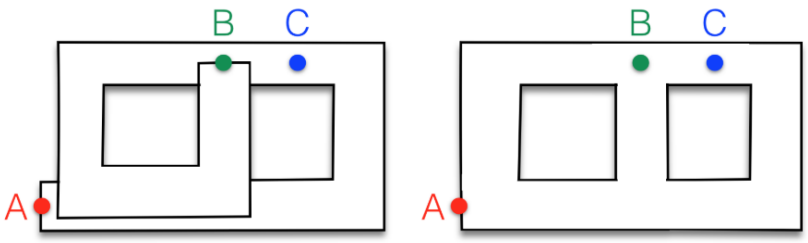

Figure 3.6: Loop closure diagram. The left side shows a map built with odometry, resulting in a long corridor travelling from A to B. The right side shows a map built with SLAM, where the system discovered shortcuts in the map. Taken from [18]

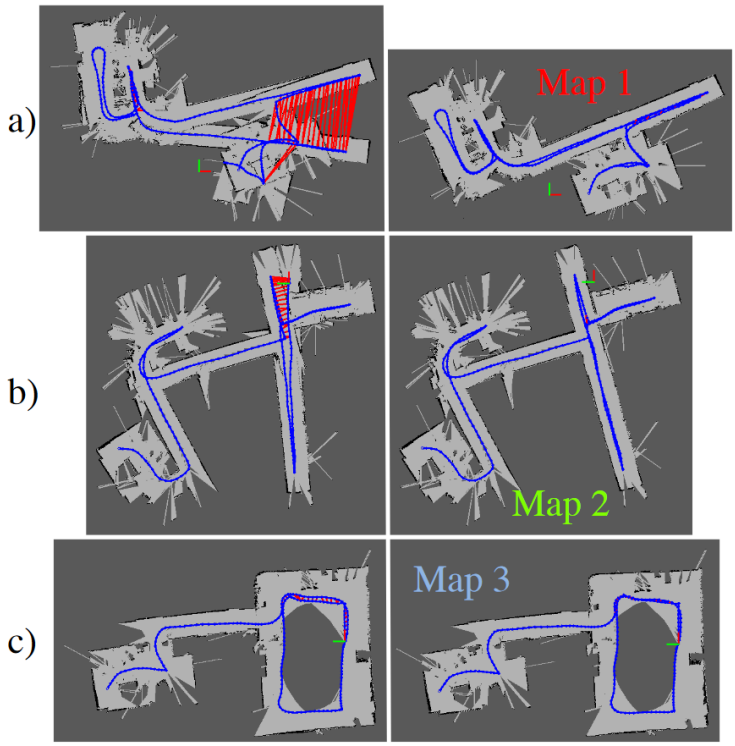

Figure 3.7: Localisation maps without (left) and with (right) graph optimisations. Loop closures are shown in red. Taken from [106]

3.2.7.2 SLAM types

The SLAM field can be broadly divided into two categories. One for systems that use a filter approach such as EKF-SLAM or particle filters and one for systems that rely on non linear optimisation techniques, which is also known as graph SLAM. The filtering approach is how SLAM was initially performed, but in recent years, graph SLAM has grown in popularity because of its improved accuracy. In [188] the authors conclude that EKF filtering can be beneficial when little computational power is available, but for any other cases, bundle adjustment based optimisation outperforms the filtering methods.

Today, most popular visual SLAM systems are based on graph SLAM, such as Rtab-Map [104], Orb-SLAM3[19] or Kimera [167], and are often based on optimisation tools such as G2o[102] or GTSAM[35].

Concerning LIDAR based SLAM, the most popular algorithm used today is LOAM [220]. Even though it has been in use for several years already, it is still performing exceptionally well in comparison with other approaches[220], as well as benchmarking datasets such as the Kitty dataset[65].

3.2.7.3 Difference Between SLAM And VO

When not considering the map component of a SLAM system, some current VO and VIO systems can look very similar to early SLAM systems. The main difference is that SLAM systems generally perform optimisations over a complete map, rather than a number of the most recent camera images. Additionally SLAM systems can detect known places and correct their map and localisation estimates through loop closure.

3.2.7.4 Application To The Moon

In [18], three parameters are given which must be taken into account when assessing a SLAM system in a given use case: the robot, the environment and the performance requirements.

The **robot** deals with the sensor suite, the motion of the robot and the available computational resources, all of which are parameters that we can directly influence, even if the compute resources on current planetary resources are quite low. Especially the slow travelling speed of current planetary rovers is beneficial to localisation systems.

The **environment** is the landing site which we are considering. This can be problematic

due to a lack of features, as well as difficult illumination conditions when the sun is positioned low on the horizon. The uncertainty about the surface terrain makes it rather challenging to assess some of these issues before the beginning of the mission. An advantage here is the fact that the environment is completely static, with the exception of the rover tracks imprinted in the soft regolith.

The **performance requirements** describe how accurate and robust the system must be, as well as the duration and area to be mapped. These factors depend on the application that we are using SLAM for and are mission-specific.

We consider two use cases here: long distance traverses and continuous operations in proximity of a lander. Both of these will have longer operations than most academic applications. While longer explorations have been achieved, they need to deal with the question on how to store large long-term map information and if all of it needs to be kept in memory. Some papers have also dealt with compressing or forgetting parts of a map. Specifically operations in close proximity to a lander benefit from long-term map consistency. As for the longer traverses, which do not intersect with themselves, the utility of SLAM becomes more marginal. At this stage, loop closure becomes less useful, and many odometry systems can provide an equally good local consistency with local optimisation. The availability of a map should however not be underestimated, as it is needed for autonomous path planning and obstacle avoidance. It is also a useful tool to support ground operators, even if the map building for this purpose could also be performed on the ground.

Currently, no SLAM systems have been used on-board a planetary rover. This is mainly due to the limited computational power available onboard. However, tests with rovers in analogue environments back on earth have been promising.

3.3 Hardware I: qualitative assessment of localisation sensors for space

In this section, we will investigate commonly used localisation sensors and outline a qualitative assessment of their applicability and usefulness in the space sector.

3.3.1 Wheel Encoders

Wheel encoders provide the most basic type of localisation sensors and are generally present on planetary rovers. It is a simple addition which not only helps localisation, but can also be used to check on the status of the motors. Additionally, it provides feedback for soil mechanics experiments, as the amount of a single wheel drift can be calculated when three other wheels are stationary, or when comparing other localisation estimates against the wheel odometry. The individual wheel rotation is measured through rotary wheel encoders, of which there are many types. As an example, commonly used hall effect encoders for example measure the magnetic field with magnets mounted around the drive shaft.

Application to Space: Wheel odometry estimated from wheel encoders is the simplest source of odometry on a surface robot, and has been used on many planetary rovers [132, 121].

3.3.2 IMU

Inertial Measurement Units (IMUs) are a popular choice in robots because they are small, lightweight, and because they have few mounting limitations. Most IMUs cover 6 Degrees

of Freedom (DoF) with both an accelerometer and a gyroscope for each axis. Some 9DoF IMUs also include a magnetometer to improve the heading estimation.

The gyroscope component measures the changes in orientation of a robot. There are different types of Gyroscopes, but generally we talk about Micro-ElectroMechanical Systems (MEMS) type gyroscopes when dealing with small sized robotics, as the optical or mechanical variants are bigger in size.

The accelerometer component can measure forces acting on the rover and respective directions. This includes gravity as well as forces generated through the rover's own movement. The gravity component is especially interesting because the accelerometer can tell us consistently in which direction is up/down. Most current accelerometers work through the help of a piezoelectric crystal, which produces a voltage when squeezed in a specific orientation. IMUs can provide absolute pitch and roll estimates, precisely because of the gravity vector. Additionally, they are useful because some graph-SLAM systems align their individual poses in their map with regard to the gravity vector, thus reducing error[218].

A magnetometer is essentially a compass indicating the direction of the planet's magnetic field. This is especially interesting to be incorporated into an IMU because the compass can eliminate drift of the yaw component. In combination with the gravity vector, this can provide more consistency in the overall orientation estimates.

Application to Space: IMUs are a popular choice on planetary rovers because of its form factor. While the accuracy of localisation estimates from this sensor alone is low, it is often used in combination with other sensors [132]. Currently, only 6DoF IMUs are used for planetary robotics because both the Moon and Mars lack a suitable magnetic field for the magnetometer to properly function.

3.3.3 Stereo Camera

The concept of stereo cameras is to rely on the slight difference in perspective between the left and the right camera, in order to extract depth cues. Our brains also use the same method to extract stereo depth cues from two images. In the simplest configuration, we assume the same setup, where two identical cameras are oriented in the same direction and positioned with an offset next to each other in a horizontal configuration (as shown in figure 3.3). Vertical configurations are also possible as shown in [109]. The distance between the two cameras is the primary factor which determines the minimum and maximum depth estimation range that can be achieved while the camera is stationary. Other contributing factors include the field of view and the resolution of the cameras. Generally, cameras are mounted in parallel, but they can also be angled slightly inwards, in order to increase the overlap between the two images for a specific distance [101]. For most of our work, we have used the ZED* and ZED2† stereo cameras, both of which have a baseline of 12 cm, and a range of about 0.5 to 20 metres. Most planetary rovers use wider camera baselines for the navigation cameras, but also make use of additional short baseline stereo cameras to cover close range obstacles. The Perseverance rover, for example, uses a Navcam with a stereo baseline of 42.4cm and a Hazcam with a baseline of 24.8cm [133]. One limitation of stereo camera systems is their maximum range limit. Longer ranges can be achieved through moving a single camera to a second location in order to produce a longer baseline (described in section 3.3.4). As shown in [149], this can be further refined with stereo cameras, as they can calculate an initial wide-baseline estimate through the odometry calculations produced during the traverse between

*https://www.stereolabs.com/zed/
†https://www.stereolabs.com/zed-2/

both camera poses.

The stereo approach requires a good calibration. Both intrinsic and extrinsic parameters need to be measured for every single camera configuration and re-calibration is necessary if the geometry of the camera setup is modified. Here, the stereo baseline is especially important, as materials should be selected that minimally change the geometry of camera baseline even under temperature fluctuations. A good example showing how significant good calibration is can be seen in [109], where a localisation algorithm is presented that is performing exceptionally well in a benchmarking dataset[65], partially because it did not assume the provided calibration as the best possible calibration.

Application to Space: Stereo cameras are the most commonly used localisation sensors in planetary robotics besides wheel encoders and IMUs. They have been proven in space exploration, and all major past and current Moon and Mars rovers have relied on stereo vision.

3.3.4 Monocular Camera

Monocular localisation approaches are often considered on drones because of the limited payload space and thus there is a wide body of existing research [146]. In order to localise with a single camera, monocular cameras need to extract depth information from the scene. However, unlike a stereo camera configuration, they cannot extract depth cues from a left and right image. Instead, when the camera is moved, depth information can be extracted from the current and previous images through a process called Structure From Motion (SFM). The primary issue with this approach is the lack of accurate knowledge of the stereo baseline. While this can be estimated through other metrics, such as wheel odometry or IMU data, any inaccuracies or drift in the baseline estimation may lead to variation in the scale of the map

and result in poor localisation performance. As a result, and as shown in [69], feature-based monocular SLAM systems consistently perform worse than stereo camera systems, although in some specific cases, the differences are not as prominent. Alternatively, learned monocular depth clues similar to the ones our brains rely on [219] are also an option, but these are likely less effective on the Moon. While occlusion clues work just as well as on earth, known-size object clues are more difficult to judge. This means that observed rocks or boulders could be of any dimension, with just the size distribution in the area being an indication of their actual scale. Interestingly, Apollo astronauts have reported similar difficulties in judging distances while on the lunar surface [15] .

While they are not able to estimate the scale of the environment, monocular camera configurations do not have a maximum depth estimation range, because the range depends on how far the camera has been moved between two images. There are scenarios where the required stereo baseline would be too large for a reasonably sized stereo camera. This is, for example, the case for observation satellites orbiting the Moon at 50 kilometres in altitude. Even close to the ground, this can be the case with drones or helicopters such as Ingenuity [12]. In such cases, monocular visual odometry should be considered as discussed in [217].

Application to Space: Similarly to the stereo cameras, monocular vision will be useful in space applications. For novel applications such as extremely small rovers[208] or UAVs[217], monocular localisation approaches are the most likely candidates, because in those cases, the platform does not provide enough space to include a large enough stereo baseline.

3.3.5 LIDAR

A LIDAR (acronym for Light Detection and Ranging) sensor is a sensor that can produce a ranging measurement by emitting a laser signal, and measuring the time it takes for the signal to return. A single point measurement can be used for example as laser altimeter during a lunar landing sequence to estimate the distance to the ground. To be useful for our localisation purposes, a more complex setup is required, with either an array of lasers, a laser on a rotating platform, or a combination of both. A single LIDAR sensor on a rotating platform is known as a planar LIDAR, and can give us a 2D view of the scene. Such a device can be found in some autonomous vacuum cleaners. A robot with such a sensor can make assumptions about the relative position of perpendicular surfaces for 2D navigation. For more complex environments, a planar LIDAR can also be used, but it requires an additional articulation to view the environment in 3D. An alternative is to use a full 3D LIDAR where an array of lasers are mounted vertically on a horizontally rotating axis. This method is preferably used by the automotive industry due to the fact that such a sensor can produce a high density point cloud and achieve distance measurements over one-hundred metres with a ranging error of $\sim$ 5cm. The error in this case with is less pronounced at closer ranges. This is significantly better than the depth estimation accuracy of most stereo camera systems. Through the rotating setup, they can scan the complete environment multiple times per second, with a resolution of several million points distributed over 360 degrees. While they can easily surpass the coverage of a single camera, the resolution is relatively sparse. In order to localise with such a sensor, we perform point-cloud matching between multiple point clouds.

Application to Space: While the localisation accuracy of both an oscillating 2D LIDAR

or a 3D LIDAR configuration are suitable for our application, the sensors themselves present some severe drawbacks. They consume a considerable amount of power through their active sensing, and the motors required to spin the sensors. Besides the additional mass of this mechanism, these moving parts also present a risk that needs to be considered for space qualifications. LIDAR devices have been used successfully in space in order to automate docking procedures with the international space station [173]. That being said, the devices in question are to big and power hungry to be considered for small lunar surface robots.

3.3.6 Solid State LIDAR

An alternative to mechanical LIDARs are solid state LIDARs, which promise to solve the problem of a complex mechanical setup with static alternatives that can diverge laser beams through other means. While these systems are promising for the automotive industry and the space industry, they are still relatively new. So far, no Solid State LIDAR has been flown in space.

Application to Space: To date, no solid state LIDAR has flown into space. This will likely change once the technology matures. The devices are interesting because they have lower power and mass requirements than mechanical LIDARs, while still providing a similar range measurement accuracy, making them an interesting contender.

3.3.7 RGBD camera

RGBD cameras are an interesting sensor because they can generally make good assumptions about the distance to an object, even in feature-sparse environments. They achieve this by projecting a pattern into the scene. This pattern is then photographed and from the distor-

tion of the projected pattern, one can estimate the distance and sometimes the orientation of the surface. The simplest pattern is a set of circles, as we can simply measure the size of the circle. The bigger the circle, the closer the object. This data can then be transformed into a point-cloud or a depth image.

There are two different methods to make this data useful for localisation. In the simplest configuration, we only have a projector and a camera. Here, we can perform localisation through matching of point-clouds with an algorithm like Iterative Closest Point (ICP)[190]. For the second approach, we need to couple the sensor with an additional camera which can record an image without the projected pattern. To achieve this, the pattern is generally projected in the Infra-Red (IR) spectrum, with one camera recording the pattern in IR, while the second camera is a normal RGB camera with an IR-cut filter. We now end up with a colour and a depth image (RGBD), the same type of data we would normally extract from a stereo camera for localisation purposes. The second method is often preferred, as these sensors often use a higher resolution colour image which provides more recognisable features than the lower resolution depth image.

The benefit of such a sensor is that it can operate well with textureless surfaces. This is especially interesting for indoor environments, but could also be useful on some parts of the lunar surface. The sensor's primary weakness is the strength of the pattern projection. Due to the inverse square law falloff of the light distribution, these sensors have a limited range. This is especially evident in outdoor environments, where the projector is quickly overpowered by the sunlight shining onto the same surface. This is also the case on the lunar surface where the illumination conditions are more intensive than here on earth.

One method to avoid this issue is combining the advantages of a stereo camera with the

advantages of a depth camera. By adding a second IR camera, stereo depth estimation can now be performed on the projected pattern, as well as regular image features. In this case, the range of the camera is now limited to the baseline of the stereo camera and not the range of the pattern (or whichever one is bigger). As an added benefit, this approach also increases the accuracy of the initial pattern depth estimation as seen in [8]. While this approach could be interesting for future work, until recently, no commercially available RGBD stereo had a suitable baseline for this application. It also remains to be seen if such a sensor would outperform a simple stereo camera on the lunar surface. The primary issue of the projection strength remains in making such an RGBD camera just as useful as a traditional stereo one in most scenarios. Nevertheless, such a sensor would provide some benefits when driving into Permanently Shadowed Regions (PSRs) which the sunlight cannot reach, or areas where traditional cameras can not pick up any surface features.

Application to Space: So far, no RGBD sensors have flown into space, and there are currently no plans to do so. Given that they rely on existing camera technology, space qualifying the technology should be feasible. As mentioned, the primary issue of the sensor is the lower detection range for outdoor applications. Once first rovers are designed to enter PSRs, the sensor could become useful because it is an active illumination sensor.

3.3.8 Time Of Flight Camera

A Time of Flight (ToF) camera relies on the same principle as LIDAR sensors, and are sometimes also referred to as Flash LIDAR. They work by emitting a light signal and measuring the time it takes for it to be reflected off an object and arrive back next to the emitter. The key difference is that a ToF camera uses a single light-source to illuminate the complete scene

rather then a single point. Next, an array of receivers behind a lens capture the returning signal similarly to a camera. This difference has a major impact on the range and quality of the system, as the focused beam used with LIDAR systems now has to illuminate the entire scene, thus reducing the strength of the signal. Similarly to the pattern projection systems found in RGBD cameras, ToF cameras suffer from being overpowered by the sunlight intensity. While it is possible to increase the power of the light source to overcome this problem, this makes it problematic to work with as it could harm human eyesight in its proximity.

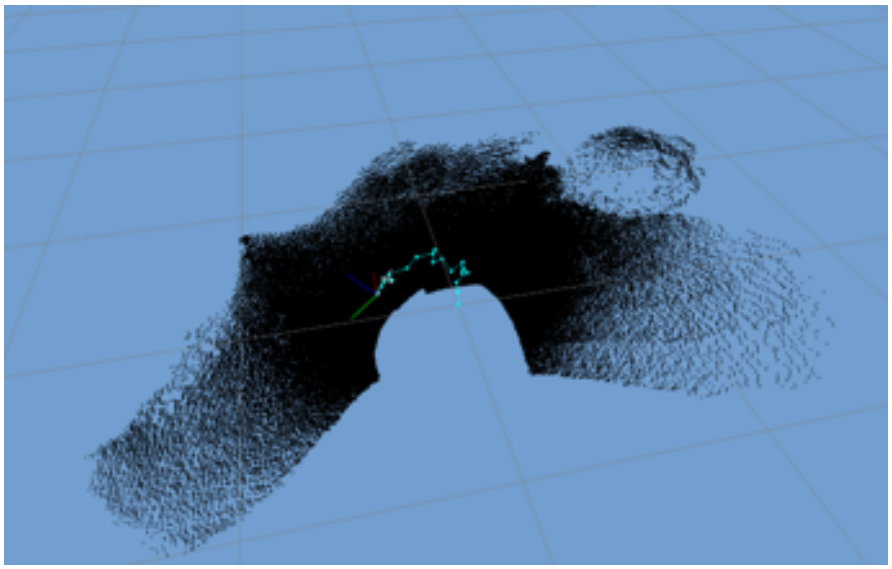

Figure 3.8: This point-cloud was generated with the help of a Time of Flight camera mounted on a rover while performing a 180 degree turn. The different sensor point-clouds were aligned with ICP

In order to provide localisation estimates with such a sensor, we can use methods operating with point-cloud information such as ICP. During our testing in the lab environment, we have successfully localised our rover on small test manoeuvres. However, during our testing, it became apparent that the small range and low resolution of our test device is problematic when dealing with environments with little geometric features. The localisation worked best

when provided with well-placed rocks or craters to provide ICP with recognisable 3D surface geometry. For simpler surface geometry, low resolution cloud matching often failed rapidly.

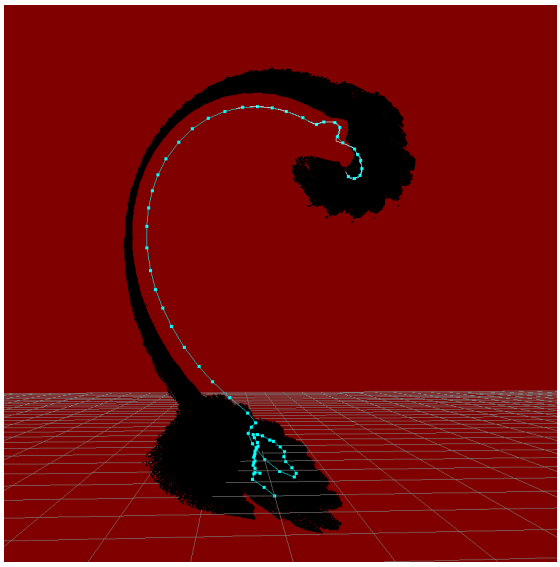

Figure 3.9: Localisation estimate based on data from a ToF camera while moving in a straight line.)

Application to Space: As described in [205], a suitable ToF camera has been modified for the use on the lunar surface, with an expected range of 2-3 metres. A similar device has also successfully flown on a cubesat [155]. While such a sensor might not be suitable for localisation purposes, it can still be used for obstacle detection, where the low range and resolution is less problematic as shown in [201].

3.3.9 Omni-Directional Camera

Omni-directional cameras can provide a 360 degree view. This makes them interesting for providing full situational awareness. Unfortunately, the downside of this approach is that the horizontal set of pixels must be divided between the full 360 degree field of view, resulting

on a low angular resolution. This means that we can only track objects which are relatively close or quite large. There are different methods to achieve this 360 degree coverage. The simplest is to point a camera upwards into a convex mirror resulting in a perfect panorama [173]. More complicated forms rely on multiple wide-angle cameras, where the images are then stitched together through a post-processing setup. In this case, we usually still encounter stitching artefacts, especially when looking at objects close to the cameras. These are the result of a perspective shift between each one of the cameras used. Similarly to the monocular camera, most omni-directional cameras are unable to directly observe stereoscopic depth. However, as shown in [110], this can be achieved through a configuration with a ring of multiple horizontally stacked cameras which cover 360 degrees and an identical vertically stacked ring providing a corresponding stereo image for each camera of the first ring [110].

Application to Space: To date, such a sensor has never flown on a planetary surface robot. While the increased situational awareness is useful, it puts considerable limitations on how to place the sensors in order to provide ideal coverage. The limited range, and the complexity of achieving accurate depth estimation is not to be underestimated. Lastly, for multi-camera approaches, the image artefacts resulting from stitching can introduce errors into any pose estimation techniques.

3.3.10 Event Camera

Event cameras (also known as neuromorphic cameras) are a new type of camera, which does not generate complete images, but rather individual pixels which have changed since the previous sample. Because this greatly reduces the amount of data that needs to be transmitted for each sample, it allows for much faster sampling rates at the order of microseconds

(1,000,000 fps). While the data format is substantially different from traditional image arrays, the technology can be used for localisation as shown in [72, 221]. This has also been shown to work on drones where high speed processing requirements are combined with low computing power [72, 203]. Currently, the main downside of such sensors is their high cost and their low spacial resolution.

Application to Space: The sensor efficiency makes this sensor interesting for space applications. Given that most of the technology is very similar to traditional camera systems, it can rely on a number of existing space qualified components. Initial lab experiments to determine the image sensor's robustness to radiation are promising. According to [166], radiation does effect the sensitive detectors, but it is also shown that this will likely not impact the ability to extract features from camera data. Additionally, an event camera has been launched into space in 2021 [46], and more data on this mission will hopefully be available soon. In simulation, first lunar applications have been tested, as can be seen in this example [182], which examines the utility of event cameras to aid lunar surface missions during the landing phase.

3.3.11 Communication Ranging

When dealing with lunar rover missions operating around a lander, the most obvious solution for communication is to use the lander as a relay instead of direct-to-earth communication. This can also be useful for localisation purposes, as we can measure how long it takes for the signal to propagate from the lander to the rover and back. Given an omni-directional antenna, this then provides us with a radius around the lander where the rover could be situated. On its own, this information is very limiting, but when combined with other sensors, can be useful to improve the localisation of an existing system. A more accurate solution can

also be seen in [39], where a ring of directional antennas is used to provide a heading estimate as well. As shown in [186], communication ranging can also be extended to swarms of robots. While the position of each individual antenna is not static in this case, localisation estimates can be estimated through the optimisation of a graph containing all the connected nodes. As opposed to the other examples provide in this section, communication ranging here is not self-contained within a single robot, it is shared between all the platforms involved in a single mission. It should also be noted that communication ranging between a rover and Earth is also feasible, but this is covered in chapter 4.

3.3.12 Sensor Comparison

When selecting a sensor for a localisation system, many factors must be taken into account. In the space industry, the Technology Readiness Level (TRL) is a common indicator if a device should be considered for flight. A device which has flown before in space will have a high TRL level, whereas a device that has only been proven to work experimentally will be assigned a low TRL level. As we have presented a number of sensors which have never been flown in space, we will discard those (Omni Camera, Solid State LIDAR, Depth Camera). We are also interested in sensors with a high localisation accuracy and we will discard monocular camera systems here because they can not estimate the scale of the environment without additional input. Equally, we remove the time of flight camera, because most such devices are too low in resolution to provide reliable localisation, especially in environments with few geometric features. This leaves us with LIDAR sensors and Stereo camera systems. Current LIDAR devices from the automotive industry would be ideal for our application, as they provide a high level of accuracy coupled with a long range. However, they do present a high risk because

to date, no compact versions have been flown in space. Additionally, stereo camera systems are superior in mass and power, which are equally important for lunar surface applications. Given the current state of the art, the stereo camera is the most suitable sensor, a fact that is validated by its common use on almost every existing planetary rover. Given that other sensor data can also be fused with stereo visual odometry, IMU and wheel odometry should also be considered. Additional monocular cameras with different viewing angles may also be helpful in addition to a primary stereo camera system. For future applications, solid state LIDAR systems will certainly be an interesting technology to follow. Similarly, event cameras have already shown promising results, and can rely on the know-how of traditional camera systems.

	Space Heritage	Range	Power	Mass	Data rate	Cost	Localisation Accuracy
Wheel	✓	/	LOW	LOW	LOW	LOW	LOW
IMU	✓	/	LOW	LOW	LOW	LOW	LOW
Mono Camera	✓	MED	LOW	MED	HIGH	MED	MED
Stereo Camera	✓	MED	LOW	MED	HIGH	MED	HIGH
Omni Camera	×	LOW	LOW	MED	HIGH	MED	MED
LIDAR	✓	HIGH	HIGH	HIGH	MED	HIGH	VERY HIGH
Solid State LIDAR	×	HIGH	MED	MED	MED	HIGH	HIGH
Depth Camera	×	LOW	MED	HIGH	HIGH	MED	HIGH
ToF	×	LOW	HIGH	MED	MED	MED	MED
Event Camera	✓	MED	LOW	MED	HIGH	HIGH	HIGH
Coms Ranging	✓	HIGH	LOW	FREE	LOW	LOW	LOW

Table 3.1: Table comparing the different possible localisation sensors. The qualities are ranked as very-high, high, medium and low.

3.4 Hardware II: Stereo camera considerations for the lunar surface

When designing small lunar rovers with autonomous capabilities, one of the questions to deal with is how to provide localisation. In the previous section, we established that stereo cameras are the most popular sensor for planetary rovers. Given the current state of the art, we

also established that while there are other sensors with interesting characteristics, including higher accuracy in localisation, stereo cameras are still the best choice when evaluating the trade-off between technology maturity, size, power and accuracy. Stereo cameras come in many configurations and can be customised for a given application. In this section, we will evaluate the different variables to consider when designing a stereo camera for a lunar rover.

3.4.1 Rolling vs Global Shutter

Rolling shutter is a type of image capture where pixels are read and recorded sequentially and there is a time delay between the first and last pixel of the image being captured. This is opposed to global shutter image capture where all pixels are recorded in parallel. This small delay between pixels is often not noticeable for most applications. Since rolling shutter sensors are cheaper to produce, they are widely used. However, for our application even small differences in timing can have an impact on localisation quality. The effect is accentuated if we either have strong vibrations in the system or when the camera is moved rapidly. The effect can also be avoided when the rover is stationary, given that we have no moving subjects in the scene on the Moon.

3.4.2 Mounting Angle

When mounting a stereo camera for lunar surface operations, tilting the camera downwards is a valid consideration. If we keep the camera horizontally, this leads to half of the capture image being above the horizon, and half of the image being below the horizon (assuming the ground is a flat plane). As the part above the horizon is mostly black, this does not provide any useful localisation features for our application. Instead, we want to tilt the camera down-

wards. We should note here that mounting the camera on a tilt mechanism would provide more flexibility in operations, but this mechanical system also presents an additional point of failure. For this example, we will assume the camera configuration of a ZED camera with a mounting height of 0.35 metres where we aim to maximise the following 3 parameters.

1. Ground visibility at the minimum depth estimation range of our stereo camera (0.5 metres).

2. Ground visibility at the maximum depth estimation range of our stereo camera (20 metres).

3. Visibility above the horizon to allow for long range mission planning.

With our camera, we have a vertical field of view of 60 degrees. If we do not tilt the camera downwards, we are missing anything that is closer than 0.6 metres to the rover (assuming the ground is flat). If we tilt the camera downwards to -15 degrees, we can see everything from 0.35 metres onwards. Since the distance between the camera and the ground (hypotenuse) is now 0.495 metres, we will also have a good convergence between the stereo images to calculate a depth image with our current image baseline. At -20 degrees, we get an extra 5 degrees of uneven terrain buffer for close range visibility. At this tilt angle, we can also see 10 degrees above the horizon, allowing us to see features that are further away, which is necessary for longer range mission planning.

3.4.3 Depth Estimation range

The depth estimation range is dependent on the baseline, FoV and image resolution. In order to decide on a threshold for the depth estimation a the stereo camera pair we need to quantify

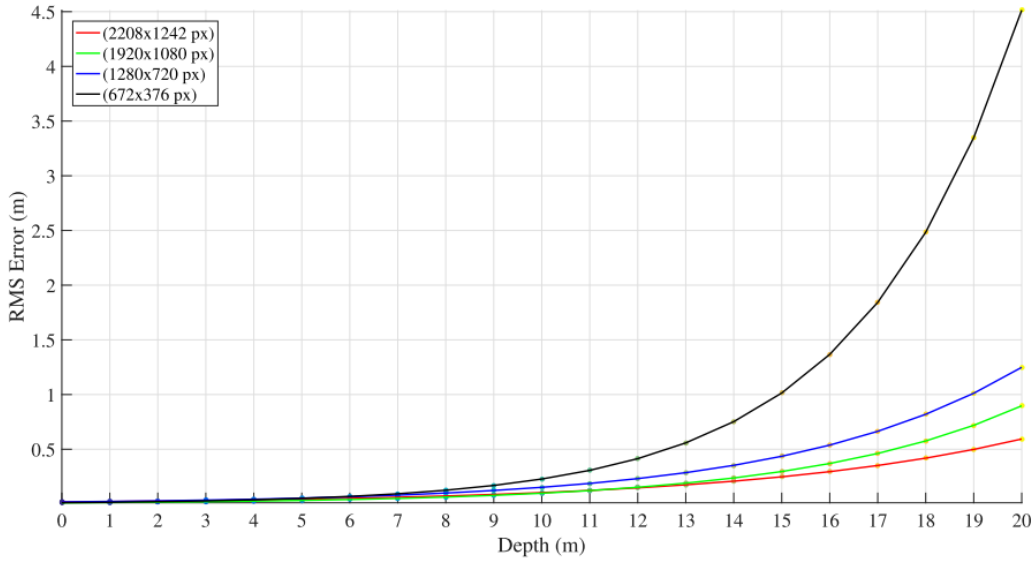

Figure 3.10: RMS error for all ZED camera resolutions (0-20m) Taken from [151]

the depth estimation error. In figure 3.10 by [151], we can observe the depth estimation error grows exponentially for a stereo camera with a 12cm baseline (the same ZED camera used in parts of this research). The graph shows the different estimate errors depending on the image resolution used. We can observe that the depth estimation error on low resolution images (672x376px) quickly exceeds a metre and even exceeds an error of 4 metres at a distance of 20 metres. For the higher resolution approaches, we note that for estimates further than 15 metres, the sensor always exceeds a depth estimation accuracy of 0.5 metres. While this error rate is less than ideal, the impact of the measurement errors does also depend on the way how it is fed into the localisation system, which is usually implemented with a confidence value for each landmark.

3.4.4 Mounting Height

One of the issues when working with smaller robots is the low profile. Most existing planetary rovers have a camera mast to give the navigation cameras a better vantage point (MER Navcam 1.5m [132]; Perseverence Navcam 1.98m [133]; ExoMars Navcam 2m [183]). This has several benefits where, on one hand, the added height lets us see past smaller obstacles. On the other hand, the surface features on the ground are easier to track from frame to frame, as there is less distortion.

3.4.5 Baseline

For smaller rovers which can only accommodate a single stereo camera, a shorter baseline is recommended as the camera not only needed to localise, but is also required to detect hazards close to the rover, which could lead to the end of a mission. The height of the mast also plays a role. If we have a two metre tall mast, being able to detect objects that are closer than two metres away is of little use on mostly flat terrain. Longer distance range estimation on the other hand are helpful for localisation purposes as well as trajectory planning.

3.4.6 Mobility

The type of locomotion can have a great impact on the localisation quality, and should be considered right from the start. Legged robots for example, have regular shocks during their movement, while robots with solid wheels pick up a lot of vibration. Changing orientation can also be different, as some robots may be unable able to perform spot turns, while they can carry out smooth longer turns instead. All these factors have consequences on the locations,

but can also be adapted to improve localisation. For instance, if a robot can only perform spot turns to reorient itself, one should consider what maximum speed the localisation can tolerate during these spot turns.

Figure 3.11: Different mobility systems tested throughout the course of this research. The DevRover on the left, the SummitXL in the middle and the Spot on the right.

3.5 Software: Rigid-body-landmark Supported SLAM

This work was performed as part of the NASA Space Robotics Challenge Phase 2 (SRC2)* (more details about this in Annex A). While the original approach and implementation was proposed by myself, this work was a collaboration across multiple disciplines and accomplished through the help of *Paul Wright*, *Frank Soboczenski*, *Maciej Zurad*, *Karthik Venkataramani*, *Swetha Pillai*, *Mathieu Labbe*, *Lukas Meyer* and *Daniel Medina*. The training data was generated by *Maciej Zurad*, *Karthik Venkataramani* and myself. The efficient pose

*https://spacecenter.org/space-robotics-challenge/space-robotics-challenge-phase-2/

training was completed by *Paul Wright*, *Maciej Zurad* and *Swetha Pillai*. The inference ROS node was programmed by *Maciej Zurad* and myself. The YOLO detector was trained by *Frank Soboczenski*. The filtering component was implemented and tweaked by myself. Lastly, the relative localisation system into which the landmarks were integrated into was setup and configured by *Mathieu Labbe*, *Lukas Meyer*, *Daniel Medina* and myself.

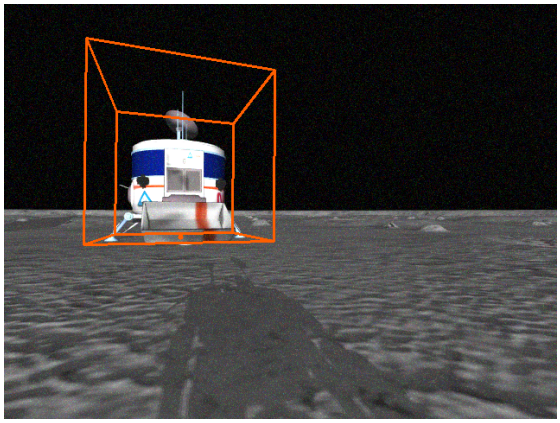

Figure 3.12: Visualisation of landmark 6DoF pose estimation

3.5.1 Introduction

As global navigation satellite system (GNSS) equivalents are currently unavailable beyond Earth, planetary rover missions rely on relative localisation approaches combined with occasional absolute localisation updates [118] to ascertain a rover position during missions. While a number of methods exist to provide absolute location updates, the simplest approach is via direct detection in orbital imagery [96, 121]. However, this method is dependent on the availability of an orbiting satellite equipped with a camera that is capable of resolving the rover in its images. As a result, this method is only of occasional use when the orbit of the satellite, the rover's position and ideal lighting conditions align [81]. Alternatively, a robot's position can

be directly estimated by comparing orbital and surface perspective imagery [71, 216, 118]. This is a time and effort intensive approach that requires human intervention for accurate results, limiting its applicability in planetary and/or multi-robot missions where signal propagation delays prevent an operator from providing real-time feedback.

Given the constraints of real-time operations, applicability to small rover systems, and need for sub-metre level accuracy, relative localisation with occasional corrections through a known landmark is a more reliable approach. Here, one possible approach is to use the lander as such a known landmark, taking into account that many planetary rover missions are performed in the vicinity of a lander. As shown in [175], fiducial marker systems such as AprilTags [150] or ArUco markers [64] can be used for this application by attaching them to the lander. However, such markers also suffer from a few drawbacks. Firstly, fiducial markers constrain lander design and take up valuable space on the surface of the lander, which would otherwise be used for solar panels, radiators and other functional components in support of the mission. Secondly, the marker pose estimation accuracy is dependent on the distance and viewing angle [176] from the rover. Any marker should therefore be as large as possible in order to reduce the pose estimation error, clashing with typical engineering requirements of making the marker as compact as possible. Lastly, multiple markers would be required for this approach in order to provide full coverage around the lander as well as to reduce viewing angle limitations, constraining the lander design even further. While it has been shown that non standard features such as light-emitting diodes (LED's) can be distributed over a flat surface to construct fiducial markers [52] to solve some of these issues, they do not address constraints with respect to the lander shape and marker size.

Recent advances in computer vision techniques have demonstrated that high accuracy six

Degrees of Freedom (6DoF) pose estimation of rigid bodies in camera images can be achieved through machine learning [17, 153, 24, 107]. This approach has the potential to replace the use of fiducial markers in interplanetary missions, assuming a rigid body is used whose features are well known beforehand.

In such a scenario, a neural network can be trained to detect the complete lander, representing the largest possible "*marker*" without any engineering modifications to the lander. Detection can thus be performed over a much larger range of distances and orientations when compared to the use of fiducial markers, given the same camera setup. Additionally, there are no restrictions on viewing angles, as it can be accounted for in the training data provided to the pose estimation network. Due to differences in geometry details and feature density, the pose estimation accuracy will vary as a function of viewing angle.

For localisation, current planetary rovers often rely on a combination of wheel odometry, visual odometry and data from an Inertial Measurement Unit (IMU) [131]. Future missions could also rely on Simultaneous Localisation And Mapping (SLAM) to build more consistent maps for path planning and localisation as shown in [175]. This is especially interesting when operating around the same location for an extended period of time. In this case, loop closure could also be used to detect when known locations are encountered again as this approach is quite accurate and can partially solve the issue of drift around a lander [105]. However, this only works when the rover and lander orientation geometry is similar to a previously encountered situation - i.e., loop closure will not be possible when looking at a lander from the 'wrong' side.

In order to address relative localisation around a lunar lander, we propose to use an existing SLAM system to which we feed our landmark position updates. Updating and correcting an

existing map with the help of fiducial landmarks has been successfully implemented in [145]. In [117], pose estimations from convolutional neural networks have successfully been used to correct a trajectory calculated by ORB-SLAM [146]. In this case however, the experiment is based on many small objects, close to the trajectory. This work also makes use of a monocular camera configuration, which is more prone to drifting than the stereo camera systems used on current planetary rovers.

To the best of the author's knowledge, this is the first application of this approach to planetary robotics. The proposed method is able to provide relative localisation updates while automatically updating its map with regards to a known landmark without the need of an artificial marker. Additionally, this is also useful for multi-robot scenarios, where these landmarks can be used to determine the pose and continuously align individual maps with regards to a global reference frame.

3.5.2 Methodology

3.5.2.1 Simulation Environment

To build this proof of concept, a simulator was used that was made available as part of the NASA Space Robotics Challenge (Phase II) (SRC2)*. It is a lunar simulation environment with full control over relevant parameters that was created using Gazebo [98]. Two surface structures are provided: a processing plant, as seen in figure3.12, and a charging station, as well as several rovers. While it is possible and beneficial to use multiple landmark objects in a graph-SLAM setup, the work presently focuses on using only the processing plant for simplicity, which we will refer to as lander for the remainder of this paper. The simulation

*https://spacecenter.org/space-robotics-challenge/space-robotics-challenge-phase-2/

can be initialised with a seed that generates random but reproducible distributions of craters and rocks on the lunar surface. The orientation of the surface structures are also chosen at random, leading to different lighting conditions for each initialisation. The simulated rovers are equipped with a stereo camera, IMU, wheel odometry and a planar Lidar. The cameras used provide a 640x480 pixel resolution with a wide field of view. Notably, the simulation includes Gaussian noise and other image artifacts in the rover cameras, which can interfere with computer vision systems.

3.5.2.2 Data Generation

Data generation to train the selected 6DoF pose estimation model, *EfficientPose* [17], presented in this work, followed an iterative approach. An initial model was trained on data collected by manually driving the rover on the lunar surface within the Gazebo simulator and extracting frames and generating appropriate labels. While the model that was trained using this data generated inaccurate inferences, it also demonstrated the network's ability to capture useful features in some scenarios. Inference errors in this case were attributed to the training data being non-representative of actual use conditions, where either a) the target object can be offset from the centre of the field of view or b) the target object can be occluded by other objects/shadows.

To address this problem, a data generation tool was created, that will arbitrarily orient the target object and the rover in relation to each other, generating viewing conditions that are more representative of actual use conditions. In the current version of this tool, the target object (e.g., lander) is initially rotated about its yaw axis at fixed angular steps. The rover is then moved through an angular step around the lander such that its distance from the lander

remains fixed. For each rover pose, the lander is rotated once around its yaw axis. These steps are repeated until the rover has taken an image of the lander from every side while being illuminated from every side. This process is then completed at three different distances from the lander (10,20 and 30 meters). At any given step, the target object can be placed anywhere within the FOV (i.e. not necessarily centered). This captures the lander and its features in a variety of orientations, distances, and lighting conditions. The simulation data is collected via rosbags[*], from which individual frames are extracted and automatically labeled. This requires synchronization between image capture and generating position labels, which is ensured by freezing the simulation when data capture occurs. Data capture involves recording the image frame as seen by the rover, along with the generation of image masks highlighting only the target object, which is required by EfficientPose. This is done in pyrender [†], using the known camera intrinsics for the simulation environment, a transformation matrix capturing viewing geometry, and a 3D model of the target object. The resulting images, masks and transformation matrix were transformed into the same format as the Linemod dataset [83] for training EfficientPose. The training dataset generated in this manner is representative of the conditions under which the rover will perceive the lander when operating autonomously.

3.5.2.3 Neural Network

The work presented here, follows the EfficientPose[17] model architecture which is based upon the state-of-the-art 2D object detection architecture family, EfficientDet[193], capable of detecting objects and estimating their 6D pose at approximately 30 frames per second (FPS). While other solutions exist, EfficientPose was chosen because at the time of experi-

[*] http://wiki.ros.org/rosbag
[†] https://pyrender.readthedocs.io

mentation, it was ranked #1 in the "*6D Pose Estimation using RGB*" on the Linemod dataset [82]*. We used the official implementation available on Github†, with a slight modification to increase the weight on the transformation component of the loss function from 0.02 to 0.5. As with the official implementation, we trained EfficientPose starting with a pre-trained EfficientDet model on the Common Objects in Context dataset (COCO; [120]).

During our experimentation process, we have primarily concentrated our efforts on curating a representative set of training data, but we have performed a limited hyper-parameter search on the component-wise weighting of the loss function. We also experimented with providing EfficientPose with pre-trained weights from training EfficientDet on problem-specific images.

3.5.2.4 Filtering

The resulting EfficientPose network finally generates reasonable pose estimation. However, a number of edge cases still need to be addressed, because they are not acceptable for relative localisation systems, as wrong pose estimation will result in large divergences in the generated map. As only a few correct detections are needed to improve the localisation estimates, any possible false positives should be discarded, even if some correct detections are removed in the process. This is achieved by relying on other available information.

An additional object detection neural network is used, which is based on You Only Look Once Version Three (YOLOV3) [161]. This was originally trained to differentiate between different objects in the simulation environment. In the simplified lunar environment of the SRC2 competition, YOLOV3 gives exceptionally reliable results, even with a relatively low

*https://paperswithcode.com/task/6d-pose-estimation
†https://github.com/ybkscht/EfficientPose

number of 2000 manually labelled images. The YOLOV3 neural network used in this work was based on the Darknet-53 architecture. However, initial results showed a suboptimal inference time. Hence, the architecture was refined to a custom YoloV3-Tiny [2] version with fewer layers and an alternate optimizer namely *AdamW* [125]. The final version performs inference tasks in less than 0.1 seconds. The overall results of this detector have shown no false positives and a model accuracy of 95%.

Because the confidence in the YOLOV3 detector is high, it is used to search for the lander, before the pose estimation is executed. This provides multiple benefits: a) The YOLOV3-based detector runs three times as fast than the EfficientPose network ($\sim$ 0.15sec vs $\sim$ 0.5sec). EfficientPose is therefore only being executed if the lander has been positively identified. b) The pose estimation was only trained on images of the lander in order to accelerate training, leading to pose estimations on other structures that it has not seen during training. Once the YOLOV3-based detector provides a detection of the lander, any parts of the image which are outside of the lander bounding box are turned black, in order to avoid estimations on other objects. In this case, the bounding box needs to be slightly bigger than the 2D bounding box from YOLOV3-based detector, as the 3D bounding box occupies a bigger space in the image. c) We filter out cases where the object bounding box is touching the edge of the frame (top/bottom/left/right), as a correct pose estimation is much harder in these edge cases. d) Based on the bounding box size, the approximate distance to the lander is known and can filter cases where the lander is too far for an accurate detection. As shown in figure 3.16 the estimation accuracy decreases with range. This step could also be supported by the stereo camera depth estimation, however this was not necessary in this case.

Once a YOLOV3-based detection passes the initial filtering steps, the EfficientPose net-

work is used to filter the remaining results again. If EfficientPose returns multiple results, these are usually faulty and are therefore discarded. This generally happens when we are very close to the target object and when the object is not fully contained within the image.

3.5.2.5 Localisation

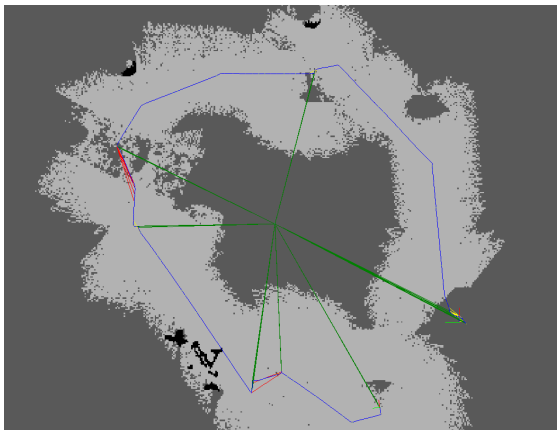

Figure 3.13: RTAB-Map graph, showing the estimated rover trajectory (blue), loop closures (red), as well as the landmark connections to the the centre (green), where the lander is located.

For the onboard rover localisation, a stereo camera is used to provide Visual Odometry updates, which are then fed into a Real-Time Appearance-Based Mapping (RTAB-Map) [104] to provide full Simultaneous Localisation And Mapping (SLAM), including loop closure. To integrate our EfficientPose position updates, we make use of the existing landmark feature of RTAB-Map, allowing us to feed landmarks with a 6DoF transformation matrix, as well as a covariance matrix describing the confidence in the provided landmark. The landmarks can be identified in RTAB-Map's graph view with a green line. As seen in figure3.13, each landmark detection along the traverse (blue line) is pointing to the same point in the centre (green lines). By default, this system works similarly to loop closure, as there is no ab-

solute position attached to these landmarks. Each subsequent detection provides additional information to the graph solver used by RTAB-Map (g2o [103]), while taking into account the covariance matrix we provide.

An additional element we need to consider here is what values to use for the covariance matrix. As mentioned, the covariance matrix describes the confidence in our landmark and this is presented by a 6×6 matrix which addresses different combinations of confidence into the six x, y, z and r, p, y values. If the confidence is not known, a simple unit matrix can be used. In our case however, we can evaluate which values to use empirically by testing a number of different combinations to best cover the entire search space.

In order to test this efficiently, RTAB-Map's database feature is being used to record rover traverses. This includes the odometry, any type of landmarks and the ground truth information of the simulation environment. Since we are only interested in modifying the links between the different components of the graph-SLAM for our evaluation, a database can be recorded during a traverse and subsequently modified in order to test different scenarios. Additional tools have also been built to completely remove certain types of links for testing purposes. Once modifications have been applied to a database, the graph is then recomputed, providing us with an updated trajectory estimate. The new estimate is compared against the simulator ground truth, which is also stored in the database. As an evaluation metric, the Absolute Trajectory root-means-square Error (ATE) [189] at the last rover position is used to provide a single value for each test. This value, allows us to compare a database without any of our EfficientPose landmarks against databases including landmarks with different confidence levels. With our approach, reproducible results can be generated rapidly, without the need to rerun the complete SLAM setup for each test.

In order to create a representative sample, we produced multiple traverses (23), while varying the distance to the lander, the total driving distance (169-867 metres), the number of times the lander is visible, and the angle at which we return to the lander. At the start of each test, the lander is in the field of view of the rover. For these tests, the rover was driven semi-autonomously through a waypoint system. For some of the traverses, loop closure of SLAM system was incentivised by driving the rover back to its original station position. The lighting conditions were also changed during some tests in order to evaluate the robustness of the EfficientPose network against illumination changes.

3.5.2.6 Multi-Robot Scenario

In addition to the single-robot scenarios, a multi-robot use case has also been taken into consideration, where we have implemented an extension to RTAB-Map which considers the lander as the centre of the map, meaning that our map is automatically aligned to the lander as soon at it appears in the rover's field of view. With the help of this extension, multiple robots are able to interact in the same reference frame, without the need of a shared map.

3.5.3 Results and Discussion

3.5.3.1 EfficientPose vs Ground Truth

Before evaluating the EfficientPose model integrated into a SLAM system, we first evaluate the estimation accuracy of a test dataset against ground truth as can be seen in table 3.2. In figure3.14 we can see the complete test dataset where the estimated pose is plotted relative to the lander. Overall, the estimations are in roughly the correct position, without any cases on the wrong side of the lander. We also notice that most errors are introduced through

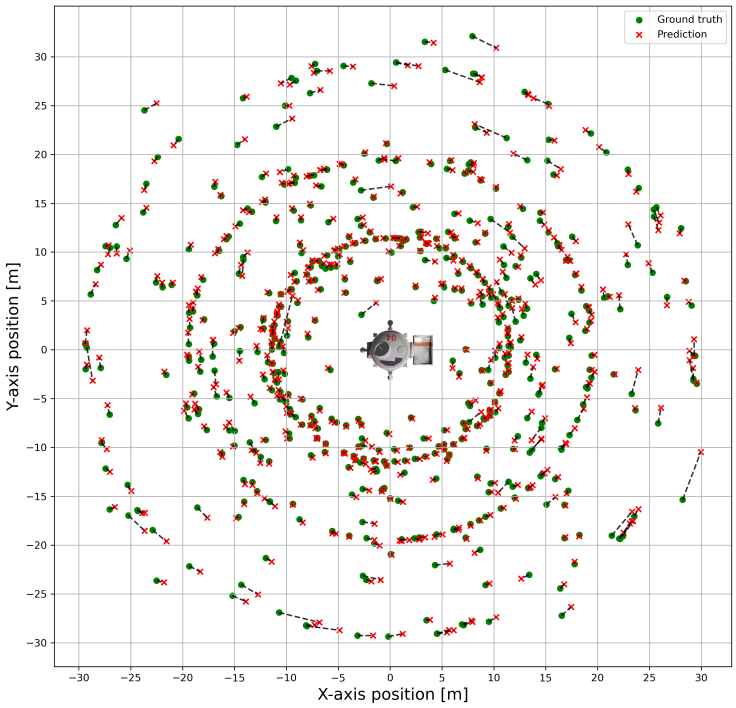

Figure 3.14: Top-down view of ground-truth (green) rover position and its corresponding estimation (red) relative to the lander. For visualisation purposes, we plot the inverse of the estimation where the lander is stationary, and the rover position is plotted

deviations in the orientation, while the distance estimate is accurate within 0.224 metres in 90% of the cases as seen in figure3.15. We can see a correlation between the estimation error and the distance to the lander, as can be seen in figure3.16. Furthermore, we can also see that there is no significant accuracy difference regarding the detection from different angles, as can be seen in figure3.17.

TranslationErrorMean in metres	0.1390
TranslationErrorStd in metres	0.0919
RotationErrorMean in degree	2.1131
RotationErrorStd in degree	2.1975

Table 3.2: Error values from EfficientPose Test dataset

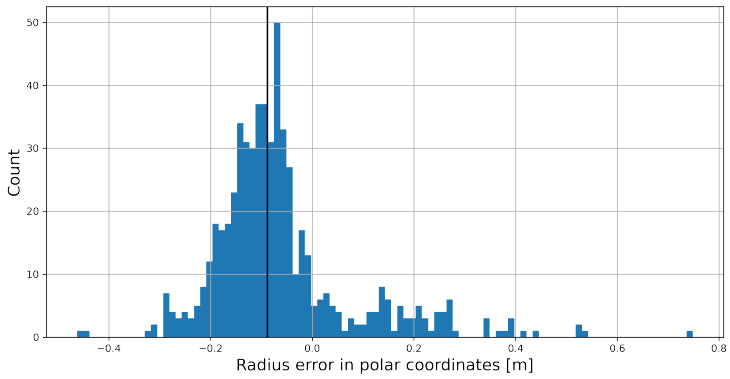

Figure 3.15: Polar distance error at a distance to the lander between 5 and 35 metres (excluding any position error introduced to orientation offset).

3.5.3.2 Landmark SLAM vs Ground Truth

In figure3.18 we can observe the impact that our method can have on the estimate trajectory. We also note that the use of a unit matrix is not ideal, and searching for a custom covariance matrix is necessary for providing useful results.

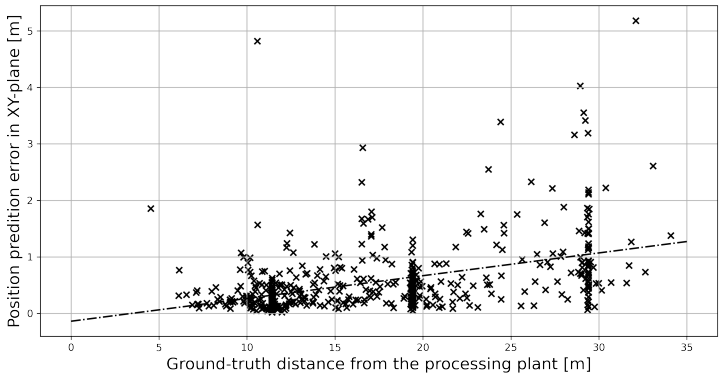

Figure 3.16: Distance from the processing plant vs position error $(R^2 = 0.448)$

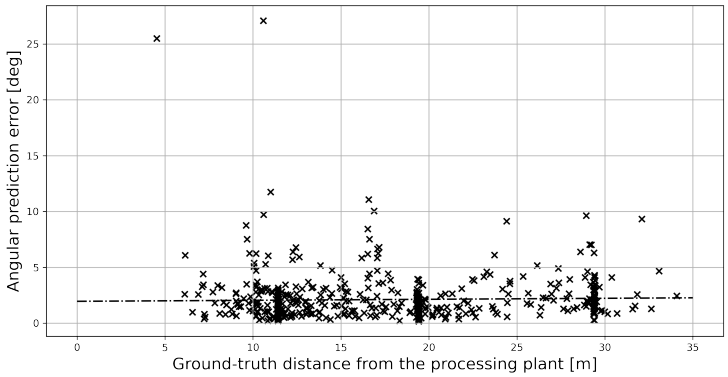

Figure 3.17: Distance from the processing plant vs angular error $(R^2 = 0.028)$

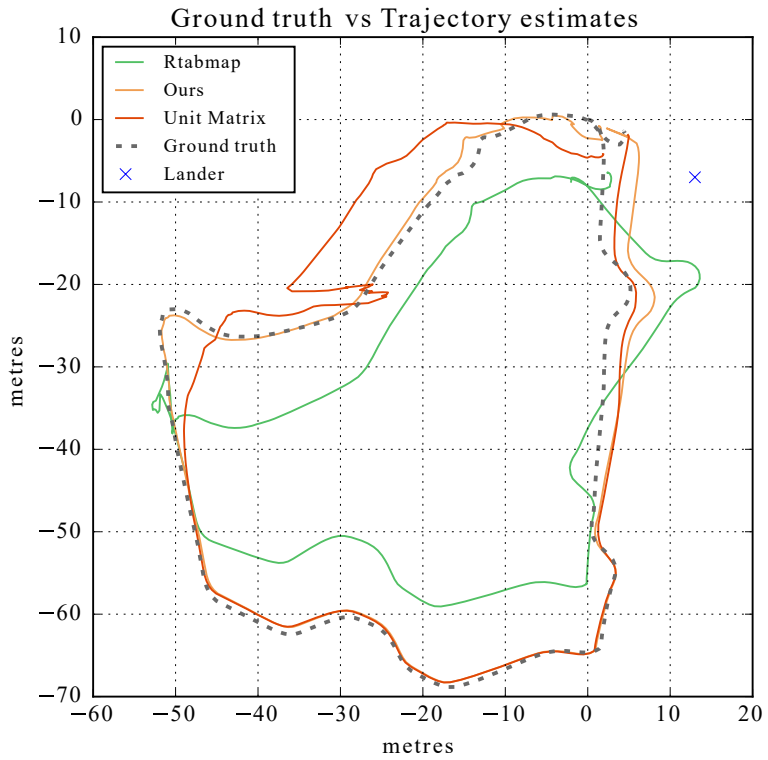

Figure 3.18: Trajectory estimate of different SLAM methods vs the ground truth measurement for a traverse number 5.

To evaluate different covariance matrix options, a range of different options have been tested to analyse the search space, starting with a unit matrix. Initial modifications were performed considering a single matrix to be applied to all landmark links. In this case, the best results were found when giving high confidence in the position estimates and low confidence in orientation estimates. The resulting accuracy can be seen in table 3.3, where we note that localisation performs better than the original SLAM system in some cases, but worse on average.

Based on the results from figure3.17, we establish that EfficientPose performs worse when the lander is further away from the camera. For our second approach, the covariance matrix is therefore modified to reduce the landmark confidence based on the estimated distance. This is not relying on a ground truth measurement, but the distance estimation provided by EfficientPose. High confidence is placed into nearby pose estimations (1 metre), and gradually lower confidence in pose estimations which are further away (1 - 31 metres). Pose estimations that further away are giving a very low confidence value and effectively discarded ($>$ 31 metres). This accounts for 47% of our landmark observations.Additionally, we also reduce the overall confidence of the odometry links. This is needed because providing increasingly higher confidence values in the landmarks will only yield marginal changes in the overall graph once we pass a certain magnitude of values. Instead, lowering the confidence of other links in the graph can also yield the desired result.

The average accuracy can be considerably improved with this approach, as shown in table 3.4. While there is still some variance in the results with our method performing worse in some cases, on average, this approach outperforms the original approach without the EfficientPose landmarks. The variance likely originates from the low number of landmark links

traverse #	original rmse	updated rmse	delta	traverse length
1	0.295	2.179	1.884	175.627
2	18.461	15.219	-3.242	311.684
3	0.722	0.797	0.075	209.179
4	0.503	1.617	1.114	294.299
5	5.882	3.842	-2.04	223.326
6	2.107	3.358	1.251	271.119
7	2.907	5.072	2.165	370.044
8	0.385	3.027	2.642	220.805
9	1.377	4.209	2.832	304.348
10	0.158	1.772	1.614	257.601
11	3.584	4.226	0.642	236.316
12	7.613	7.255	-0.358	169.742
13	8.556	8.436	-0.12	295.943
14	12.631	13.066	0.435	189.871
15	0.396	6.185	5.789	189.259
16	0.133	3.549	3.416	169.073
17	0.715	1.33	0.615	199.123
18	1.157	3.454	2.297	273.337
19	0.418	3.616	3.198	424.867
20	3.281	7.577	4.296	819.445
21	2.592	5.663	3.071	778.842
22	6.096	8.565	2.469	866.768
23	0.803	7.238	6.435	481.025
Average:	3.512	5.272	1.76	336.158

Table 3.3: Table displaying the original localisation accuracy in comparison with the updated localisation accuracy of 23 different traverses. In this case, high confidence was placed in the position values and low confidence was placed in the orientation values.

traverse #	original rmse	updated rmse	delta	traverse length
1	0.295	0.321	0.026	175.627
2	18.461	15.496	-2.965	311.684
3	0.722	0.801	0.079	209.179
4	0.503	0.805	0.302	294.299
5	5.882	2.562	-3.32	223.326
6	2.107	2.044	-0.063	271.119
7	2.907	2.9	-0.007	370.044
8	0.385	0.596	0.211	220.805
9	1.377	1.35	-0.027	304.348
10	0.158	0.196	0.038	257.601
11	3.584	3.565	-0.019	236.316
12	7.613	7.465	-0.148	169.742
13	8.556	7.63	-0.926	295.943
14	12.631	13.922	1.291	189.871
15	0.396	2.832	2.436	189.259
16	0.133	0.449	0.316	169.073
17	0.715	1.608	0.893	199.123
18	1.157	1.164	0.007	273.337
19	0.418	0.421	0.003	424.867
20	3.281	3.277	-0.004	819.445
21	2.592	2.594	0.002	778.842
22	6.096	6.217	0.121	866.768
23	0.803	0.89	0.087	481.025
Average:	3.512	3.439	-0.072	336.158

Table 3.4: Table displaying the original localisation accuracy in comparison with the updated localisation accuracy of 23 different traverses. Here, a variable covariance matrix was used based on the estimated distance to the lander.

in the overall graph (35 on average), where even small changes in accuracy can have an large impact on the overall trajectory estimation.

traverse average	original	updated	delta
$<$ 250 metres	3.236	3.412	0.176
$>$ 250 metres	3.724	3.460	-0.264
$>$ 300 metres	4.492	4.143	-0.349

Table 3.5: Localisation accuracy with regards to traverse length.

In addition to these findings, we would also like to highlight that our pose estimation error is constant as noted in subsection 3.5.3.1, and will hence outperform relative localisation on longer traverses. In comparison to image feature-based loop closures, the proposed system is performing worse, but loop closures often only occur if we return very close to locations we have previously visited.

In addition to these findings, we observe that on average, our method performs worse than the original on shorter trajectories below 250 metres and better on longer ones as shown in Table 3.5. This is likely because on shorter traverses the visual odometry estimates are very accurate and we are more likely to worsen the overall trajectory estimate. On longer trajectories, the growing error of visual odometry makes it easier to compete. Furthermore, the loop closures that can help reduce this drift only occurs occasionally while our landmark detection is triggered more frequently.

3.5.3.3 Discussion

In this work, it has been shown that the presented method is beneficial to autonomous lunar surface operations because it can regularly update the robot pose with regard to a fix reference point, without the need to approach a fiducial marker. The detection works well at

distances up to 31 metres. While loop closure can produce better results, driving back to a known location is a time-consuming process without a guarantee for success. Additionally, our method is also robust against illumination changes occurring throughout the lunar day, which is not the case for all loop closure systems. The presented approach is especially interesting for longer traverses that return to the lander from a different direction.

Our work opens up an interesting use case for multi-robot systems relying on a global reference frame, which is shared between all robots. The primary issue such systems encounter is the need for synchronising maps between multiple rovers. In our case, we can guarantee almost identical map alignment without communication, provided that the robots are operating close enough to a known structure like a lunar lander.

While EfficientPose produces consistent results, the average orientation estimate error is still within several degrees. As this is a relatively new approach, and with only limited training time for our network, we expect that new pose estimation algorithms will be able to increase the accuracy of our localisation method. As next steps, we therefore, propose to evaluate different pose estimation networks with additional training data, including distances further than 30 metres from the lander. Higher resolution camera sensors, or cameras with a narrow field of view are also reasonable to consider with current planetary rovers in mind [133]. Lastly, our results have only been demonstrated in a simulation environment which does not necessarily reflect real-world applicability. In order to address this Sim2Real gap [93], we need to test our approach in a lunar analogue environment with a more realistic lunar lander.

3.6 Summary

In Section 3.3, we have presented a qualitative assessment on different localisation sensors, and their applications to space. As a result, we consider that given the current state of the art, stereo cameras are providing the best trade-off, a fact that is reinforced by all current planetary rovers using stereo cameras. Based on this, in Section 3.4, we have evaluated different factors which should be considered when selecting or configuring a stereo camera system for a small lunar rover.

In Section 3.5, a novel method has been proposed to extend current graph-SLAM systems by including a lunar lander as a landmark. This can be used as an alternative for loop closure systems. The method can also be applied to continuously align the map with regards to a known reference point, which is an interesting application for multi-robot systems requiring a common reference frame in order to collaborate. Instead of commonly used fiducial markers, we have made use of recently developed 6DoF pose estimation techniques, which rely on convolutional neural networks and a monocular camera. This creates an advantage in that our markers are as large as the complete lander, essentially transforming the largest object in the area into a fiducial marker. Our current results show improvements with longer traverses and we have successfully demonstrated our proof of concept. Since the applied 6DoF pose estimation is a novel approach, we expect that improvements in this area will also lead to better results for our landmark-SLAM and rival traditional loop closure systems.

In addition to the methods proposed in this chapter, one of the key takeaways from this work is that it is difficult to fully comprehend the complexity of a localisation system if we don't look at it as part of a larger system. The localisation accuracy is impacted by many design

decisions. Some, such as the mobility system, are hard to modify. Others, including driving speed, or the mounting height of the sensors are constraint by the general mission concept, and often result in trade-offs. This also became clear when working on multiple different rovers. Therefore, any such autonomy systems should have rapid development cycles which focus on completing a concise mission with a specific set of constraints. The work on the NASA robotics competition presented in Annex A is a great example of this, as it provided a good insight of what is feasible with different sensors, and how precise and robust good localisation must be in order to solve specific tasks.

4

Absolute localisation

4.1 Background

Apart from the relative localisation we explored in chapter 3, localisation with regard to an absolute reference point is also an important topic for planetary rovers. A compass is a good example of an absolute localisation system, as it provides us with a heading of the magnetic north, regardless of where we are on earth*. Depending on the mission concept, rapid absolute position estimates can be essential, for example, to plot scientific surface measurements on orbit orbital imagery or, for instance, when trying to take a surface sample of an anomaly detected through remote sensing [27]. Another example can be found in [206], where the absolute localisation on the lunar surface was required in planning the ascend and the subsequent orbital rendezvous of a sample return capsule.

Absolute localisation can also improve the quality of our relative localisation, or at the very least, bring down the precision requirement of relative localisation solutions. This is because absolute localisation estimates, while experiencing high error rates, have constant error rates for every measurement. Relative localisation on the other hand, is often much more accurate between single measurements, but as their error is cumulative, it will eventually outgrow the error of any absolute localisation system. While our robotic systems do and should rely on relative localisation systems, we should make use of absolute localisation systems to correct the relative drift in the pose estimation on a regular basis in order to prevent unbounded error rates.

On Earth, absolute localisation in outdoor environments can be considered a solved problem for most applications due to the availability of Global Navigation Satellite System (GNSS)

*with some limitations closer to the poles

receivers. This is however, a problem which has not been solved for deep space missions. Once we leave direct vicinity of our earth, we can no longer rely on GNSS satellites for absolute localisation. Instead, other methods need to be considered, both for orbital and surface operations. In this chapter, we will first explore existing methods for absolute localisation, before presenting two new machine learning approaches, as well as the datasets generated to train them.

4.2 State Of The Art

Absolute localisation for planetary robotics is a relatively young topic. Most applications are related to operating satellites, as mankind has had decades of experience with orbit determination of satellites travelling through space. While some of these methods can be applied to surface operations, the accuracy required to safely operate a satellite is often lower than what we would expect for surface operations. Surface localisation has only really become a topic with the first planetary rovers landing on Mars.

4.2.1 Global Navigation Satellite System (GNSS)

For earthbound applications, the most common method to turn to when highly accurate localisation is required is GNSS. This approach relies on a receiver acquiring a signal and being to able to calculate a range estimate to at least three satellites. As the approximate location of each satellite and the distance to each satellite are known, the receiver's position can be estimated through trilateration or multilateration (as opposed to triangulation, the measurement of angles). This approach can be used for outdoor applications [142], as well as satellite

operations [86, 114]. The accuracy of such systems depends on a range of factors, including the number of visible satellites and the availability of single or multi-band receivers to better estimate the atmospheric disturbances. For more accurate measurements, the use of a fixed ground-station close to the rover can be used to improve the accuracy. As the position of the ground-station receiver is known to be fixed and exposed to similar disturbances, one can estimate the error offset, and remove it from the rover receiver data. This method is also known as differential GNSS (DGNSS) or differential GPS (DGPS).

Unfortunately, there are currently no operational GNSS satellites constellations around Moon or Mars. For lunar applications, it could be technically feasible to detect GNSS signals from earth in a lunar orbit, or the lunar surface. The distance to earth puts some limitations on the achievable accuracy, which lies somewhere between 100 [49] and 200 metres [58]. This accuracy however, has never been demonstrated and relies on larger and highly sensitive GNSS receivers. As such, we do not consider GNSS as a feasible solution for private entities localising on or around the Moon in the near future.

4.2.2 Radiometric Measurements

With interplanetary satellites, the most commonly used methods for tracking are radiometric doppler and ranging. The accuracy of these methods depend on a number of factors, including the distance between the ground-station/relay and the satellite, type of antenna, the total measurement time and the communication frequency used [195]. Additionally, Very Long Baseline Interferometry (VLBI) can be applied to further increase the accuracy through the simultaneous use of multiple ground stations. However, these radiometric ranging methods are not applicable to our rover use case, as this would require our rover to have direct to earth

communication with a powerful receiver. As we are using the rover's lander as a relay through an omni-directional WiFi antenna, a pose estimation between lander and rover is limited to a distance measurement. Nevertheless, this method could be applied to determine the lander's position, as it is equipped with direct-to-earth communication. As shown in [206], this can result in a position estimation accuracy of 100 metres. If the lander is communicating via a relay satellite, even more accurate ranging could be estimated as shown in [70].

The downside of these radiometric methods is however, that they require multiple days of data, time and data-bandwidth which may not be available for certain mission concepts. Aside from radiometric ranging, Laser ranging can also be considered. This has been used successfully for the Lunar Reconnaissance Orbiter (LRO) [222, 135]. The estimated accuracy of this method for determining LRO's lies within 200 metres, with some outliers [184] (shown in figure 4.1). This was achieved through a combination of Doppler, ranging, laser ranging and star trackers. Since location and distance from earth are similar to LRO and as there are no atmospheric disturbances on the Moon, we can expect a similar position accuracy for our lander, given the same sensor suite.

4.2.3 Star Trackers

Star trackers are a well-established method for absolute orientation estimates in space. In simple terms, a star tracker is a highly sensitive camera, with a long baffle* to avoid illumination from unwanted sources, such as our Sun or other parts of a spacecraft. It then compares the pattern of detected stars against a database to determine which part of the sky the camera is looking at. The basic principal has been used for centuries by navigators at sea. As the Moon

*A baffle is a mechanical system, whose function is to shield the light coming from sources outside the field of view (FOV) of the camera. [171]

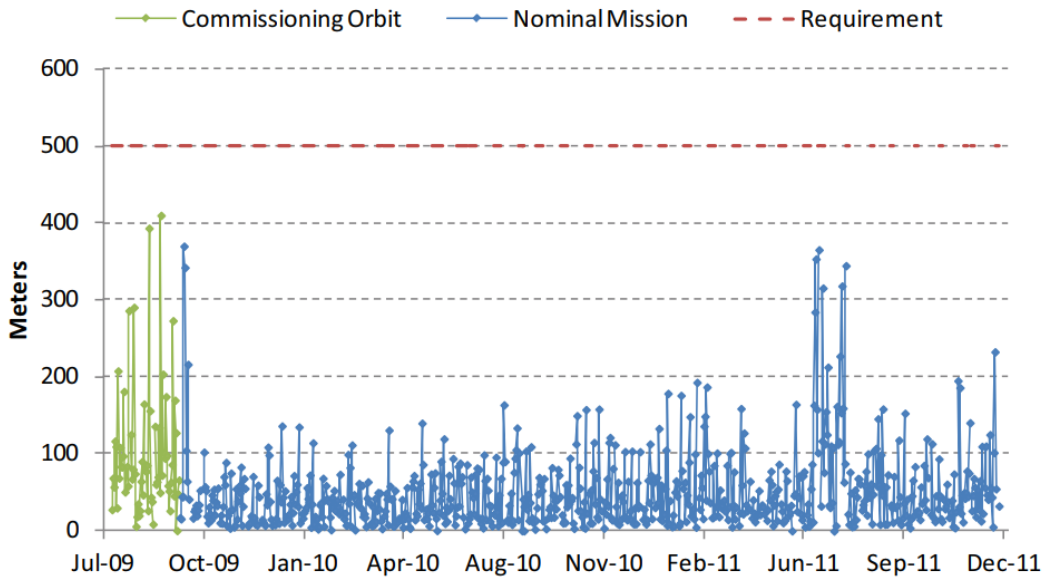

Figure 4.1: RMS total position definitive accuracy, early commissioning orbit and complete nominal mission orbit. This figure is taken from [LUNAR RECONNAISSANCE ORBITER ORBIT DETERMINATION ACCURACY ANALYSIS, S. Slojkowski] and used with permission of NASA [184]

does not have an atmosphere, the stars are visible throughout the lunar day. With the exception of times when the star tracking is pointing too close to our Sun, this approach can also be used to estimate the rover's orientation. The upcoming Viper mission is an excellent example in their use of star tracker [27]. Current star trackers can deliver a heading estimate within an accuracy of less then 10 arcseconds [54] and at an update frequency between 0.5 and 10 hz [119].

The map of the sky that the star tracker provides can be used with two separate approaches for a full 6 DoF pose estimation. The simple approach can support existing odometry methods by providing accurate orientation estimates. This can significantly improve the pose estimation, as even small relative orientation errors lead to large relative position errors over time as shown in [63]. While this example shows good results in theory, it would benefit of further

research, as the results are only compared to a pure wheel odometry (WO) approach, which is known to yield low orientation accuracy, especially when used on a differential drive rover platform.

Another approach is to produce a full pose estimation with star trackers alone. Provided accuracy is based on accurate ephemeris data and therefore relies on an accurate on-board clock. The resulting accuracy is also limited, as shown in this theoretical example, where a position accuracy of 200 metres is being mentioned [181]. While this seems significant, it can still be useful for longer traverses or to determining the position of a landing site within a larger landing ellipse. While these methods could also be applied to Mars rovers, they are less useful in this case. As Mars has an atmosphere, star trackers can only be used to their full potential during the night, when the stars are fully visible. Besides a heading estimation, a full pose estimation would also be far less accurate, as this estimate varies with the radius of the planetary body.

4.2.4 Sun Tracker

Beside Star trackers, a more simple approach can also be to directly use our Sun as reference point. Special sun sensors exist [185], which can detect the direction of the sun within 0.5 degrees accuracy. In turn, this can then be used to estimate the rover heading as described here [88]. Such a sensor however, even if extremely light, takes up precious resources in rover space and development for a limited detection accuracy. Due to the restricted field of view, multiple such sensors are also needed for certain use cases. Instead, approaches which make use of already existing cameras on the rover should be considered to detect the sun and thereby estimate the rover heading. With the help of an accurate on-board clock, an ephemeris model of

our solar system, an IMU, and a rough estimate of our position on the Moon, we can estimate the rover's heading. The estimation process is composed of four parts. Firstly, the x, y coordinates of the sun need to be determined in a 2D image, which can be achieved with computer vision. Secondly, the direction of the sun relative to the rover can be estimated based on the camera intrinsics and extrinsics. Thirdly, we estimate the sun direction with respect to the gravity vector from the IMU, as if the rover was on flat ground. Lastly, we estimate the rover's absolute orientation based on the ephemeris data, and where the sun should be based on our current rough location on the Moon. The first three steps have been demonstrated in [42] by using simulation data and lab testing. During lab experiments with some ground truth measurement uncertainty, the mean error lies between 7.9 and 39.0 degrees, depending on the dataset used. While the lab results are less promising, the simulation results show that the approach can provide an accuracy of around 1 degree under ideal circumstances on a wide angle lens (150 degrees FoV). While it is clear that a star tracker is a more suitable sensor in this case, the sun tracker approach is still promising given the hardware limitation of microrovers. That being said, additional research is still necessary to more reliably assess the accuracy of this method under realistic conditions and with different FoV lenses. This approach should also be considered for larger rovers as redundancy concept for star trackers, eliminating the need for duplication of critical hardware.

4.2.5 IMU & Compass

In recent years, many odometry solutions have shown that inexpensive 9 degrees of freedom IMU units can be used to increase the accuracy of odometry estimates and SLAM. The most common approach is to include the IMU's accelerometer and gyroscope in direct pose esti-

mations as shown in [159, 158, 66] (as mentioned in subsection 3.3.2) . We can, however, also look at the absolute measurements, such as the gravity vector which can be deduced from the accelerometer. Since the gravity vector always points towards the centre of the planetary body, we can use this information to correct the orientation of our pose estimation . The gravity vector will, however, only give us a pitch and roll direction. Heading or yaw estimates can be calculated from a compass for terrestrial applications. On both Moon and Mars however, the magnetic fields are not useful for this application [62]. The Moon does not have a global magnetic field [207] which would be necessary for compass usage. Additionally, the Moon's magnetic field is also much weaker than what we are used to here on Earth [157]. Therefore, for localisation purposes on the Moon or Mars, only 6DOF Inertial Measurement Units with an accelerometer and gyroscope can be used. This is also what makes Sun or star tracker especially interesting for planetary robotics, as they can help estimate the yaw angle that a magnetometer provides here on earth.

4.2.6 Detection In Satellite Imagery

Currently, the most common method used for estimating the position of a lander or rover on the lunar surface is direct satellite imaging. This method is generally used as ground truth measurements for other localisation approaches [121]. To achieve this, an orbiting satellite is required to pass over the desired location under ideal lighting conditions. As currently only a limited number of lunar orbiters are equipped with high resolution cameras (LRO [164] and Chandrayaan-2 [26]), it can be a matter of weeks before an image can be taken, often exceeding the mission lifetime of a rover. In the resulting images, landers (Apollo landers [96]) as well as larger rovers (Lunokhod-1 [94]) are directly visible . Photometric anomalies

have also been observed in orbital imagery for the trajectories of vehicles such as the Apollo-era Lunar Rover Vehicle (LRV) or Chang'e 3, resulting from the regolith being disturbed during the traverse [96, 95, 121].

4.2.7 Descent Imagery Localisation

If the mission concept does not allow waiting for an orbiter flyover, a lander can also be localised through descend imagery. For this process, an image is taken at a high altitude matched with a georeferenced image from an imaging orbiter like LRO. The final landing position is then estimated by taking the sequence of descend images and estimating the pose update from the previous image to the current image, until the lander is on the ground. As shown in [197], an accuracy of 30 metres can be achieved with this method within a time-frame of 30 minutes after landing. Still, this method does require a high data bandwidth to downlink the imagery, making it less suitable for Mars missions.

4.2.8 Satellite Imagery Matching

As rovers do not have any descend imagery, another approach is to match surface perspective images from a rover directly with orbital imagery. This technique has the advantage in that it can be achieved regardless of orbiter availability, as long as orbital imagery has been obtained beforehand. As LRO imagery almost covers the entire lunar surface, even providing multiple high resolution images for many locations, this makes it the most attractive source of information for absolute localisation.

4.2.8.1 Manual Matching

For current Mars missions, orbital and surface perspective imagery are matched manually. This process is tedious and slow (30-60 minutes [T. Parker, personal communication, 9 August 2018]), and requires a downlink to earth. As described in [71], this process relies on finding recognisable features (e.g., hills, craters, larger boulders,..) in surface perspective imagery, which can also be seen in the orbital imagery. Through triangulation, a rough position estimate can then be calculated ($\sim$ 100m according to [25]). As described in the example, this process is ideally performed on a panorama picture which covers 360°, in order to include features in all directions. The accuracy of this method is however limited by the resolution of both image types, as well as the distance to the features. To increase the accuracy, this can then be extended by manually matching a reprojected surface image to an orbital image, until the two overlap ([T. Parker, personal communication, 9 August 2018]).

4.2.8.2 Automated Matching

When matching surface perspective images to orbital imagery, the distortion which appears when attempting to match an image from the surface to an image taken at a 90 degree angle from a satellite, is the biggest constraint. In [178] this is addressed, by equipping a rover with a flash LIDAR, allowing for the projection of surface images onto accurate surface geometry. With the help of this automatically generated textured surface geometry, accurate orthographic reprojections can be produced of the rover's surroundings. The resulting images are then directly matched to satellite images with an error rate of less than 2 metres within a 300 $\times$ 300 metre search window. The results are however produced in simulation with limited information about the rendering setup. Computer vision algorithms generally yield

better results in simulation than on real data, due to the lack of noise. A similar LIDAR device also has never flown on a planetary rover, although the use of a stereo camera would also be possible, but at the expense of reprojection accuracy. The research does, however, confidently address variations in illumination conditions between surface and orbital imagery which would be problematic on a real mission.

4.2.8.3 Rock And Feature Matching

Direct matching of rocks distributions has been shown to perform well in [37, 84, 126], even with sub metre accuracy. One of the methods described detects rocks in both satellite and surface perspective imagery, and then tries to randomly match the rock distribution, minimising the number of non overlapping rocks. Another method applies direct image based feature matching in locations with distinct surface textures, such as outcrops. The authors however even state themselves that they rely on the availability of good features which can be used to match to satellite imagery, thus limiting the locations where this can be applied. This is especially tricky around lunar landing sites, which are generally picked for being free of obstacles and thus devoid of features. As the work is applied to the MER rovers, high resolution satellite imagery (0.25 metres per pixel) is also available. As the best imagery with reasonable coverage on the Moon has a resolution of 0.5 metres per pixel, far fewer recognisable features will be available for matching. Additionally, the research has mostly been applied to locations with almost flat surface geometry. Lastly, the method required a good amount of computational power, making it difficult to process directly on a rover.

4.2.8.4 3D Terrain Matching

As Digital Elevation Models (DEM) are available for both Mars and Moon at a high resolution ($>$ 1 metre per pixel), they can be considered for direct matching with 3D surface geometry. The advantage of this approach is that it avoids the problem of distortions which occurs when attempting to directly match surface and orbital imagery.

As shown in [20], LIDAR data can be used for this type of matching, and with the help of geometric feature detection, computational cost can be kept to a minimum. The method does however, rely on matching enough good features. This can be a problem depending on the amount of variation in the terrain, and how much of the surrounding area is occluded by the shape of the terrain. The data from a Mars analogue and a 13 metre per pixel resolution DEM shows that the proposed method can yield results around 20 metres with outliers of up to 80 metres. The primary issue with this method lies nonetheless with the LIDAR sensor itself, which to date, has not flown on a planetary rover due to the mechanical complexity and power requirements.

A different matching use case is shown by [84], where a high resolution DEM (1 metre/pixel) is matched with a DEM from the stereo camera of the rover. The matching is achieved by treating the DEMs as images and directly matching the attitude intensity. The achieved accuracy lies at 90 metres. This could probably be improved as the DEM matching is not the primary goal of this work. Instead, the described matching process is only applied to limit the search space for a rock matching approach.

4.2.8.5 Horizon Line Matching

A simpler way to match DEM with surface imagery is to only look at the horizon line. For this purpose, we look at the *Visual Position Estimator for Rover* (VIPER) algorithm as proposed by [31]. The orientation is considered to be known in order to reduce the search space, and panorama images have been used to increase the possible features to match. From the images, the algorithm detects mountains and then searches for mountain peaks in the topological map. With terrestrial data, the method achieves an accuracy below 100 metres in a mountainous regions. In a more recent example by [25], the same method has been applied to imagery from Mars, where the MER rovers have been localised within 27 and 51 metres on Mars. These results however, have been achieved under ideal conditions, at the edge of a crater, where the added crater features provide a very distinct horizon line.

Figure 4.2: Examples of automated horizon line detection (red). Source: [10]

A variation of the horizon matching approach can also be found in [10] (shown in figure 4.2). In this case, the horizon line itself has been extracted and matched to a dataset of lines horizon lines which were previously generated from the DEM. Although, in this example, the results have been less accurate, this is possibly due to the much larger search area, and the use of single images instead of panoramas. Generally, the accuracy of this method will always be limited by the large distance to the features used for matching. Additionally, the method

is unlikely to perform well on flat terrain.

4.2.8.6 Terrestrial Methods

For additional state of the art methods, we look towards terrestrial examples, where we can rely on GNSS ground truth measurements for validation. Here, localisation based on satellite imagery has already been demonstrated [197, 13], especially with a focus on urban areas with abundant and distinct features, such as roads and buildings. Despite this, few terrestrial studies have focused on natural unstructured environments, which would be applicable to the lunar surface. This is primarily due to the limited availability of geo-referenced ground imagery in these areas. The preferred data source for most of these studies is *Google Streetview* * because it has been georeferenced and provides a 360 degree view. Nevertheless, other datasets from study sites exist, which have included terrestrial lunar/martian analogues (Devon Island, Canada (e.g., [113, 61]); Atacama desert, Chile (e.g., [209])). Unfortunately, most of them are low in resolution, have limited coverage and and are relatively small in size. Especially the dataset size is a limiting factor for machine learning applications where data volume is critical.

As discussed in [16], most of the presented terrestrial methods pose some issues for the planetary rover use case. Besides the mentioned repeatibility and openness of the results, not all of the methods are feasible on a planetary rover. Additionally, most methods lack the validation in environments similar to the lunar surface.

*www.google.com/streetview/

4.3 Orbital Map Matching With Reprojected Surface Perspective Imagery

In this section we propose a new method to perform absolute localisation on other planetary bodies. As described in section 4.2, the current state of the art contains a wide range of methods for absolute localisation. However, as GNSS systems are not available for Moon or Mars, this is especially challenging for our use case. Additionally, a number of presented approaches are not applicable as they either require too much time (radiometric approaches [206]), or require additional sensors which have never flown on planetary rovers due to engineering challenges (LIDAR [178]). Therefore, the most promising methods are camera based, which compare already existing orbital imagery with surface perspective imagery (horizon line matching [31], terrain matching [84]). Some methods have successfully shown image feature matching can be achieved when enough recognisable features are available (outcrop feature matching [37]). Here, we believe that current methods can be improved through the use of machine learning in order to successfully localise in a wider range of scenarios. Terrestrial approaches have shown that machine learning methods can be used to match images from completely different points of view (terrestrial matching [197]). To the best of our knowledge, these methods are promising, they have not been applied to the unstructured environments that we deal with for planetary surface robotics.

We propose to compare reprojected surface imagery with orbital imagery, through the help of a neural network. We limit our assumptions to currently available technology: high resolution imagery available through orbiters (LRO [164]) or descent imagery (Chang'e 3 [121]);

a surface rover with a single camera on top of a mast which can be rotated through turning the mast or the rover in order to take images in all directions (Perseverance [133]).

We take 4 surface images which are then reprojected and stitched on a flat surface to make them comparable to the orbital data. The resulting image, while heavily distorted, can then be compared to a number of satellite images from the same location with the help of a neural network. With this method, we iterate over the complete satellite image of the area, until we find the location with the highest likelyhood of a match. In order to overcome the previously described issues of limited availability of geo-referenced surface perspective, we use a virtual synthetic lunar environment to train and evaluate our approach. (This work has previously been published in [216] and was done in collaboration with *Ben Wu*, *Ross Potter*, *Andrew Chung* and *Timothy Seabrook*)

4.3.1 Methodology

4.3.1.1 Data Generation

With the following method being a machine learning approach to localisation, one of the important aspects we need to solve first is the availability of enough location-referenced training data. Current datasets (e.g., Apollo mission, martian rover traverses or available datasets from terrestrial datasets) only provide a limited number of photos and with varying degrees of ground truth position accuracy. The primary limitation here was the size of the available datasets, as large amounts of data are preferable for machine learning approaches (100.000 samples or more, although not impossible with smaller datasets). We considered two different options: 1) either to create our own dataset in a terrestrial lunar or martian analogue or 2) using a synthetic virtual planetary environment. For both of these options, adequate ground

truth information can be collected, and orbital imagery can either be generated or is already publicly available. As real data is always preferable, we initially started working on our own dataset by sending a stereo camera and a GPS receiver to a martian analogue expedition on Devon Island (Canada). Due to the timeline of our project and the expedition, we decided to focus on our second option with the expectation of using the expedition dataset for validation instead. Unfortunately, the expedition did not provide any useful results in the end, as it was plagued by bad weather and our data collection had to be dropped (figure 4.3). While the virtual environment was not our first choice due to the sim2real gap, it does have the advantage in that it eliminates most effects which could influence the results in a positive or negative way, due to the full control we have over our simulation environment.

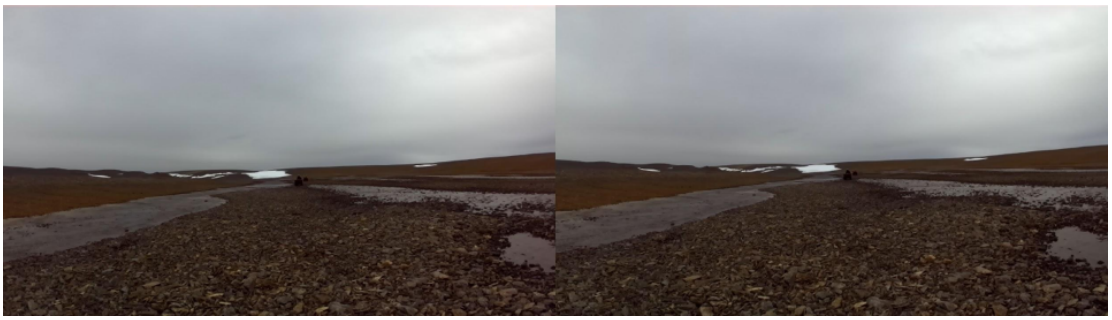

Figure 4.3: Stereo images from Devon Island expedition before the traverse vehicle got stuck in the mud.

The dataset we have generated is comprised of planetary surface (i.e., rover-perspective) images and corresponding satellite images. The synthetic test environment was built using an existing Moon Landscape* v3.0 and Unreal Engine† 4, a free‡, real-time game engine (described in more detail in section 2.2.5.1). The data was generated on a powerful desktop computer and later uploaded to a server for pre-processing and training. Notably, other studies

*https://www.unrealengine.com/marketplace/the-moon
†https://www.unrealengine.com/
‡for non-commercial purposes

have also relied on generated data using rendered lunar terrains (e.g., [178]).

Within the 8 km × 8 km synthetic lunar landscape, three distinct regions (figure 4.4) were used for dataset generation: a training zone (2.05 km × 2.05 km), a validation zone (1.05 km × 1.05 km), and a testing zone (1.05 km × 1.05 km). The zones were chose such that each one of them would provide both flat and cratered regions (figure 4.5) while being far enough from the border to avoid generating images showing the edge of the world.

Unreal Engine allows for perfect control over multiple parameters such as sun angle, lighting conditions, and rock and crater distributions within the synthetic environment. In our work, a single sun angle of 30° above the horizon was chosen to avoid complications arising from challenging lighting conditions. While future work should consider different illumination conditions, roughly matching the time of day on the surface with satellite images should be feasible as there are multiple orbital images available for most locations on the Moon. Environmental features, such as craters and rocks, can also be added following a realistic distribution and at cm-scale resolution. The original environment we used was adjusted accordingly to match a more realistic rocks distribution. Though Unreal Engine has the capability of ingesting other datasets (such as lunar and martian DEMs), only the synthetic test environment of Moon Landscape v3.0 was considered for this proof-of-concept.

An automated pipeline was built within Unreal Engine to place the rover at random locations within the synthetic lunar environment, capturing four ground perspective images (front, left, rear, right, spaced 90° apart) with minimal overlap. Figure 4.6 displays a set of such images taken at one location within the lunar environment. The camera, which had horizontal and vertical fields of view of 90° and 50.6°, respectively, was placed 2m above the surface and tilted downwards by 15° to simulate the rover camera height and orientation.

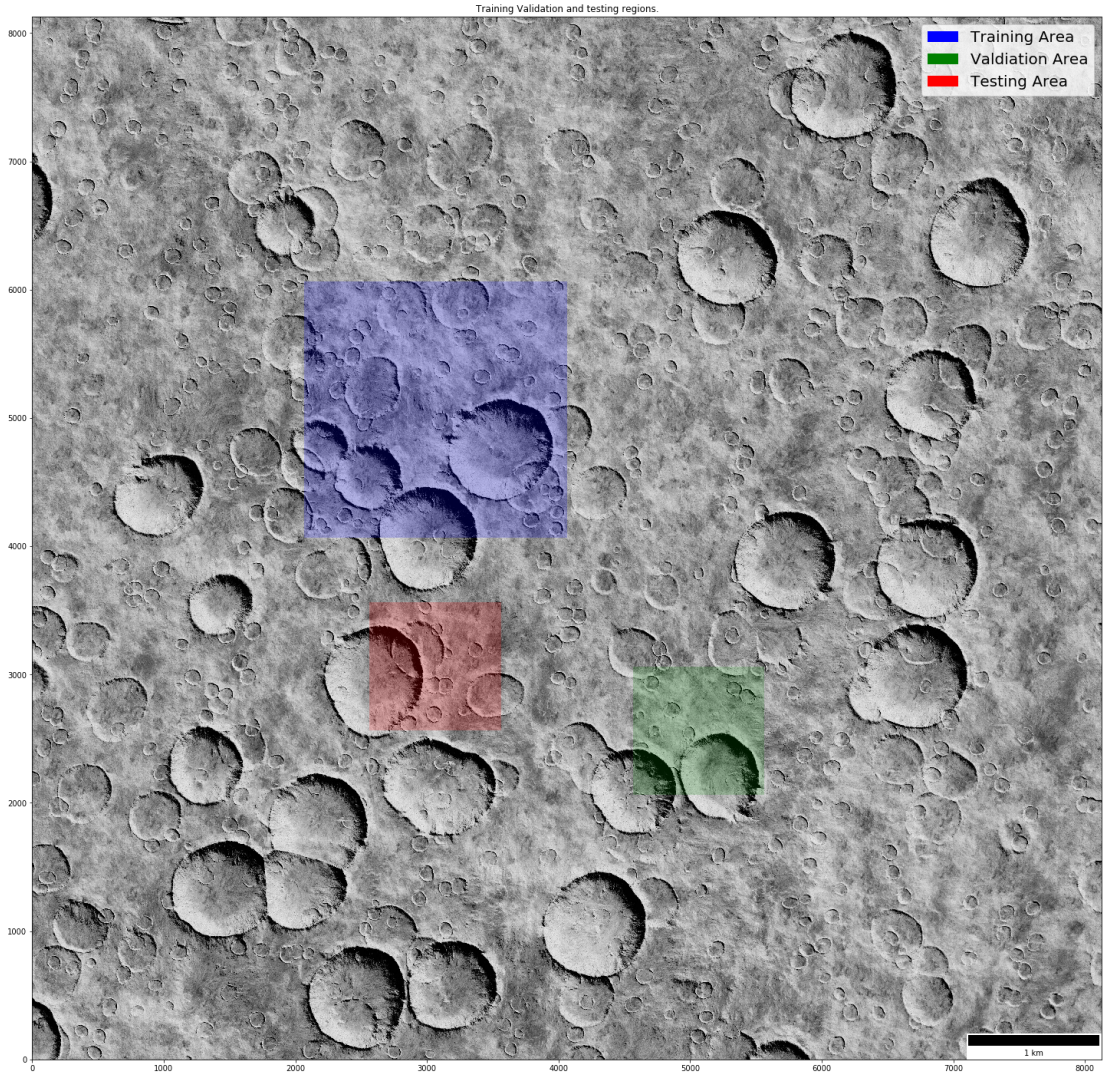

Figure 4.4: Complete map of the $8\ \mathrm{k}m \times 8\ \mathrm{k}m$ environment. The different zones are highlighted in colours for the training, validation and testing datasets.

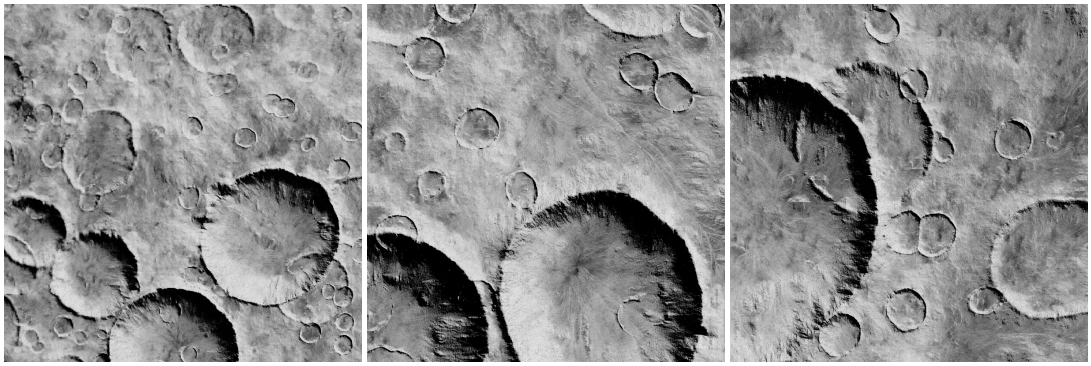

Figure 4.5: The 2.05 km × 2.05 km training zone (left panel), 1.05 km × 1.05 km validation zone (middle panel), and 1.05 km × 1.05 km testing zone (right panel) within the synthetic lunar landscape.

The mast height was selected based on the Exomars rover configuration [183]. The camera tilt angle was selected to maximise the ground view in the images and to avoid capturing large parts of the sky, which does not provide usable features for localisation (assuming that the cameras are exposed for the lunar surface and not the stars). We assumed that the complete rover orientation is known and therefore, the images are always taken in the same direction. Pitch and roll angles can be inferred from an inertial measurement unit (IMU), while the yaw angle could be from star trackers (e.g., [30, 47, 63], accurate to arcsecond-scale [119, 47]), or via sun position and time of day (e.g, [204, 44, 6]) for martian applications. This assumption significantly reduces the matching search space from five degrees of freedom (DoF) down to two, as we now only deal with the horizontal translation coordinates. For each location, coordinates, orientation, sun angle and image size were recorded to a text file. The images were extracted from the Unreal Engine using the Fraps* video capture tool, which enabled images to be saved to disk at 60 frames per second. In order to match the images to the metadata, ID numbers were encoded into the first row of the images, as the top edge of the frames were

*http://www.fraps.com/

not used for our approach. This approach was chosen because directly recording and saving images to disk inside Unreal Engine is significantly slower and only at the order of several images per second. The final output was converted to grayscale before further processing as our environment does not present any colour. The satellite images were collected separately by iterating over the complete environment while taking top-down images. These images were then stitched into several large images, from which the required orbital images were extracted based on the requested coordinates.

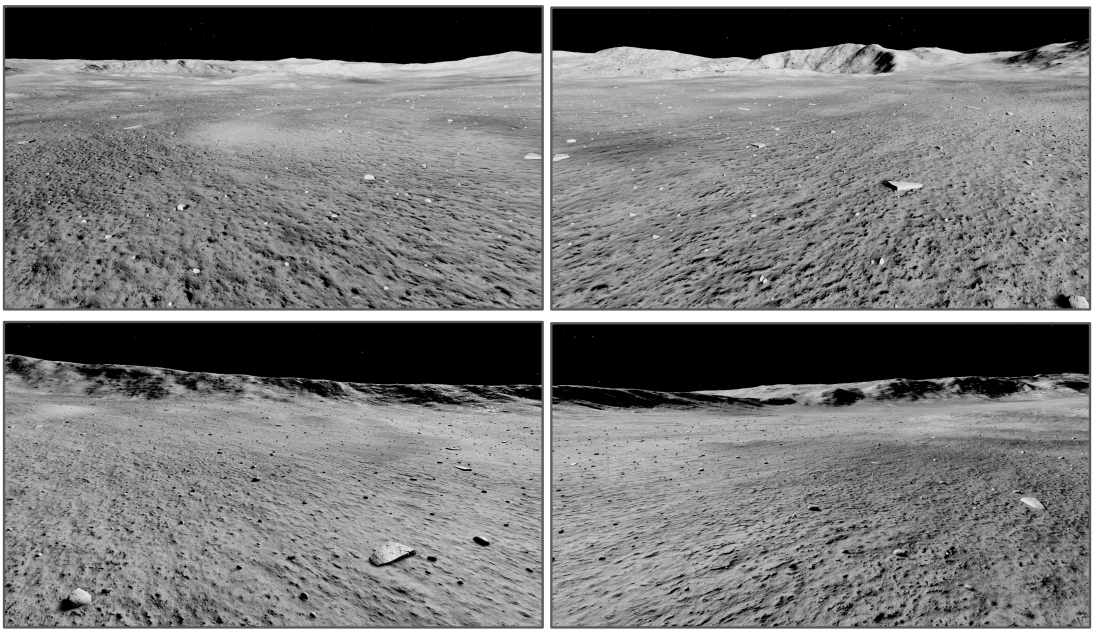

Figure 4.6: Example ground view images taken at one location within the testing region. Clockwise from top left panel: front view, right view, rear view, left view.

4.3.1.2 Data Processing

Each set of four ground perspective images was processed into a pseudo-aerial image using custom Python scripts utilising the OpenCV library. Specifically, each rectilinear ground

perspective image was cropped to remove areas further away than 25 m ahead of the rover location, approximating the surface as a locally flat plane. The remaining nearby landscape was reprojected using the camera matrix to form one quadrant of an equivalent top-down view. The quadrants were then smoothly stitched and scaled into a representative 50 m x 50 m aerial image with a pixel resolution of 0.05 m.

Figure 4.7 shows the image obtained by reprojecting the views taken in Figure 4.6. The respective ground truth satellite image is shown alongside for reference. As the ground perspective images were reprojected onto a flat plane, distortions are inevitably introduced. Deviations from the satellite image depend on the topography and can range from minimal for open terrain to significant near craters and sharp elevation changes. While 3D re-projections were considered, they were not used due to the added complexity, as well as the additional data collection requirements. These images were downsampled from 1000×1000 to 224×224 (0.22 m resolution) to be used as input to our neural network. This reduced resolution is still comparable to the best available satellite images for Mars (0.25 metres per pixel (m/px): Mars Reconnaissance Orbiter High Resolution Imaging Science Experiment [137]) and higher than that for the Moon (0.5 m/px: Lunar Reconnaissance Orbiter Narrow Angle Camera [164]). Even higher resolution imagery would be expected during the landing phase of a mission, as highlighted by the Chang'e-3 spacecraft, which collected images at 0.05 m/px around its landing site [123], further aiding localisation accuracy. High resolution images comparable to those captured from landers and rovers may also be obtained via aerial vehicles (e.g., Mars 2020 helicopter [11]). Thus, neural networks utilising higher resolution inputs may be explored in future work.

To build the training dataset, all reprojected images were paired with a satellite image from

the training zone. The pairing was performed in such a way that in 50% of the cases, the satellite image matched the location of the ground view image exactly. For the remaining 50% of the dataset, a non-overlapping random satellite image with the same physical scale was paired with the ground view image.

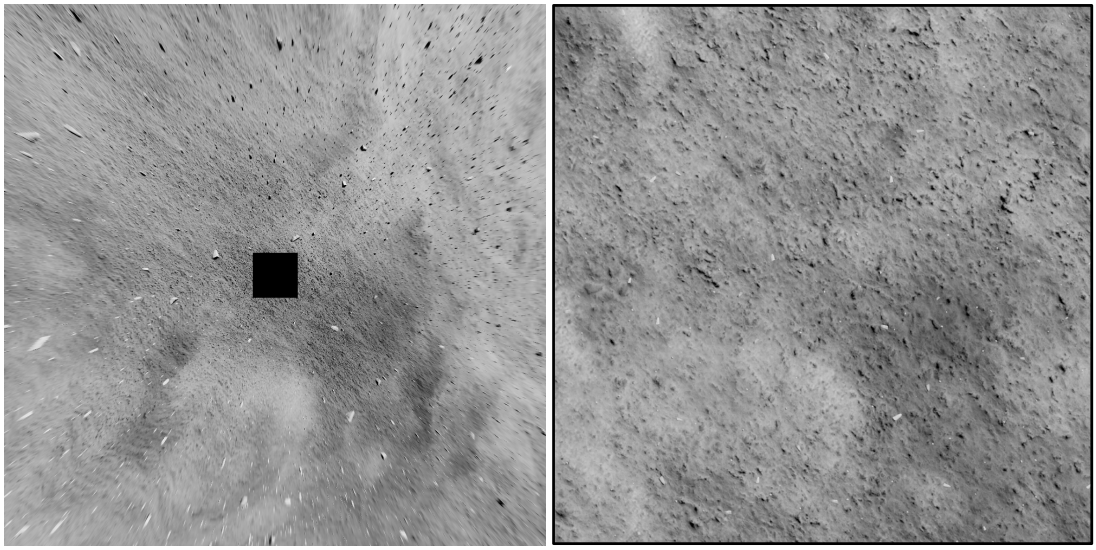

Figure 4.7: (Left) Example of an aerial reprojection using the ground views in Figure 4.6. The black square encompasses the camera position and indicates regions near the rover not imaged due to the limited vertical field of view. (Right) The corresponding ground truth satellite view. In both images, the region represented is $50\ \mathrm{m} \times 50\ \mathrm{m}$.

4.3.1.3 Neural Network

The labelled (matching/non-matching) image pairs were used to train a neural network to identify matching pairs of reprojected surface-perspective images and satellite images. The resulting model, Planetary Localisation Neural Network (PLaNNet v0), is depicted in Figure 4.8 and described below.

PLaNNet is a Siamese neural network. Each head of the network feeds into a pre-trained

50-layer ResNet v2[77, 78] feature extractor*. The pre-trained weights of the feature extractor come from training on the ILSVRC-2012-CLS image classification dataset, and were not modified during the training of our network. Each head takes a 224×224 px RGB image as input, with intensity values in the range [0, 1]. As we use grayscale images, the single channel is replicated across red, green, and blue channels before feeding into the network. Although inefficient, this allows the use of pre-trained weights, which was critical to the success of this project within the time constraints of NASA Frontier Development Lab. Each feature extractor outputs a one-dimensional feature vector of length 2048. These two feature vectors are concatenated into a single 4096-element one-dimensional vector. This is fed into a 256-neuron fully connected layer with 30% dropout probability, used to produce the final match/no-match logits vector. Softmax is applied to the logits to produce the match/no-match probability distribution for a pair of input images.

Thus the trained network, given unlabelled image pairs, outputs the probability that the two image locations "match". This is a classification problem, and so we use cross-entropy loss. The probability of the pair being a match was used as the similarity score.

4.3.1.4 Localisation

Localisation of a given position was performed by comparing the corresponding reprojection query image with a series of satellite-view candidate images. The array of candidate images was extracted via a sliding window which sampled 50 m$\times$50 m regions of interest within the 1.05 km$\times$1.05 km testing zone at regular spatial intervals. At each window position, the enclosed 50 m$\times$50 m ground truth satellite image received a similarity score, with respect to

*https://alpha.tfhub.dev/google/imagenet/resnet_v2_50/feature_vector/1

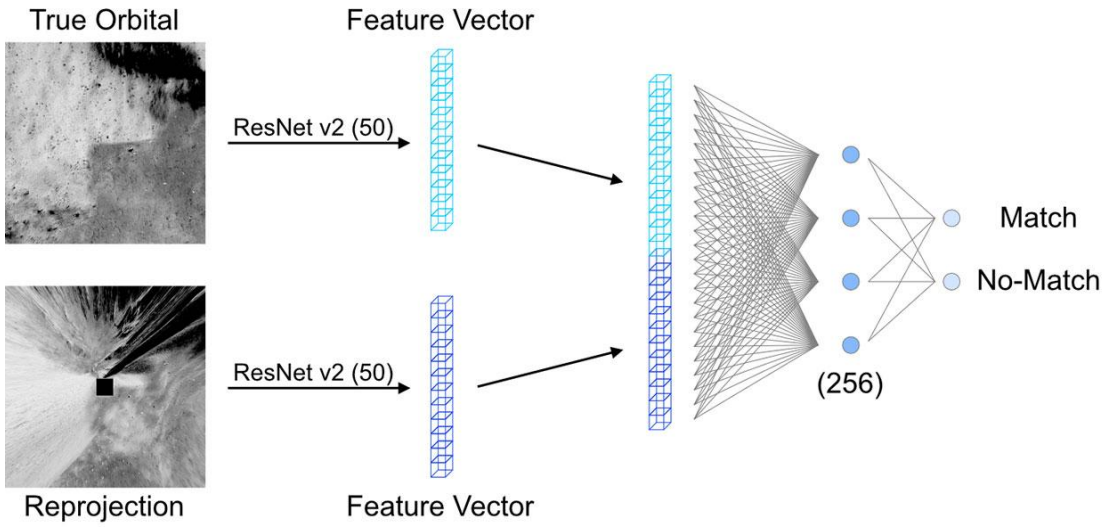

Figure 4.8: Schematic illustrating the PLaNNet v0 architecture.

the query, via the neural network. All candidates were then ranked based on similarity score to determine the top inferences for absolute location.

4.3.2 Results and Discussion

4.3.2.1 Dataset Generation

In total, 2.4+ million surface-perspective images corresponding to 600,000+ distinct locations were generated within the synthetic environment, which was itself divided into separate training, validation, and testing regions. For each location, 4 specific outputs are produced: (1) a set of 4 surface-perspective images, (2) metadata for the location and camera, (3) a processed top-down reprojection view, and (4) the extracted ground truth satellite image. The surface images were captured in 1920×1080 px resolution and reprojected to form 1000×1000 px images with 0.05 m/px resolution, representing a physical area of $50\,\text{m} \times 50\,\text{m}$.

The corresponding ground truth satellite image also covers 50 m×50 m and is 1000×1000 px. These data types together comprise our full dataset, the Lunar UNreal Assets (LUNA) Localisation Dataset. The dataset is approximately 10 TB and its details are shown in Table 4.1. The breakdown of the training, validation, and testing dataset splits is shown in Table 4.4. The LUNA Localisation Dataset and project source code are publicly hosted online*.

Table 4.1: Summary of the LUNA Localisation Dataset

Item	Physical Scale	Resolution	Quantity
Training Region	2.05 km× 2.05 km	41000× 41000 px	1
Validation Region	1.05 km× 1.05 km	21000× 21000 px	1
Testing Region	1.05 km× 1.05 km	21000× 21000 px	1
Surface Images	90°×50.6°	1920 × 1080 px	2.42×10^6
Satellite Images	50 m × 50 m	1000 × 1000 px	6.06×10^5
Reprojected Images	50 m × 50 m	1000 × 1000 px	6.06×10^5

4.3.2.2 Localisation

Figure 4.9 illustrates results using our localisation method on three randomly selected positions. In these cases, we first performed localisation testing in a 300 m×300 m sub-region within the aforementioned 1.05 km × 1.05 km testing region, using a sliding window step

*http://moonbench.space/

Figure 4.9: Localisation results for three positions are shown, indicated in the upper, middle, and lower sets of panels, respectively. In each set, the blue panel displays the top-down reprojected ground image (50 m x 50 m), the green panel displays the matching ground truth satellite image (50 m x 50 m), and the red panel displays the testing environment (300 m x 300 m). The ground truth location is highlighted as a solid grey square within the testing environment, while white square outlines show the neural network's top N=5-10 matches to the reprojected image.

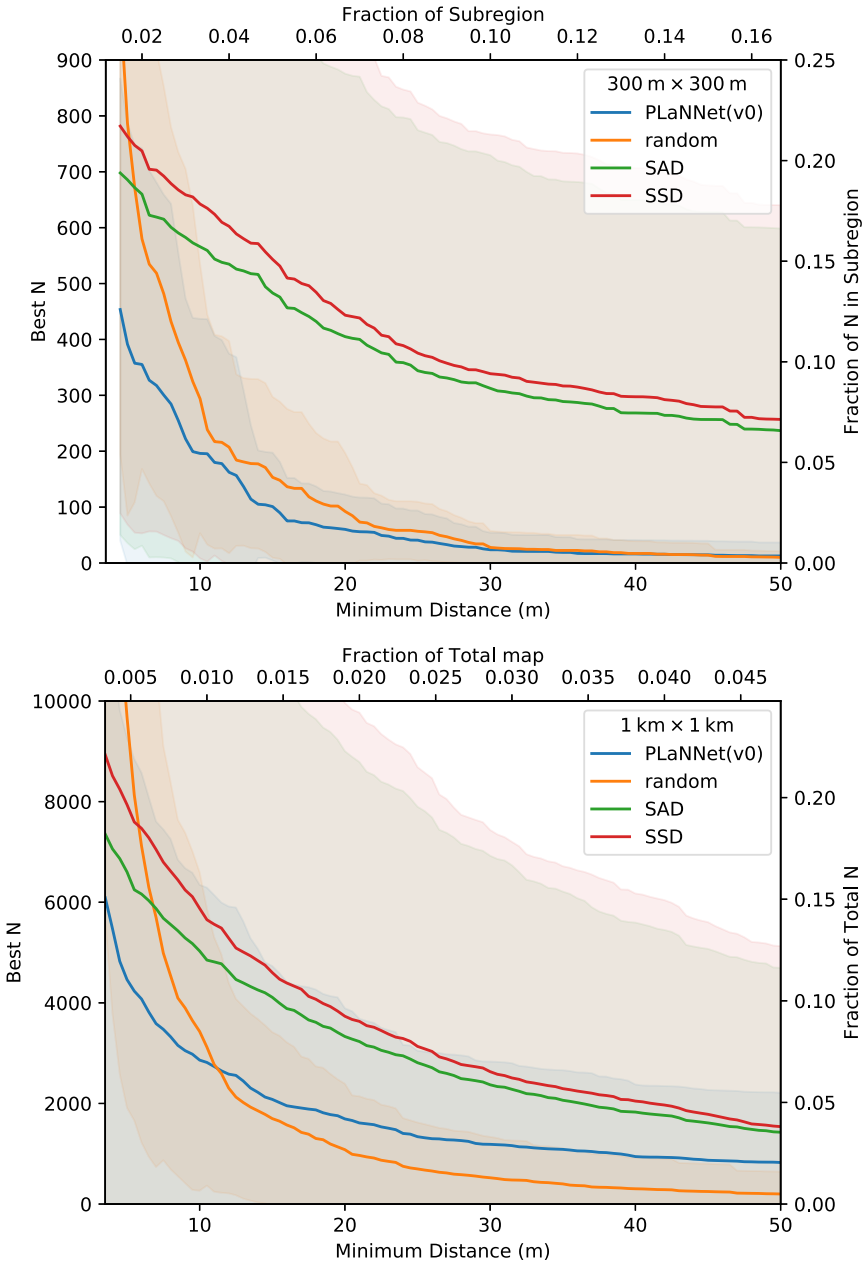

Figure 4.10: Localisation benchmarks comparing PLaNNet against random sampling, SAD, and SSD. Given a minimum distance from the ground truth location, the rank of the N^b "best" choice as determined by the various methods is found. Averages for 50 locations within the $300\ \mathrm{m} \times 300\ \mathrm{m}$ sub-region (upper panel) and 300 locations within the full $1.05\ \mathrm{k}m \times 1.05\ \mathrm{k}m$ testing region (lower panel) are plotted and their respective standard deviations are shaded. The distance and top N are also shown as fractions of the region length and total included N, respectively.

Table 4.2: Breakdown of the Datasets

Dataset Type	Number of Locations	Fraction of Total	Description
Training	5.62×10^5	0.93	Used to fit the model
Validation	0.30×10^5	0.05	Used during training to tune model hyperparameters
Testing	0.14×10^5	0.02	Used at the end to check the network with unseen data samples

size of 12.5 m. For each position, the reprojected image is displayed in addition to its ground truth location within the testing sub-region alongside the top N=5-10 matches as determined by the neural network.

Our method is further compared against traditional benchmarks in Figure 4.10. The localisation performance from PLaNNet, random sampling, the sum of absolute differences (SAD), and the sum of squared distances (SSD) are shown. For each method, the query reprojection image is compared against each sliding window satellite candidate image and assigned a similarity score. Here, window step size of 5 m is used. Over a collection of different random locations, the N^{th} best choice which satisfies a certain minimum distance from the ground truth location is found. Thus, for a given minimum distance, the lowest value of Best N is desired. PLaNNet achieves the best localisation performance overall in both the 300 m $\times$ 300 m sub-region (3600 candidate points) and 1.05 km $\times$ 1.05 km testing region (40,401 candidate points). In both cases, the neural network requires, on average, only 5% of the available candidate regions to localise within 10 m and 10% to localise within 5 m. SAD

and SSD perform approximately a factor of 2 worse. Random sampling achieves the worst performance for localising within 5 m.

The current system is able to reduce the search area by 90-95%, providing valuable input for any human-in-the-loop localisation. By severely reducing the search space in an automated fashion with high confidence and with calculation times of order seconds, the workload and time required by teams to localize successfully may be sharply reduced.

Furthermore, this method could be used as a starting point upon which to develop additional systems to refine localisation, as vast reductions of search space would allow for greater computational resources to be dedicated to such algorithms. The network itself is also relatively small and inference can be run in a matter of seconds (on a current laptop). As such, it could potentially be run onboard a spacecraft, given that we can load the satellite images and the Neural Network model on-board. The performance will depend on the size of the region to search, as well as the selected image resolution. Alternatively, reprojecting the surface images onboard is also feasible, resulting in only needing to downlink a single 224×224 pixel images, which is the more realistic scenario, especially for the Moon, where the communication delay is only at a couple of seconds, and such a small file can easily be downlinked.

4.3.2.3 Discussion

The presented approach is a step forward in expediting localisation by reducing the search space and time and can serve as a guideline for future approaches to improve absolute localisation on planetary surfaces. Extensions to this short-study proof-of-concept include: a greater parameter study within the synthetic environment (e.g., multiple sun angles and illumination conditions), utilising 3D reprojections to overcome flat plane restrictions with

additional sensor data, training the response of the network on the sliding window outputs end-to-end, training the network from scratch instead of using pre-trained weights. On the simulator side, the quality could be improved by switching from a fully synthetic environment to a terrain based true lunar or martian DEM's to increase the realism of the virtual environment. The lack of rover tracks could also be addressed here, since rover tracks are features which are not present in the orbital imagery but would be present on real surface perspective imagery. To overcome the sim2real gap, experiments using real data, from lunar or martian analogues on earth (e.g. Devon Island) would also be helpful to prove the feasibility of these method on a further step. For the training and testings parts, we currently use the same virtual environment, even though we select different regions for the training, validation and testing datasets. Future work would need to test how well training in a virtual environment or lunar analogue can help to run the inference part on the lunar or martian surface. This is essential because we currently do not have enough data to train the network with real lunar or martian data. We expect this part to be feasible, because our network is primarily learning the distortion between the reprojected images and the orbital images. The actual feature extraction part is based on an existing pre-trained neural network (ResNet 50), which has been successfully used in a variety of different applications. [92, 99, 115, 60].

While our approach is a step forward to accelerate the absolute localisation process, it still requires human supervision. The primary reason for this, is that we can not guarantee that the first choice of our neural network is the actual location. Instead, the network excels at finding similarly looking locations, regardless of how close they are from the actual target. The likely reason for this is that our approach is only feeding the network information from a 50×50 metre section around the rover. It excludes any information on the horizon, which

could be used to differentiate between locations with similar surface features. In order to overcome this issue, we therefore propose a new method in the next section, which compares the complete surface perspective imagery against available satellite imagery.

4.4 Surface Perspective Imagery Matching With Reprojected Orbital Imagery

As described in the previous subsection (4.3.2.3), the initial method we presented for absolute localisation, while an improvement over the state of the art, has some shortcomings. In this section, we try to address these limitations with a modified approach which can also include horizon features in the matching process. In theory, these horizon features should help to differentiate two locations with similar surface textures. With this modification, we aim to improve the localisation accuracy and speed, as well as to increase the possible size of the surface area to be analysed. In more detail, we propose the following approach (figure 4.11).

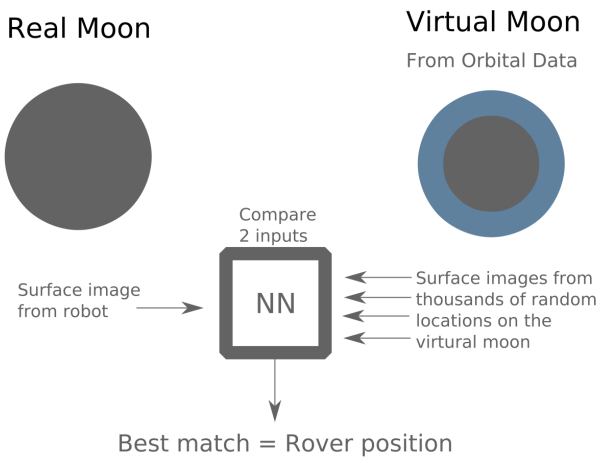

Figure 4.11: The proposed method compares a surface image with thousands of surface images from a virtual environment based on real data.

First, we build a virtual 3D environment of the expected landing site with satellite imagery and digital elevation models (DEM) as this data is already available for most regions of the

Moon * and Mars †. Once we have images from the surface of the real Moon, we randomly spawn our robot in different locations of our virtual environment and we take surface images in each one of these locations. These images are then compared with the real surface images by a neural network, which has been trained to return a likelihood of a match. After iterating randomly over the region of interest in our virtual environment, we select the location with the highest likelyhood of a match. As we know the location of the matching image from our virtual environment, we now also know the location of the actual position on the Moon.

As in our previous works, we demonstrate our method in simulation, where we have full control over all parameters. Once we can prove our hypotheses in a virtual environment, we can then expect to move onto experiments using real data. While we specifically mentioned the Moon, this work could also be applied to any planetary surface with a GPS denied use case, including Earth.

This work has been completed in collaboration with *Ben Wu*, with Dr Wu focusing on the Neural Network architecture, and myself focusing on the data generation part. Any other components were done collaboratively.

4.4.1 Methodology

4.4.1.1 Dataset Generation

For this approach, we need to produce two separate environments, and *real-virtual-moon* with the highest possible surface details, as well as a *reprojected-virtual-moon*, which has been built only from orbital imagery from the first environment.

*https://trek.nasa.gov/moon/
†https://trek.nasa.gov/mars/

For the *real-virtual-moon*, we make use of the same virtual environment inside Unreal Engine* as presented in section 4.3.1.1. With its detailed procedural textures and random rock distributions, the environment provides a high level of detail down centimetre level. The terrain uses a height map at a one metre per pixel resolution. This resolution limitation is, however, not noticeable in most locations as the terrain is relatively flat, and because the surface shaders provide the illusion of terrain detail with surface textures and bump mapping (top half of figure 4.12). The only difference to the data generation part was the manner in which the data was collected. As opposed to the previous method of using Fraps† to capture the image data, we used our now openly available data generation tool ‡. The difference in the capturing method is that we now use Unreal Engine's internal screenshot capturing method rather than relying on an external tool. This simplified the generation process as it is easier to match the terrain coordinates to each image. Yet it also takes significantly longer to run. As the original dataset was no longer available due to it's size, we had to generate it from scratch. Although the random seed and therefore the random locations are different, the environment is identical in any other way. For the *reprojected-virtual-moon*, we make use of data that could be available from orbital data alone. In our case, this is a combination of a Digital Elevation Map and satellite imagery. As the Unreal height-map is already limited in resolution with 1 metre/ pixel, this is used as is. This type of DEM resolution is a reasonable assumption, as 1 metre/ pixel DEMs have already been generated for the Moon using existing satellite imagery and a technique called photoclinometry [100]. For the surface textures, we have generated our own orbital images in Unreal Engine by moving a perspective camera

*https://www.unrealengine.com/
†http://www.fraps.com/
‡www.moonbench.space

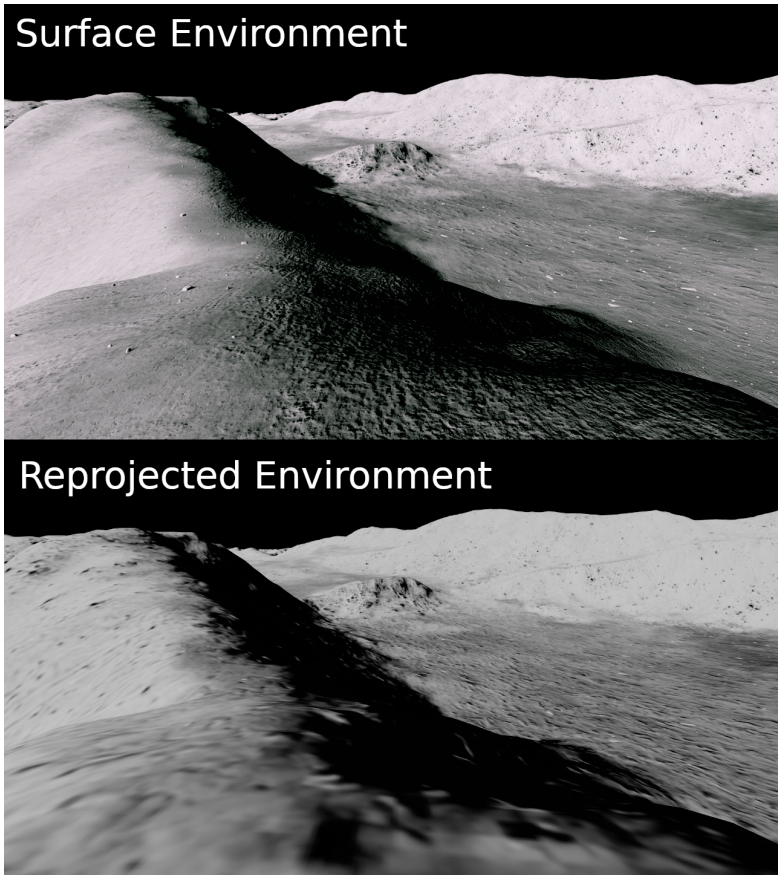

Figure 4.12: Both 3D environments used for the localisation process, top is the high resolution simulation environment, bottom is the reprojection environment based on lower resolution satellite imagery.

at a high altitude (50km) over the complete map. The resulting data was then stitched into one complete image of the surface. This was done similarly as before, but this time, the complete 64 square kilometres of the terrain needed to be mapped, while we only had mapped the regions of interest previously (training, validation and testing areas). Since we are dealing with a completely virtual environment, we could also make use of orthographic view cameras for this satellite imagery capture. However, due to technical limitations of Unreal Engines 4, these do not render the scene in the same way as perspective cameras do, resulting in illumination and shading differences. As a result, we have to use perspective images while accepting small amounts of distortion, which are limited instead through a narrow field of view at a large distance, as well as very small capture area of only 50 by 50 metres. Through the described process, we rendered the complete map (8×8km) of our Unreal Engine environment from above in sections at a resolution of 5cm per pixel resulting in a total image size of $163,840 \times 163,840$ pixels. We picked this resolution because this is currently the best possible satellite imagery available on the Moon, and it can be achieved during the landing phase ([122]). Once we have successfully demonstrated our approach for this resolution, we intend to downgrade the imagery resolution to 0.25 and 0.5 metres per pixel, which match the resolution currently publicly available for Mars [137] and the Moon [164]. As opposed to the previous method, experimenting with different satellite image resolutions does require re-rendering the surface perspective in our reprojected environment.

As render engine for our reprojected environment we made use of Blender* and Cycles†. Blender was chosen due to its python interface, which allowed us to quickly build a pipeline

*https://www.blender.org/

†https://docs.blender.org/manual/en/latest/render/cycles/introduction.html

around our use case. Eevee * was also initially considered as render engine, because path tracing is not necessary for our pre-shaded environment, speeding up render times considerably. However Eevee does currently not support headless rendering on machines without a display. Therefore, Cycles was used as it can run on a server farm, which helps to speed up the data generation on a different scale. While we could also have used Unreal Engine for this step, we believe it is important to have a clearly separated render engine for the second environment, as there would also be a clear difference between images from a rover and the reprojected surface in a scenario using a real robot. In order to load the captured orbital imagery into Blender, it had to be subdivided due to the large amount of data to be handled.

Inside Unreal Engine, the subdivision of the surface area was not necessary for two reasons. Firstly, unlike Blender, Unreal supports large geometry objects based on a texture of 8129 x 8129 pixels (where the distance between two pixels equals 1 metre in our case). In order to display such a large object, Unreal automatically subdivides the object into *quads*, some of which are shown at a lower resolution, depending on their proximity to the camera. The second issue is that the original environment used procedural surface textures. In order to produce an exact pixelated version of the same texture, they had to be saved as an image. In our case, as the terrain is very large at a 5cm resolution, this resulted in a terrain texture of $163,840 \times 163,840$ pixels, which is too large for a single texture image.

Therefore, both the DEM and the surface textures were subdivided into smaller tiles of 256*x*256 m, and a high and low resolution version was made for each tile. The DEM tiles are at a resolution of 257×257px and 33×33px while the texture tiles are of a resolution of 5120×5120px and 640×640px. The DEMs were subsequently converted into a 3D

*https://docs.blender.org/manual/en/latest/render/eevee/introduction.html

mesh with Open3D. With the help of custom Python scripts, the resulting geometry and textures were loaded into Blender as can be seen in figure 4.13. As textures already have the shading backed in, no additional shading was applied. For the rendering process, the areas

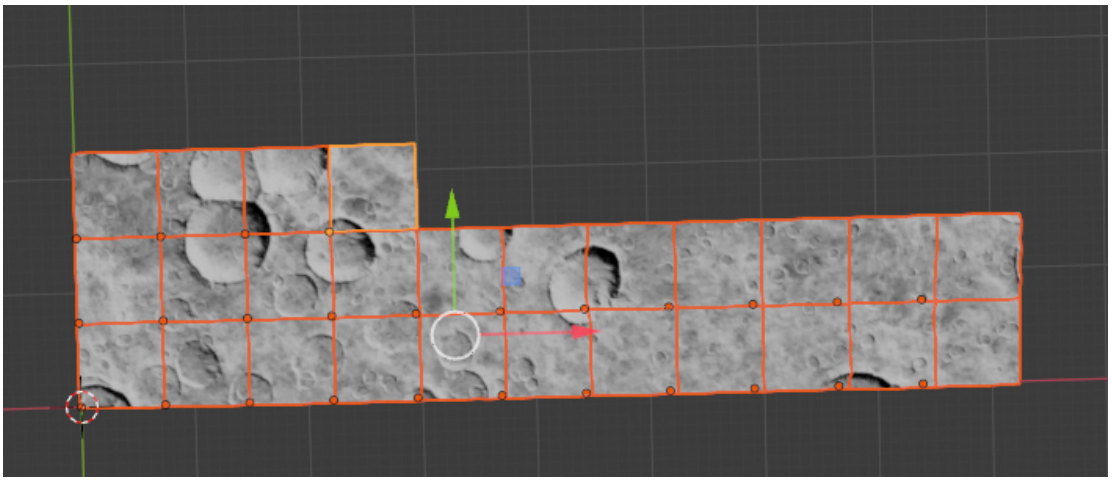

Figure 4.13: Subdivided terrain tiles loaded into Blender.

to be rendered were limited to a 1×1km locations. The main location, as well as the bordering tiles were loaded at high resolution, while the remaining tiles were replaced with low resolution versions. This was done as Blender was not able to load the complete environment at a high resolution (A $1 \times 1km$ surface area at a 1 metre per pixel resolution results in $(999 \times 999) \times 2 = 1996002$ polygons). In the top half of figure 4.14 we can see the difference in surface geometry between high and low resolution tiles. The bottom half shows the two different texture resolutions.

As visible in figure 4.12 the primary difference between the images is the lack of surface texture solution in the reprojected environment, as well as the rocks that are now baked into the surface texture. This is exactly what one would expect for a reprojected environment from a real satellite image. We took great care to match the positions of objects and cameras

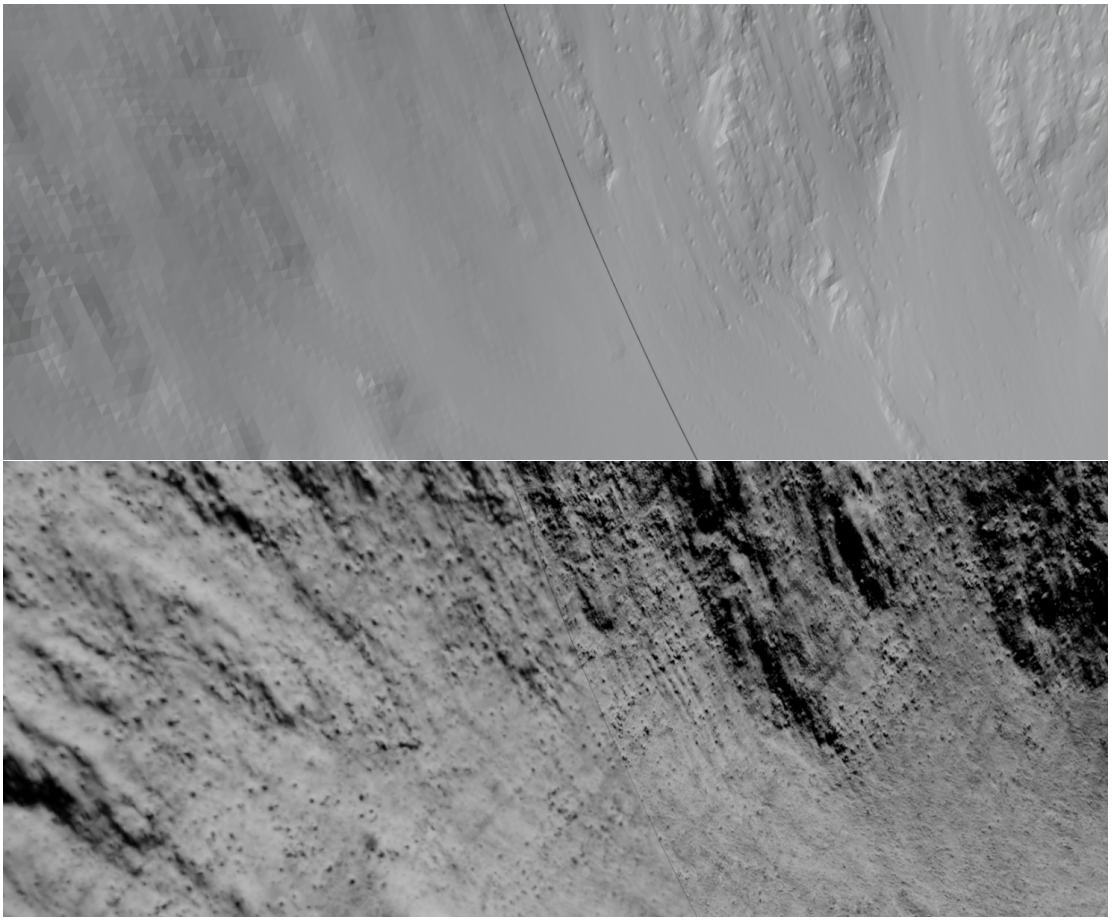

Figure 4.14: Adjacent low (left) and high (right) resolution tiles. The top image chose the shaded geometry while the bottom image shows the final textured tiles.

in Blender exactly, as can be seen on the horizon where the surface geometry and the textures are an exact match. For the rendering process we initially considered rendering new Blender locations around every single location from the original Unreal Engine dataset (as shown in figure 4.15). As we already produce a dense coverage distribution inside Unreal Engine (210.000 locations), a similar distribution can also be generated from the origin locations. To simplify the setup, we therefore rendered the exact same locations inside Blender, which we are then able to pair selectively with the Unreal locations.

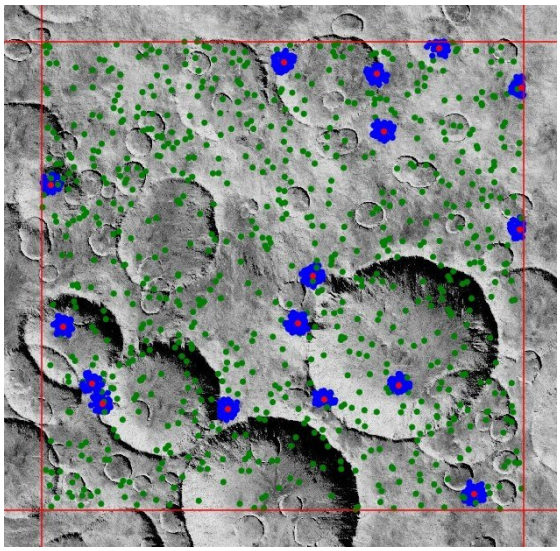

Figure 4.15: Original distribution considerations for training a pair of surface and reprojection images. Red are the surface images, green are random reprojection locations, and blue are reprojection locations based on a Gaussian distribution offset.

4.4.1.2 Generated Datasets

In order to train our network, we need to divide our terrain into different locations for training, validation and testing. All of these have no overlap and are sufficiently far enough away from the edge of the map. We also ensured that all three environments experience different

types of terrain, such as flat areas and crater areas. For this test, the selected locations were identical to the ones presented in the previous approach in section 4.3.1.1 and figure 4.4. For this test, we have generated 210.000 locations in our surface environment, and another 210.000 locations in our reprojected environment. The surface locations were randomly distributed and for the reprojected locations we initially considered picking locations that are close to existing Unreal Engine locations for training (figure 4.15). However due to the limitations in rendering time, this would have required extensive additional render times, each time we would test the training process with a different location matching distribution. Instead, we decided to render the same locations in the reprojected environment as in the real surface environment. Due to the high density of locations we have rendered this way, we were still able to test different approaches to match up image pairs for the training process.

4.4.1.3 Neural Network

The network layout proposed for this approach can be seen in figure 4.16 and is similar to the one described in section 4.3.1.3. The network is also a Siamese neural network, where the true surface images and the re-projected surface images are loaded into two respective branches of the network. The primary difference between the two approaches is that the images are fed into the network are at their full HD resolution. The four images (front, right, back, left) are simply stacked, without the need for any coplex preprocessing or stitching. As before, we make use of a re-trained 50-layer ResNet v2[77, 78] feature extractor*. The pre-trained weights of the feature extractor come from training on the ILSVRC-2012-CLS image classification dataset, and were not modified during the training of our network. Because the

*https://alpha.tfhub.dev/google/imagenet/resnet_v2_50/feature_vector/1

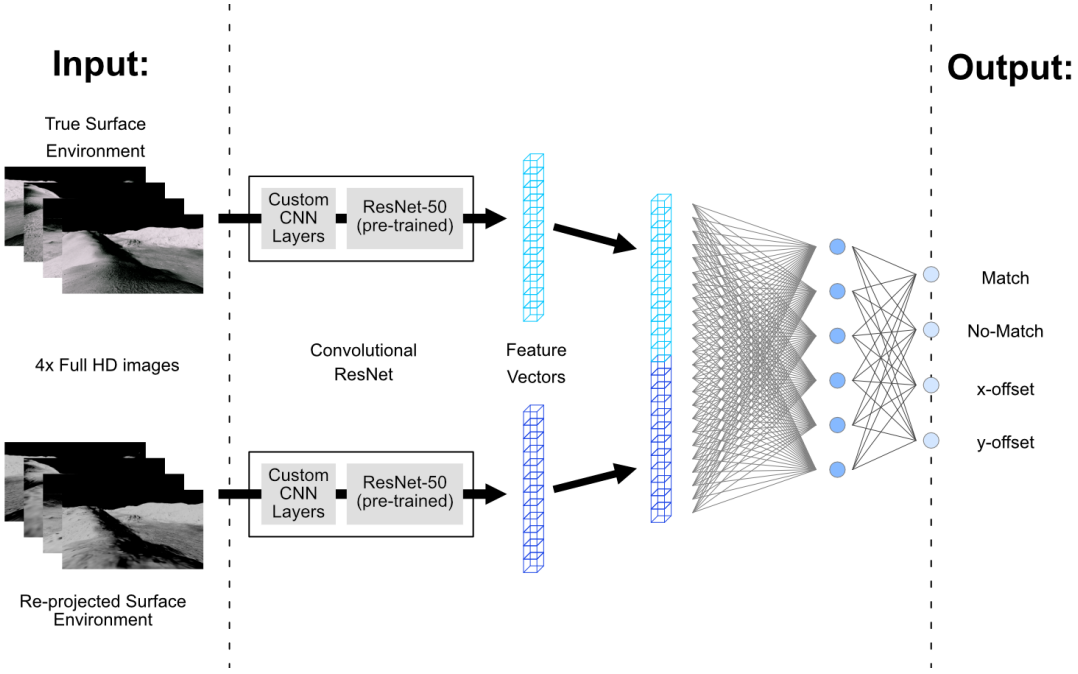

Figure 4.16: Proposed network layout

ResNet-50 network has an input of 224×224, we need to add our own custom CNN layers to connect to original input image vectors to the input of the pretrained Resent setup. Additionally, the input images also need to be converted from greyscale to RGB. The resulting feature vector of each branch is of length 2048, and the two vectors are concatenated into a single 4096-element one-dimensional vector. The other difference in our network is the output, as we not only seek to learn a match/no-match probability, but we also hope to learn the horizontal (x, y) offset. This change should help guide our solution to the correct location if the network is confident that it is close to the actual target location

4.4.1.4 Localisation

For the final localisation step we use our trained model to compare a real surface image with a set of rendered images from our projected surface environment. With rendering times in mind, a dataset of uniformly distributed locations should be rendered beforehand in our reprojected surface environment in order to localise as fast as possible. Given two sets of images, our model then returns the likelihood of a match between the two locations. The simplest approach is to sample our complete dataset of pre-rendered location. At the end, we then pick the location with the highest likelihood and add the estimated x and y offset values to its position. We currently estimate that our inference of our model would take at the order of one second, which can quickly add up, depending on the surface area and the dense density of the surface coverage is. As our method can easily be run in parallel on multiple machines, executing this in less than an hour is, however, not an unreasonable assumption for a method that we run back on earth.

In order to speed up detection, a more optimised version can also be considered. In this case, we will initially pick locations at random until we hit a location with a high matching value. We then continue to pick locations within close proximity, in order to validate the original match. Additionally, since our network also provides an estimate of the horizontal offset (x-offset, y-offset), we can also pick the next location taking into account the current offset direction. This process can be repeated until we maximise the likelihood of a match based on all the images in the proximity. While this approach considerably reduces the number of images that need to be processed, it also can also fall into a local minimum.

To limit the risk of this happening, we propose a hybrid approach, where the complete map is sampled with a uniform distribution of density d. Of these locations, then only the

top n number of locations are considered for a further proximity search.

Furthermore, we can also try to improve our estimation by stepping outside of our pre-rendered dataset. Once we have our highest likelihood match, we can use the x and y offset values to render new locations from our reprojected surface environment to further minimise the localisation error. Additional locations can also be rendered to validate the results. To note, this method does require more processing time, as our render times were at the order of several seconds for a single image. Nonetheless, this is not an unreasonable approach. Further optimisation on rendering side could also reduce rendering times. The quality of the presented methods still need to be evaluated experimentally.

4.4.2 Results and Discussion

4.4.2.1 Results

In this work, we have presented a detailed new approach for absolute localisation which should be an improvement over the method presented in 4.3. Unfortunately, the work on the machine learning model has not been fully completed at the time of writing. Therefore, we currently do not have any results on the feasibility of the proposed network architecture and the accuracy of the subsequent inference and localisation process. The data generation component has been completed and a detailed description of the generated data can be found in tables 4.1 and 4.3.

4.4.2.2 Discussion

While the work shown in this section is still in progress, we have described our proposed approach in detail. The dataset generation has been completed and accounts for a signifi-

Table 4.3: Summary of the LUNA2 Localisation Dataset

Item	Physical Scale	Resolution	Quantity
Training Region	2.05 k$m\times$ 2.05 km	41000$\times$ 41000 px	1
Validation Region	1.05 k$m\times$ 1.05 km	21000$\times$ 21000 px	1
Testing Region	1.05 k$m\times$ 1.05 km	21000$\times$ 21000 px	1
Surface Images	90×50.6	1920×1080 px	840.000
Reprojected Surface Images	90×50.6	1920×1080 px	840.000

cant part of this project, thus enabling the machine learning work. Direct image comparison through Neural Networks has been demonstrated before and should be easier to perform in this case due to the lack of distortions. This and the addition of horizon features makes us confident that this method will perform at least as well as the previous approach. While we are less confident of the x and y offset metrics, they are not required for this approach to function. That being said, they would provide be a helpful addition over the state of the art. Aside from the continuation of the current work, we also propose to refine the satellite imagery with corrected orthomaps. While real satellite imagery is also not truly orthographic, the pushbroom cameras and the post processing applied in this case enable reasonably accurate orthographic views which our current perspective views are not comparable to.

In any case this, the limitations of this work are very similar to the approach described in section 4.3, and could be improved through a better simulation environment, a wider scope of test parameters, as well as field tests with real data.

Table 4.4: Breakdown of the Datasets

Dataset Type	Number of Locations	Fraction of Total	Description
Training	200.000	0.95	Used to fit the model
Validation	5.000	0.024	Used during training to tune model hyperparameters
Testing	5.000	0.024	Used at the end to check the network with unseen data samples

4.5 Summary

In section 4.3, we presented a novel proof-of-concept approach to planetary rover localisation: surface perspective-to-satellite perspective image matching using machine learning, designed to improve localisation efficiency and accuracy compared to current methods. The outcomes of this project were: generation of a localisation benchmark dataset ($\sim$10 TB data) and generation tools - available to the community for testing and training their localisation methods; a demonstrated proof-of-concept for automated absolute localisation using the ResNet convolutional neural network; and a method to assist human-in-the-loop localisation. The resolution used within the generated lunar landscape was comparable to that of the highest resolution martian images, meaning these methods could be directly applied to martian environments. Based on these results, we then proposed an improved approach in section 4.4 where we presented surface to surface matching between real and reprojected environments, using machine learning. The outcomes of this project were: generation of a

localisation benchmark dataset (∼3.0 TB data) consisting of two renders in two virtual environments; a proposed method for automated absolute matching and offset detection using the ResNet convolutional neural network; and a proposed method to perform automated localisation in GNSS denied locations. The satellite image resolution used was equivalent to resolution gathered during a recent lunar landing, meaning these methods could be directly applied to lunar environments. As future work for 4.3, we propose to improve imagery with true 3D surface reprojections using stereo camera imagery. For section 4.4 we are currently finalising the training of the models in order to gain some comparable results. As additional work, we propose to replace the orbital imagery used in this work with truly orthographic imagery with the help of a different simulation environment. Lastly, both proposed methods require some validation using real imagery in order to overcome the sim2real gap, before they are applied to a real mission. This should be done in a lunar or martian analogue environment, in order to gain a large number of different sample locations with accurate ground truth (GNSS). This would also address some of the issues of the limited parameter space (fixed lighting conditions, no rover tracks), which were deliberately chosen in order to reduce the complexity of this proof of concept work.

5

Conclusion and Future Work

In this chapter we provide an overview of the results achieved throughout the different chapters of this thesis before providing a prospective outline on where the field appears to be heading. Here, we will discuss which technologies will most likely advance the field of pose estimation on other planetary bodies in the near future.

5.1 Conclusion

In this thesis, we have addressed the localisation issues which arise from having very short (less than 14 earth days) lunar surface missions. This is especially interesting for commercial New Space operators considering this type of mission to avoid a costly night time survival. To support this goal, we have worked on three separate angles. The combination of the methods presented are a guideline on how to best build reliable localisation systems which can be used to increase the science return of lunar surface robots and any other planetary surface missions.

A. Testing and Validation

Objective: *Configure an array of testing environments to validate lunar localisation systems on Earth.*

In chapter 2, we have looked at how localisation methods can be validated for the Moon. As with any new approach, the potential users of such methods require some degree of confidence that they will be effective. This is especially difficult for space applications, due to the inability to test our methods in the target environment. To achieve this, we have considered different approaches, resulting in two new indoor testing facilities, the evaluation of four different simulation environments (three of which were also configured by the author), as well as three field tests in a quarry with different robots. Because none of these testing environments can individually reproduce the conditions on the lunar surface, a combination of multiple approaches will be necessary, at least until regular testing on the lunar surface becomes a financially viable option.

B. Relative localisation

Objective: *Evaluate different localisation systems in order to recommend a relative localisation system for the lunar surface covering both hardware and software.*

In chapter 3, we have considered the measures needed to best speed up rover traverses with the help of relative localisation systems. Here, we focused on both the hardware and software sides of localisation systems which can fit on small planetary rovers. For the hardware, we examined different sensors in their usefulness for localisation and their applicability for lunar surface operations. Given that stereo camera systems performed the best in this trade-off, we also covered considerations for configuring stereo camera parameters. In dealing with the software component, we covered the state of the art, and presented an extension to graph-SLAM systems. Through our work, we have shown that it is possible to improve existing SLAM localisation systems in the specific use case of localisation around Lunar landers. Additionally, this method can also be used to enable multi-rover systems, without the need for map exchanges or fiducial markers.

C. Absolute localisation

Objective: *Build an absolute localisation system to speed up rover operations immediately after landing.*

In chapter 4, we have looked at methods to determine a rover's positions in orbital imagery to speed up operations immediately after landing, as well as to support mission concepts reliant on following trajectories pre-planned on orbital imagery. In order to achieve this, we have

generated a large dataset with co-referenced images which is now publicly available*. With the help of this dataset, we have trained a neural network to compare orbital imagery with re-projected surface perspective imagery. The resulting network has the capacity to reduce the search space of possible locations by 90%. From the results and lessons learned, we have also explored a second approach which addresses the short-comings of the initial method. For this work, we have generated a second dataset combining surface perspective image and re-projected surface perspective image. While a resulting network which has learned to compare both of these inputs is still being worked on to validate this approach, we expect an improvement over the method that we initially tested.

5.2 Main contributions

The main contributions of this thesis can be summarised as follows:

1. Inception, development and qualitative assessment of different benchmarking pose estimation techniques for the lunar surface, including simulator environment, indoor lunar yard and lunar analogue field tests.

2. Qualitative assessment of different sensors and algorithms for their localisation suitability on the lunar surface.

3. Development of an extension for a relative localisation system which is automatically corrected through the detection of a lunar lander.

*www.moonbench.space

4. Development and validation a new absolute localisation approach through surface perspective and orbital imagery.

5. Development of a second absolute localisation based on the results of the first approach.

6. Generation of two different datasets to facilitate machine learning-based absolute localisation.

5.3 Future work

In this section we will explore where the work accomplished could be adapted or extended upon in the future.

A. Testing and Validation

The next step to improve the illumination accuracy of simulation environments is to include Hapke shader or a Hapke-approximation Shader. On the field test side, a more extensive field test campaign will be necessary. On one hand, this could be addressed by moving into a basalt quarry instead of a sandstone quarry, where the soil properties are more similar to what we could expect on the Moon. Ideally however, the next field test would be performed in a region with volcanic ash such as the tests performed by DLR on Mt Etna. While a basalt quarry works well, the ground is generally removed in flat layers, where we still see the steep walls of the quarry in the field of view, which is not the case on Mt Etna.

In regards to testing sites, the ideal scenario will be to test rover systems in their operational environment, which in our case, is on the Moon. However, even once this will more easily

accessible, the testing strategies outlined in this thesis will remain relevant due to the cost of testing on the Moon. The most significant changes will occur on the virtual simulation side. It is already clear that upcoming computer game engines are trying to address the scale issue, which happens when faced with managing a large amount of detail near the camera while also needing to provide detail in the distance on the horizon. The recently released examples of the upcoming Unreal 5 engine, display some stunning examples of this in action (shown in figure5.1). Along with the added detail in the environment, more accurate lighting models are slowly being used in the computer games engine. While ray-tracing has been purely used for offline rendering purposes until recently, we are now seeing more games containing some real-time ray-tracing capabilities.

Figure 5.1: Image taken from 'A first look at Unreal Engine 5' [48]

B. Relative localisation

Our novel SLAM approach to simply localise around a lunar lander is promising. While we have successfully shown that it works, it still needs further research, as it is currently only a

proof of concept. Image-based rigid body pose estimation algorithms are advancing quickly, and it clear that better results can be achieved than the Efficient Pose network we have trained. Given the modular nature of our solution, it would be easy to replace the current network with a new network. The second problem which needs to be addressed is the Sim2Real gap. Using a virtual simulator to build a proof of concept is perfectly acceptable, especially due to the full control over all parameters. To properly judge its efficacy in a real scenario, the experiment needs to be performed with real data. This is especially important when it comes to image data. Therefore, as future work, we propose to make use of a mock lander in a lunar analogue environment in order to gain more confidence in our proposed method.

C. Absolute localisation

In this area, we would like to complete the second machine learning approach which was presented in section 4.4. Since the first method yielded promising results which were primarily limited by the lack of available data on the horizon, it seems likely that the improved approach will provide better results. Depending on the outcome of this ongoing experiment, more scenarios can be explored. Since we plan to use this technology on a moving rover, the next logical step is to feed a complete image sequence into our localisation system to further reduce the uncertainty of the position estimate. Alternatively, we could also extend this work by assuming that the starting position is known through other means, and our method could be used to primarily search during a traverse and within the vicinity of the already known position. This could lead to a further speedup in estimation speed as well as a reduction in the computational complexity where the resulting algorithm could potentially run directly onboard a rover.

For absolute localisation, a significant advancement is expected with the availability of bet-

ter satellite imagery. With any image-based localisation, accuracy is directly related to the image resolution available. Additionally, better orbital imagery will allow for more accurate pre-planning of surface missions, which will make absolute localisation more relevant. Eventually, GNSS systems will be available around the Moon, rendering image based methods obsolete. But even at this stage, the methods outline in this work will continue to remain relevant for surface robots targeting other planetary bodies.

5.4 Outlook

In this section, we provide an outlook on the evolution of the topics discussed for the near future. This section is an informed personal opinion, based on past and current research, as well as comparable examples from different other relevant fields.

5.4.1 Autonomy Risk

In order to truly assess how pose estimation will be performed on planetary rovers in the future, we need to contemplate how the rovers themselves and their missions will change, as this will have the biggest impact on the possible design changes to localisation systems. Today, planetary rovers like Curiosity, Perseverance or the Chang'e 4 are designed as a single complex system where failure is to be avoided at any cost. This leads to expensive systems which only large institutions can afford. As a consequence, operators are less willing to accept risks in their concepts of operations, which could lead to the end of the mission. This includes additional autonomy for any robotic systems. We cannot simply anticipate every

possible situation that can occur during surface operations. For some smaller systems, it is possible to test them exhaustively, but this is only feasible when the set of possible situations is limited. For instance, this can be the case when we consider subsystems that works in an enclosed box where every possible state has been pre-planned. When dealing with unknown environments, and especially when we make use of camera sensors, this becomes impossible. A relevant comparison are autonomous cars. Despite thousands of hours of driving have been recorded by different companies, engineers still find potentially dangerous edge cases, as demonstrated by recent incidents*. While these cars can safely navigate many situations, the current public perception is that they are still not reliable enough. As we can not pre-emptively consider all possible scenarios a system will experience, a 100% safety guarantee will never be given. Therefore, the discussion about reliability of autonomous systems must prioritise what degree of risk is acceptable.

For current planetary rovers, this is a difficult question to answer, as it takes millions of dollars to build and launch these systems. When they finally arrive at their destination after a treacherous journey, a single mistake can still render the rover inoperable. As this could lead to the end of an expensive mission, such mistakes are to be avoided at all cost. As a result, most operation is manual and slow due to the communication delays. Such a conservative operational approach is therefore a major obstacle in exploring large areas of the Moon and Mars. As a result, it is unlikely that major autonomous systems will be widely adopted until multi-rover configurations are launched. Once such systems are available, the loss of a single robot no longer leads to the end of the mission and results in redundancy in numbers. At

*https://www.bbc.com/news/technology-54175359

this stage, the risk of autonomous operations will be easier to accept and to facilitate these multi-rover systems, smaller, less capable rovers will eventually be used. While this may have a negative impact on the sensor suite, the benefit is that multi-robot components will also make new localisation systems possible, like for example the multi-robot radio localisation shown in [174].

5.4.2 Sensors

Current space applications are limited by sensors which are suitable for space. LIDAR, the most promising and accurate localisation sensor available for terrestrial applications, has never been used on a planetary rover. Although the timeline is currently undetermined, solid state LIDAR systems could change this, as they promise to provide the benefits of traditional LIDAR point-cloud information without the need for complex mechanical systems. For terrestrial applications, there are presently only a handful of research devices that could be used for localisation. Even if they become widely used, the cost factor for smaller multi-robot systems is not to be underestimated. With cost, we not only consider the direct pricetag, but the combined cost of mass, power and volume, required in comparison to a stereo camera.

A good comparison we can see for this is found in cheap consumer cleaning robots. While some companies offer devices with technically superior LIDAR sensors (planar LIDAR's), most autonomous cleaning robots still rely on cameras for localisation. This is most likely due to the complete trade-off when taking into account, cost reliability and required accuracy.

While camera technology is well understood and regularly used in space, there are still many advancements and discoveries ahead. In fact, event cameras have already shown that

they are highly beneficial for localisation accuracy and efficiency. Furthermore, apart from the image sensor, most components are identical to existing cameras and thus easier to upgrade to. For small rovers, the switch from passive to active sensors is unlikely to occur in the near future as power is a scarce resource. If cost, reliably and power consumption drop to a reasonable level, LIDAR systems are preferable due to their higher accuracy.

When compared to other systems, it appears stereo cameras with an IMU will remain to be the preferred sensor combination. The added cost of an additional camera sensor is not negligible, but the added accuracy is substantial. Given this trade-off, it seems unlikely that stereo cameras will be replaced by monocular camera systems, even if there have been some improvements in depth estimation in monocular camera systems through machine learning. This is, with the exception of systems, where the robot is too small to contain an adequate stereo baseline, such as in extremely small rovers. Another example would be aerial robots where the large distance to the ground requires a much larger stereo baseline.

5.4.3 Software

With considerations to software, the current systems are mostly limited due to the insufficient computation power available on today's radiation hardened processors. Once this limit is lifted, or at least increased, the widely-used SLAM systems that we can observe on terrestrial robots will also be used in space. In state of the art research, we see interesting examples

of semantic SLAM. For this technology, a semantic segmentation network is used to label all pixels in the scene based on the object that they belong to. This is then used to help the SLAM system distinguish different features when performing feature matching. On longer rover traverses, it is not clear how helpful this would be in environments that are mostly composed of the same objects and materials. However, for multi-robot missions, it could be useful for separating man-made objects from the background.

When looking at robotics frameworks, ROS is at the frontier of new robotics developments. It is only a matter of time until it will be used in space. This is especially evident as NASA has already indicated their interest in building a modified version of ROS2 for use in space.

A

NASA SRC2

The following is the content of the technical report that was delivered as part of the NASA Space Robotic Challenge: Phase II (SRC2). This work was a collaborative effort to deliver a fully autonomous multi-robot solution that can search, extract and deliver underground volatiles in a virtual lunar environment. A total of twenty-five researches were involved in the making of this solution: *Sara Jennings, Adam Cobb, Francisco Rodriguez Lera,Frank*

Figure A.1: NASA SRC2 competition logo (source: NASA)

Soboczenski, Chelsea Sidrane, Manuel Castillo-Lopez,Ben Wu, Maciej Żurad, Sara Gregg, Jose Delgado, Paul Wright, Zahi Kakish, Valentin Bickel, Karthik Venkataramani, Swetha Pillai, Sanjeev Narayanaswamy, Krzysztof Żurad, Ignacio López-Francos, Atılım Güneş Baydin, Lukas Meyer, Dietmar Backes, Daniel Medina, Mathieu Labbe, Miguel A. G. Santamarta, Philippe Ludivig

A.1 Introduction

The goal of the Space Robotics Challenge is to develop a robust system that allows for a heterogeneous, multi-robot team to autonomously complete tasks envisioned for a lunar in-situ resource utilization (ISRU) mission. In Phase I, this involved finding and localizing volatiles, coordinating between an excavator and a hauler robot, and detecting a known object. In this first phase, these tasks could be solved independently, greatly simplifying the complexity of

the problem. In Phase II, the above mentioned tasks had to be solved in combination in a single environment, while also dealing with the added constraint of accounting for the state of charge of the rover batteries.

From the perspective of implementing a solution, the challenge represents the intersection of various robotics fields, namely, perception, localization, planning, and control. Our approach to the challenge was to complete a prototype solution in each of these areas using one or more ROS nodes, and then iterate on them until they were sophisticated enough to coordinate with each other in order to complete the intended task.

While we were unable to complete all of the requested tasks, we were able to create a proof of concept solution to each of the challenges. The implementation, performance and learnings from each of these sub-challenges is described in the following sections of this report, followed by a final discussion and description of performance of the complete solution.

A.2 Methods

A.2.1 Perception

The rover is equipped with multiple sensors which are used to complete the different tasks. There is a wide baseline stereo camera, an IMU, a planar lidar and the volatile sensor. We developed several nodes for the perception subsystem to perform a variety of functions including camera noise detection and removal, object detection (e.g. rocks, processing plant, other vehicles), object pose estimation, and sun direction estimation.

A.2.1.1 Image noise

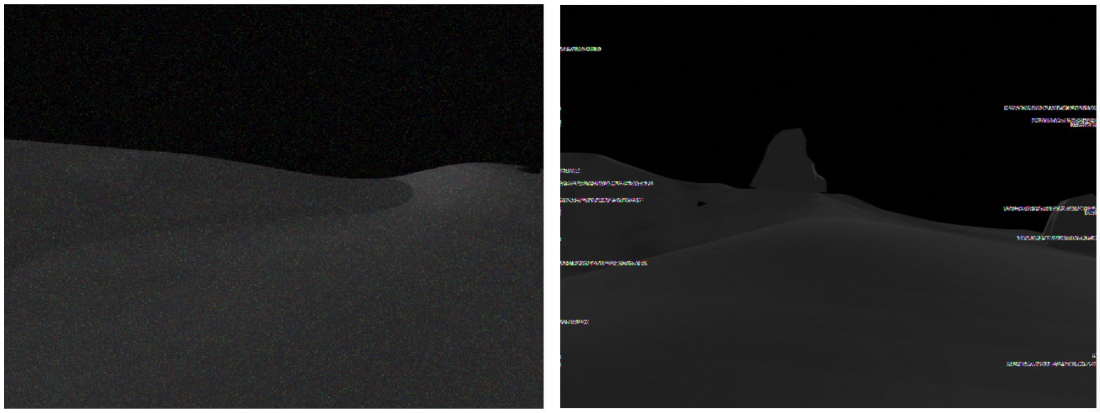

Figure A.2: Left: Image affected by Gaussian noise (Type 1); Right: Image affected by a combination of Gaussian noise and horizontal banding noise (Type 2).

The stereo camera onboard a given rover is affected by two types of noise: gaussian noise and a type of horizontal banding noise (Fig A.2). This noise is problematic because it affects various downstream tasks like SLAM navigation. To mitigate its effects, we developed a noise detector that measures the noise intensity using a signal-to-noise metric. We subsequently choose to apply a noise removal filter if the noise strength exceeds a manually predefined threshold. We performed empirical experiments with various filters to provide noise removal. The filtering performance was measured by evaluating the accuracy of a visual odometry algorithm on the pre-processed images. In those tests, the non-local filtering provided the best balance between filtering performance and processing requirements. In the end however, due to the limited performance for both testing and the final evaluation, we decided against using the filtering approach, and only used the noise sampling approach. In our localization system (more details below) the visual odometry is given the raw data, while the SLAM system is provided the image feed where noisy images are removed. As the SLAM system is operating

at 1hz, this was fine, until the camera frame rate of the simulator was lowered to 5hz. At this stage we had to add an extra constraint, where the SLAM system is fed an image once per second, regardless of the noise level, if all current images are over the noise level.

A.2.1.2 Sun heading

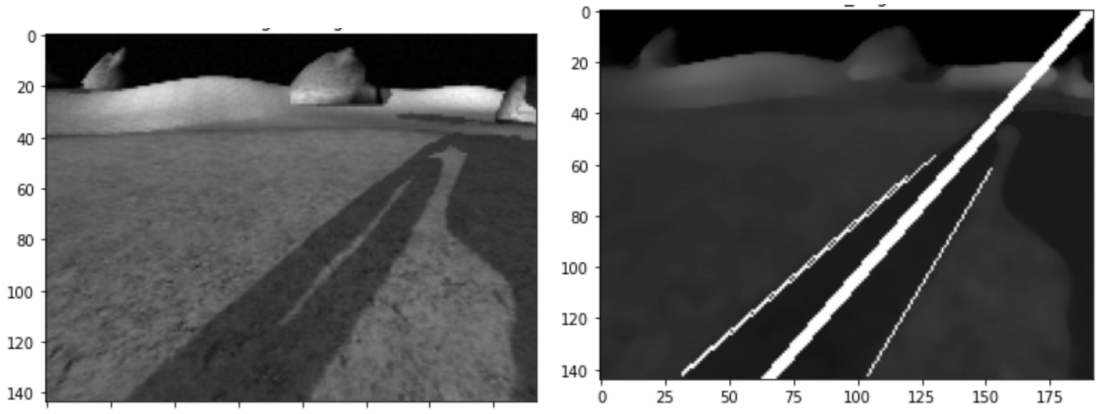

Figure A.3: Left: Image from the camera onboard the rover where the projection of the shade of the rover. Right: prediction of the sun direction as a white line drawn over the rover shadow.

A crucial element in planetary robotic autonomy is absolute localization. In order for the rover to localize itself relative to its internal map, we developed a method to estimate its orientation using the projection of its own shadow with respect to the Sun's position. More commonly this is done using the sun-disk, but this was not available in the given simulator. As the terrain was mostly flat, we used the shadow approach instead. We applied purely computer vision techniques to first determine and outline the shadow of the rover (Fig A.3), and then to determine the direction of the sun and draw a line in the image representing this direction. Due to the added complexity of integrating this approach into our tools, as well as the already high accuracy of our relative localization, we ended up not using this method for the final delivery.

A.2.1.3 EfficientPose

A key component of the challenge is to develop the ability of the hauler robot to return mined resources to the processing plant. In order to effectively do so, the hauler must be capable of identifying the 6D pose of the processing plant. This provides the hauler with information regarding the location of the plant in space (x, y and z coordinates) as well as its orientation about each axis. To accomplish this, we used the official implementation of the EfficientPose algorithm [17] which has demonstrated state-of-the-art performance on the Linemod benchmark dataset. EfficientPose utilizes an EfficientNet [192] backbone, combined with a bidirectional feature pyramid network (BiFPN) and several prediction subnetworks that predict the class, bounding box, spatial location and rotation of the object. The algorithm was pre-trained on the Linemod dataset, and subsequently trained on a dataset generated from the Gazebo simulator that captured the processing plant at various distances and orientations from the perspective of the rover. Additional data augmentation techniques used by Bukschat [17], were not used to generate training data, as experiments involving the addition of such datasets in the training data indicated an undesirable increase in the mean rotation error.

The orientation accuracy of the final trained model did not meet the performance criteria required to carry out the task. While we tried testing the system with position estimates only, this approach only provided limited advantages over loop closure. Finally, given the performance limitations of the grading hardware, we chose not to investigate this approach for improvements.

While a trained EfficientPose model would have significantly simplified tasks requiring collaboration between different robots, we managed to solve the pose estimation problem

using a combination of traditional approaches.

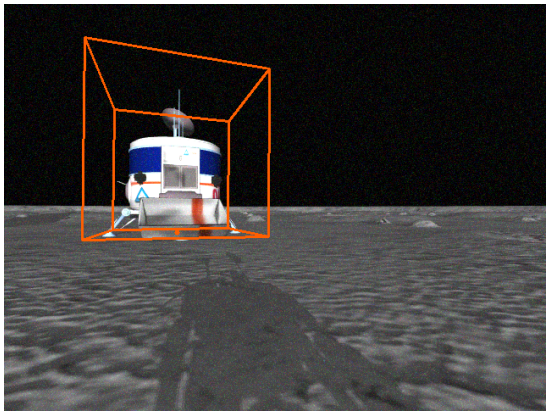

Figure A.4: EfficientPose was used to obtain the pose of an object relative to the rover

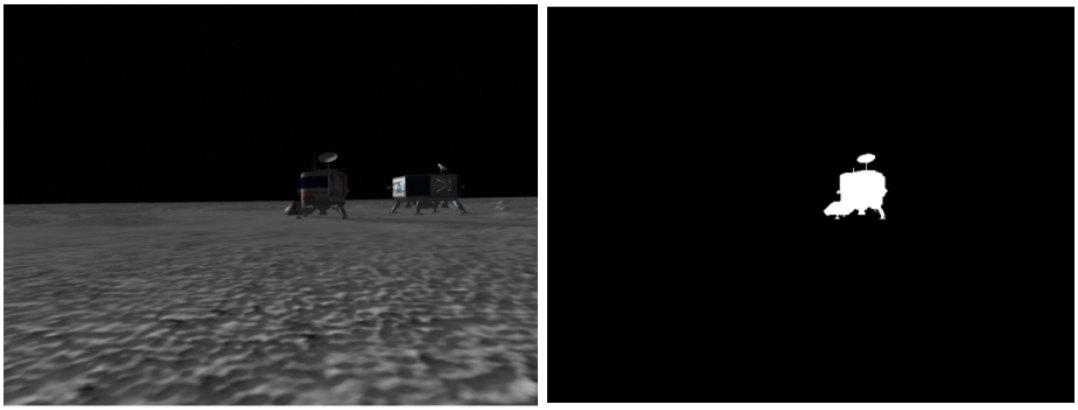

Figure A.5: EfficientPose was trained using RGB images (left) generated from Gazebo, along with relevant masks (right)

A.2.1.4 ML Detectors

We also deployed several vision machine learning models such as a multi-class detection model for rovers, processing plant and base station (Fig 5). Additionally, models for detecting craters

(Fig A.6), the collection bucket for the excavator as well as a rock detection model (Fig A.8). All models were trained using a modified darknet of the You Only Look Once (YOLO) framework [160]. The neural network based models were trained on images obtained from the Gazebo simulator, and were both automatically and manually labeled using LabelImg [200] (Fig A.9).

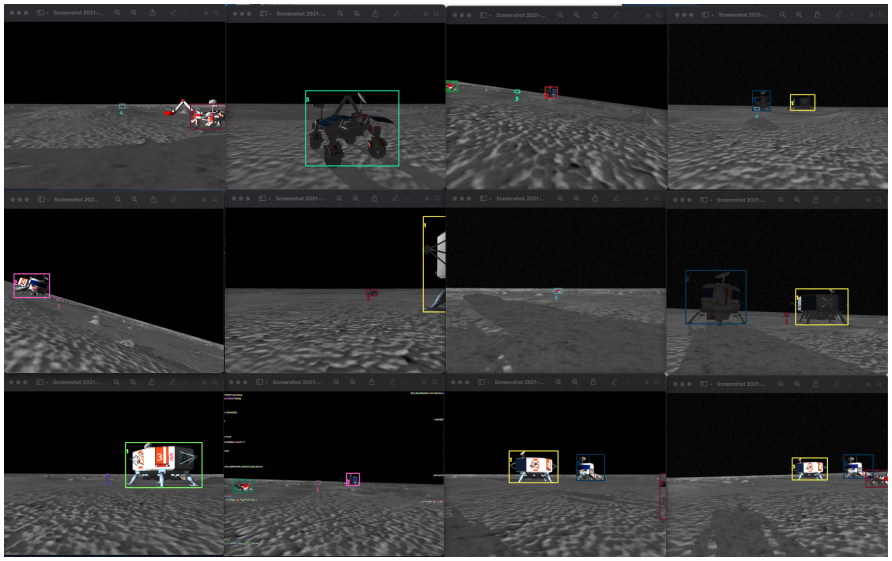

Figure A.6: Summary overview of the multi-class detection models for base-station, processing plant and rovers.

Figure 7: Summary overview of the rock detection model.

A.2.1.5 ML Detectors Implementation

For the YOLO detector implementation in ROS, we combined the detections from the left image with the point-cloud from the stereo images. Based on the bounding box, we cropped the point cloud, and then averaged the remaining points to estimate a relative XYZ offset to the camera. Depending on the object that needed detecting, we used different filtering techniques, to reshape the size of the bounding box, as well as to filter the point clouds. For

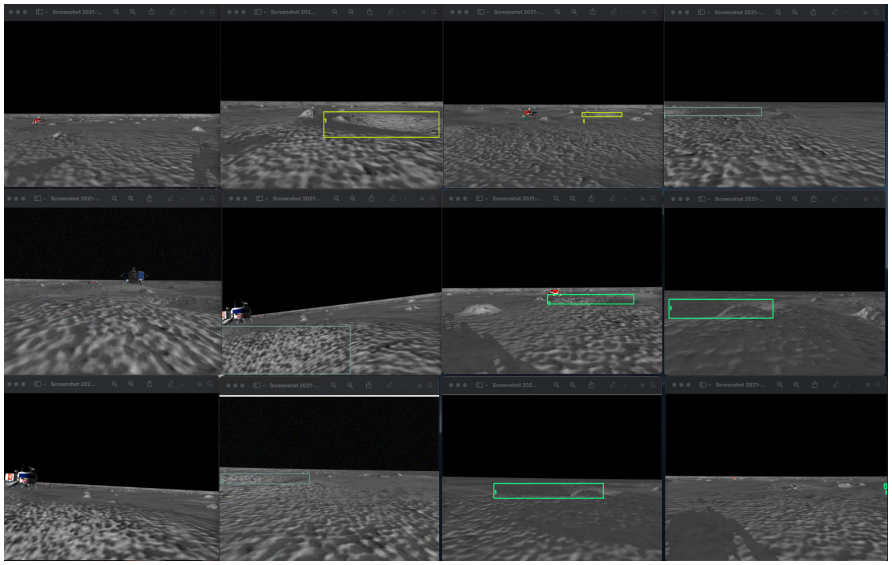

Figure A.7: Summary overview of the crater detection model.

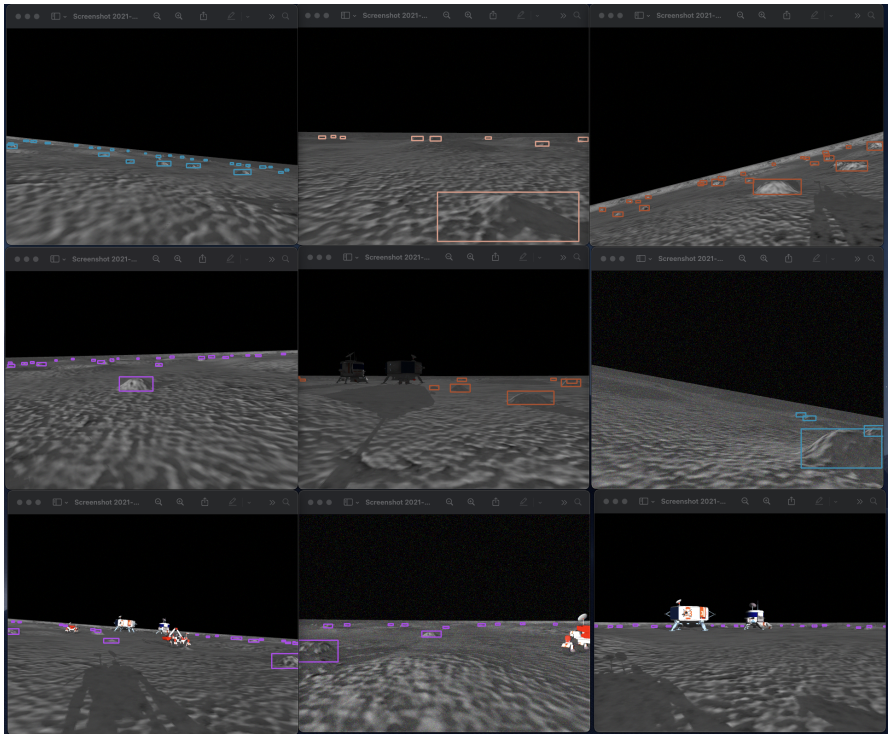

Figure A.8: Summary overview of the rock detection model.

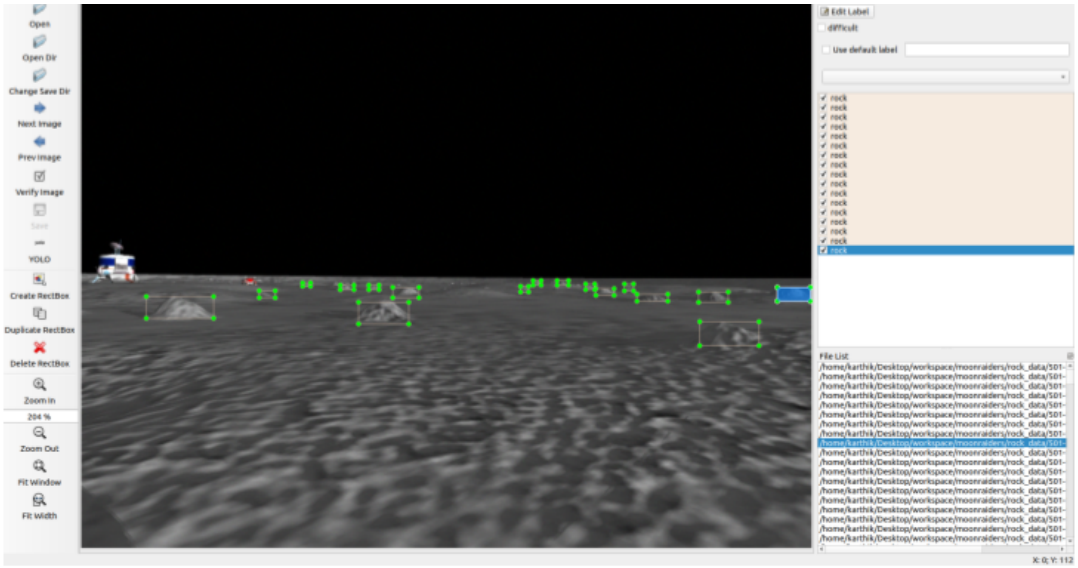

Figure A.9: LabelImg was used to annotate images obtained from Gazebo.

the rock detection for example (Fig. A.10), we only selected points at the bottom of the detection window to avoid the selection of points behind the rock (the blue bounding box is the YOLO detection, while the green bounding box is the reshaped detection). As only rocks in close proximity were relevant in this detection, and due to the decreased estimation accuracy at longer ranges, only rocks at up to 10 meters were considered in this case. For the detection rovers and stations (Fig. A.11), we instead scaled the bounding boxes uniformly, as the center of the bounding boxes generally yielded the best points for a range estimation. In this case we also filtered out darker points from the pointcloud, as the rovers and objects generally provided bright detection surfaces.

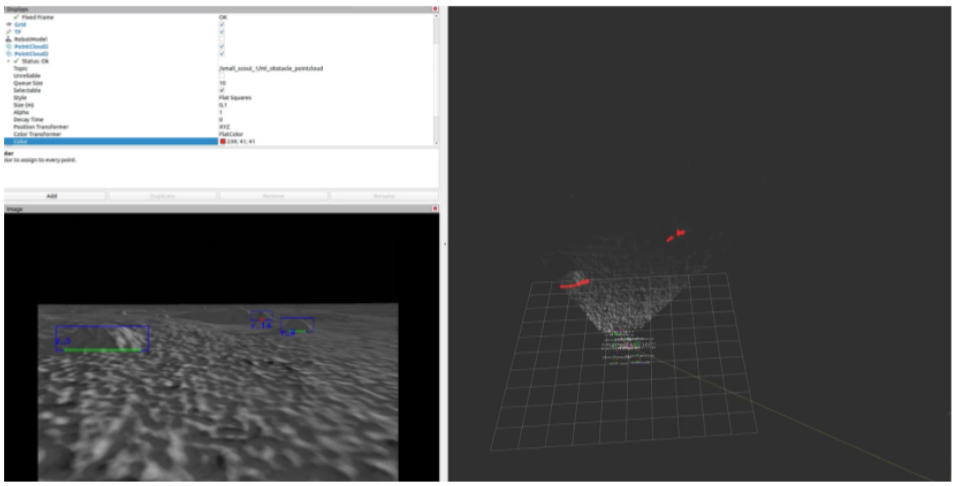

Figure 5. rviz display showing nearby rock detections in the RGB camera (bottom left) and in the point cloud (right)

Figure A.10: rviz display showing nearby rock detections in the RGB camera (bottom left) and in the point cloud (right)

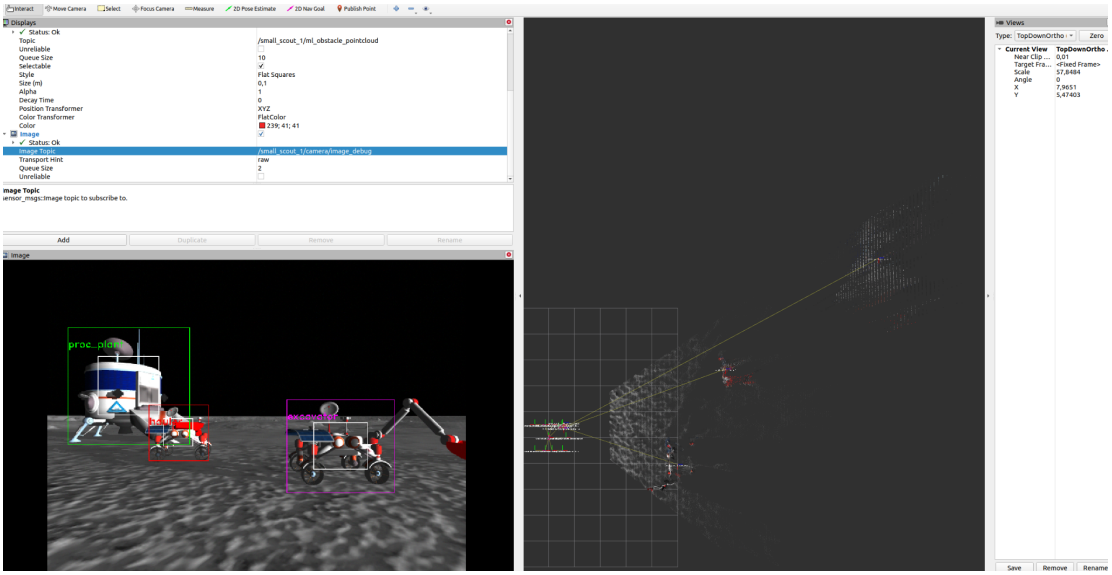

Figure A.11: rviz display showing nearby object detections in the RGB camera (bottom left) and in the point cloud (right)

A.2.1.6 Volatile Detection and Localization

To locate volatiles, we decided to follow a strategy of stopping the scout when the volatile sensor first detected a volatile and performing a maneuver to get multiple readings so that the volatile can be located. The initial strategy we considered was to take advantage of the fact that the scout would encounter any volatile in a semi-circular area enclosed by the volatile sensor detection range and bisected by its velocity vector. The advantages of this method are that only small deviations are required from the planned path in order to locate the volatile. More details are given in Figure A.12. However, in the end, we decided to implement a simpler strategy of travelling along the circumference of the initial volatile detection circle to obtain two more readings. After three readings are obtained, the volatile can be located.

A.2.2 Localization & Navigation

For the localization system, we tested a number of different approaches, but in the end we relied on a purely stereo image based approach. Our system is composed of a stereo visual odometry system which provides continuous odometry updates at a rate of 5 Hz. This odometry is then fed into a SLAM system based on Rtabmap [104], which is operating at a rate of 1 Hz. Rtabmap was selected due to the multitude of features it provided in it's implementation. To increase the accuracy of the SLAM system it is provided with filtered images, where noisy images are discarded. Rtabmap also performs loop closure to correct for drift after longer traverses. This requires some tweaking as the terrain looks very similar, especially due to the repetitive surface texture. Loop closures therefore do not happen very often and require the rover to be in almost the same location to work. Besides this, Rtabmap also pro-

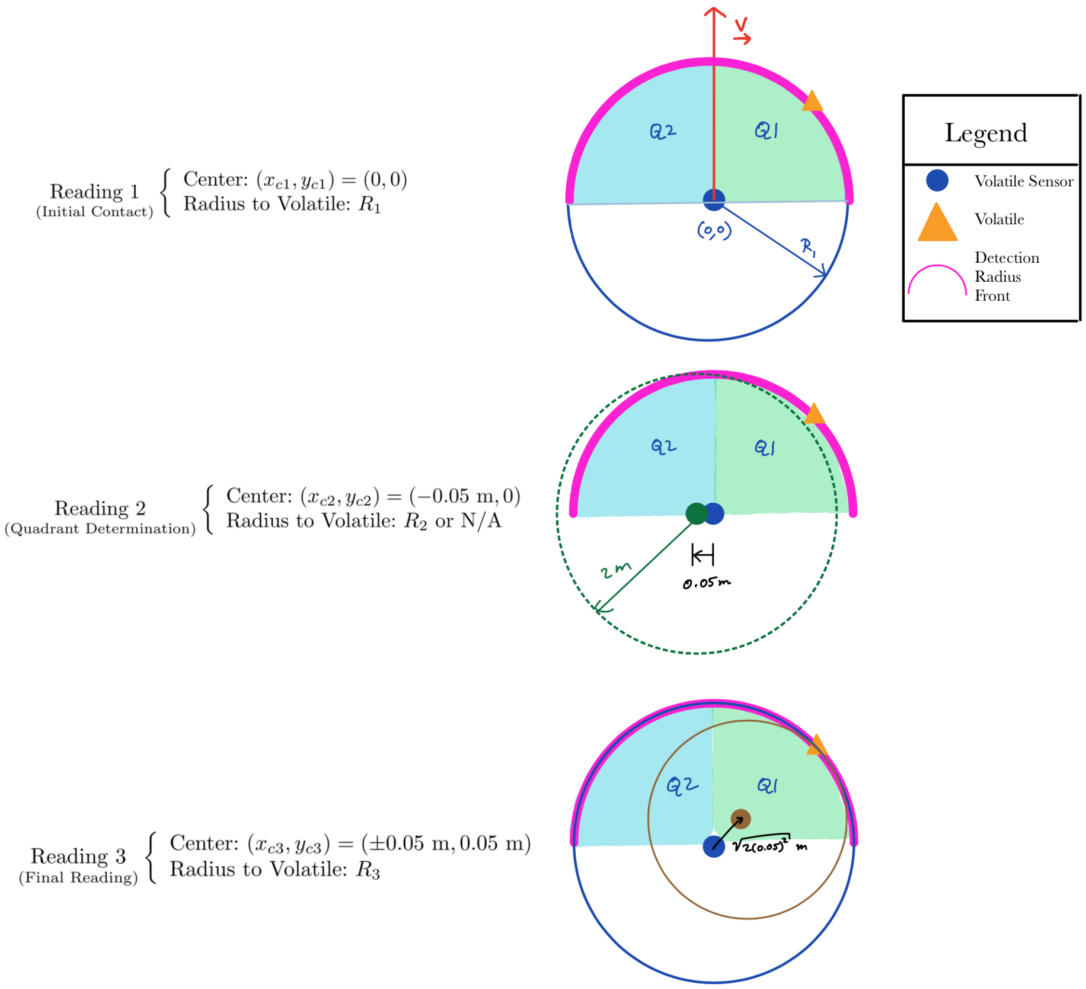

Figure A.12: The semi-circular detection strategy initially considered for volatile detection and location. In the top diagram, the volatile is detected by the volatile sensor. In the second diagram, the scout moves left 5cm. Depending on whether or not a volatile was detected in the second stage, the quadrant containing the volatile can be determined. In the third diagram, the scout moves a certain distance at a 45 degree angle into Q1 or Q2, based on which quadrant was determined to contain the volatile in the second step. If in Q1, the scout moves to (0.05 m, 0.05 m) and if in Q2, it moves to (-0.05 m, 0.05 m). The volatile can be located after this step using the intersection of the first and third volatile sensor readings.

vided the gradient cost map which was used for the navigation part. Fig. A.13 shows the one of the benchmark plots which was used to evaluate the accuracy of the localization system.

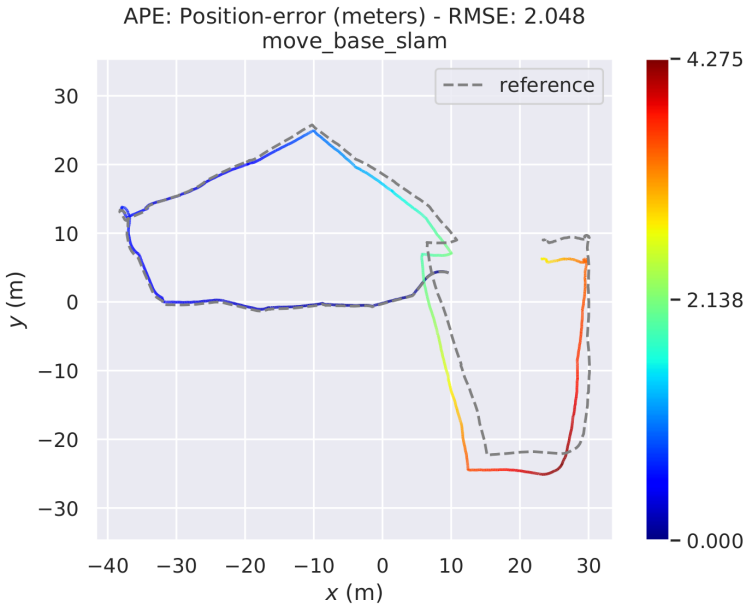

Figure A.13: rviz display showing nearby object detections in the RGB camera (bottom left) and in the point cloud (right)

A.2.2.1 Initial absolute localization

In order to have all the rovers aligned in the same reference frame, we used the initial absolute localization service provided by the organizers. We called this service at the very beginning of the simulation. Subsequently, we published an additional topic for each rover, which converted the rovers relative localization to an absolute pose. This absolute pose was also used for rough rover collaborations. It is important to note here that while each rover was operating in the same reference frame, the drift of the localization was independent for each rover. Therefore, the localization system was only used for rovers to meet near to each other, while the ML detectors were used for operations that required a higher accuracy.

A.2.2.2 Other tested odometry approaches

Besides VO based odometry, we also tried to implement the inertial measurement unit and the wheel odometry into our system. From our observations and tests however both of these sensor configurations experienced some major drawbacks. As opposed to the first phase of the competition, the IMU noise has significantly increased, making it more noisy than what one would traditionally experience with such a sensor. The resulting Visual Inertial Odometry (VIO) was found to be less accurate than simple visual odometry (both VinsFusion and OpenVins). This was likely due to a combination of the noisy IMU, as well as the simplicity of the gazebo environment, which led itself well to visual odometry estimates. Besides the VO, we also experimented with wheel odometry (WO). This approach worked well on flat areas, but was difficult to deal with on slopes, such as in the crater or hills. This was due to vibration introduced by the suspension of the robots. In the end we therefore reverted back to our original VO implementation.

A.2.2.3 Navigation

For the navigation part, we used move_base, which is a well known local planner. This tool took the generated cost maps and it produced a path using the A* algorithm. As our controller is using command velocity commands as an input, this can be directly connected to move_base. The local planning is then continuously updated with each update to the cost map. (Fig. A.14)

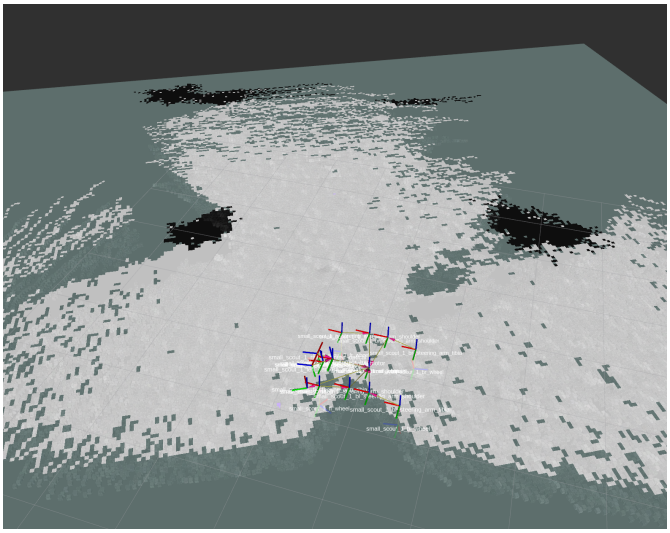

Figure A.14: Figure 13. Costmap showing the free terrain (white tiles), obstacles (black tiles) and unexplored terrain(gray tiles)

A.2.3 Planning

A.2.3.1 Volatile Searching and Mining Strategy

Before proceeding to design a waypoint generation scheme for the planner, we examined the relevant rules to find the best way to maximize the score. The relevant competition rules are listed below: There is no penalty for not collecting certain volatile types. Since the volatile zones are distributed evenly within a circular band with inner radius 28m and outer radius 93m, it is optimal to search the area close to the inner radius first before going to the outer area. The competition will contain 72 volatile locations, known as zones, which can be mined for volatile clods. Each volatile zone contains 20 clods of the appropriate volatile. Digging at the exact location of the volatile zone will get the maximum of 5 clods per scoop, but digging 2m away from the volatile zone gives only regolith. If the scoop is between 0m and 2m from the zone location, then the number of volatile clods in the scoop will be between 5 and 0 -

extra space on the scoop will be filled with regolith. If everything is perfect and the scoop is right over the volatile location, then 20/5 = 4 scoops are necessary to deplete the zone of volatiles. Some volatile types have mass thresholds which must be satisfied before volatile depositions into the hopper of that type are counted. For example, if 70kg of ice, or 7 clods, have been deposited cumulatively into the processing plant hopper, then only 1pt is contributed towards the Total Score. To emphasize, scoring for threshold-type volatiles starts from 0 after the threshold is met. After adjusting for the minimum threshold, analysis shows that Carbon Dioxide is slightly less valuable per clod (0.925 adjusted point/clod) compared to Ethane, Methane, or Methanol (1 adjusted point/clod) - the other types fall in between. This was calculated by dividing the [Volatile Field Points - Adjusted for Minimum Quantity] values by the [Volatile Clod/Types on Field] values.

After analyzing the points above, it was determined that since the differences between the adjusted points/clod are small enough, we can treat each clod of any material as being roughly worth 1 point. In addition, the threshold requirements for any volatile (maximum 9 clods for carbon dioxide) can be satisfied by fully mining the 20 clods in a single zone. Thus, a good strategy would be to treat all volatile types equally and focus on extracting 20 clods from each detected volatile zone. Choosing which volatile zone to target next should be done based on proximity to the excavator and capacity left in the hauler.

Two types of planners were created for the competition to provide the scouts with high-level goal waypoints during their search for volatiles. These consisted of a "predetermined" and "frontier" planner. In the end, the "predetermined" planner was used for the competition due to its smoother trajectories and simplicity, as it requires less input from the rest of the system to work.

The "predetermined" planner provides a scout rover with a predefined path resembling a "flower" pattern generated through a combination of multiple ellipses. If these goal waypoints are in a location that the rover cannot move to, the state machine queries the next point.

The "frontier" planner uses a more generic and complex approach to waypoint generation. It employs a 2D grid to represent the world and assigns a reward value to each cell, depending on the potential for discovering volatiles. The highest reward cell is the next goal location, and a new goal is queried either when the rover has arrived at the goal or when a new route is requested. The reward function is calculated based on the cell's proximity to large unexplored areas, the relative heading difference from the rover's current heading needed to travel there, its overall distance from the rover, and finally if a volatile has been detected via the rover's camera in that cell.

Once the state machine has a goal location from the "predetermined" planner, it conveys this request to the lower level planner, the standard ROS package "move_base". This is used for lower-level planning and obstacle avoidance. The output of this package was a linear and angular velocity command that was sent to the controller. In order to avoid obstacles, this package required a map as input from localization as well.

A.2.4 Control

For our controller, we use a combination of an Ackerman steering control and a spot turn control, which is used depending on the required angle of orientation. The controller was tweaked to improve the quality of the localization, and to provide a smooth driving experience. An earlier version of our controller resulted in numerous spot turn operations which

are less than ideal for localization accuracy. In some specific scenarios, like the approach to the processing plant by the hauler, some ROS-nodes were also given direct control over the wheels, in order to simplify the control loop

A.2.5 State Machine

The high level design of our solution is based around a state machine that manages the behavior of each of the rovers. When the high level system is in a certain state, each relevant rover is commanded to perform a specific action, such as ”drive to the next waypoint” or ”scoop regolith”. Further, each rover has its own state machine, which deals with low-level tasks for maneuvering itself, avoiding obstacles, reaching waypoints and recharging its battery. A global state machine deals with the more complex digging operation which requires the collaboration of all robots. The following two flow charts show the operations dealt with on a local and global level. (Fig. A.15) (Fig. A.16)

A.3 Results and Discussion

Below is a description of the tasks and the outcome of each task:

A.3.1 General Rover systems

1. Relative vision based navigation - SUCCESS
2. SLAM system with loop closure - SUCCESS
3. Gradient based obstacle detection - SUCCESS

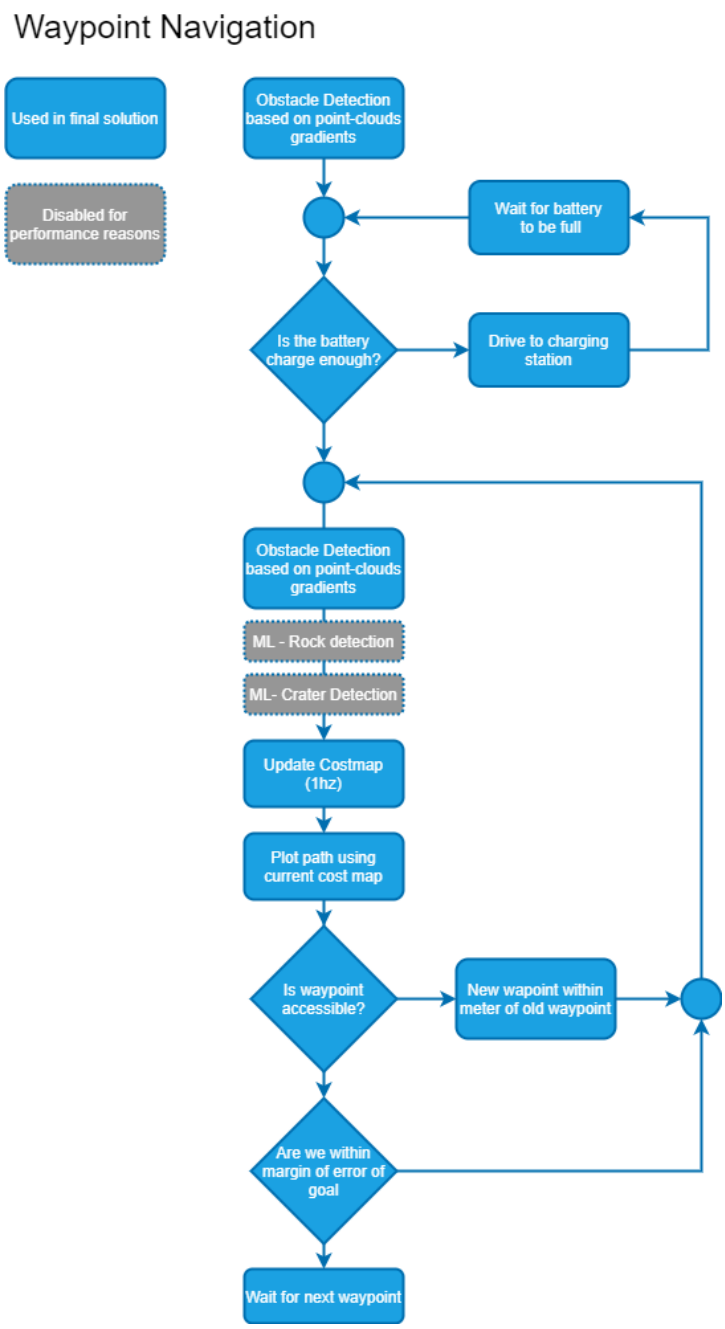

Figure A.15: Solution Rover State Machine

Figure A.16: Solution Global State Machine

4. ML based rock detection - SUCCESS*
5. ML based crater detection - SUCCESS*
6. ML object detector (Processing plant, recharding plant, rovers) - SUCCESS
7. Regular battery checks and recharding - SUCCESS
8. Processing plant pose estimation for relocalization - 50%
9. Waypoint based navigation - SUCCESS
10. Local and Global path planning and replanning with Costmap - SUCCESS
11. State machine to decide on rover behaviour - SUCCESS

*Note: Due to limited performance on hardware and because of uncertainty around the solution behavior on the final grading hardware, the ML rock and crater detection were disabled in the final submission.

A.3.2 Task specific systems

1. Scout search pattern for volatiles - SUCCESS
2. Detecting volatiles - SUCCESS
3. Determining exact volatile position relative to Scout - SUCCESS
4. Position Scout above volatile - SUCCESS
5. Position Excavator close to scout - SUCCESS

6. Determine volatile position relative to Excavator - SUCCESS
7. Position Excavator next to volatile, to enable digging - SUCCESS
8. Position Hauler in the field of view of the Excavator - SUCCESS
9. Position Hauler next to Excavator - 80% - would need to use rock detection (working) to determine if the field is clear, and reorient otherwise
10. Digging - 50% excavator control working but needs refinement
11. Hauler return to base - SUCCESS
12. Hauler aligning with processing plant and dumping volatiles - SUCCESS

A.4 Conclusions

A.4.1 Logistical Considerations

This challenge was complicated by our large and globally dispersed team, our team has 25 members across 7 countries. Our team had started out remotely which provided us with the ability to continue working uninterrupted where other teams may have had to have shifted with the impacts of COVID. The access to hardware required for the competition also proved to be a challenge for our team with different members having different systems and accessibility however cloud computing resources and docker containers were essential for our team to work in the remote distributed environment. Virtual tools like Zoom, Mattermost, Google docs/slides, gitlab were useful for our remote team to interact and work together.

A.4.2 Future Work

One interesting aspect of the competition was the integration of various machine learning techniques and the ROS integration, easing implementation could be an area of exploration to develop new workflows or frameworks.

Another consideration for future work would be to take the developed software solutions and to test them out in a real life environment. The simulator provided vital information and constraints but still falls short of a real world scenario. The actual output would also vary with the hardware implementation put on a real robotic system in a lunar-like environment and would be interesting to test out as a future phase of the competition and could result in some unique hardware solutions.

B

Scientific Dissemination

B.1 Publications & Posters

1. IAC 2018 - Bremen, DE

   Poster presentation with a 5 minute speaking slot.

   EXPLORATION OF THE LUNAR SOUTH POLE THROUGH AUTONOMOUS

NAVIGATION AND MAPPING SYSTEMS FOR MAXIMISING SCIENCE RETURN

https://iafastro.directory/iac/paper/id/46207/abstract-pdf/IAC-18,A3,IP,8,x46207.brief.pdf?2018-08-01.22:32:37

October 1–5, 2018

2. IROS 2019 - Macau

   Paper publication with 15 minute presentation slot.

   Absolute Localization Through Orbital Maps and Surface Perspective Imagery: A Synthetic Lunar Dataset and Neural Network Approach

   https://ieeexplore.ieee.org/abstract/document/8968124/

   Nov 4-8, 2019

3. ICRA / RAL 2020 - Paris, FR

   2nd Co-author on a paper publication with a 15 minute presentation slot. Converted to a virtual conference due to COVID19

   A Real-Time Approach for Chance-Constrained Motion Planning with Dynamic Obstacles.

   https://arxiv.org/pdf/2001.08012.pdf

   May 31, 2020 – Aug 31, 2020

4. IAC 2020 - Cyberspace Edition

   Paper publication with a 10 minute presentation slot, which was converted to a virtual presentation due to COVID19.

   Building a piece of the Moon: Construction of two indoor lunar analogue environ-

ments

https://orbilu.uni.lu/handle/10993/45539

October 12–14, 2020

5. ISAIRAS 2020 - Pasadena, US

   Paper publication with a 30 minute speaking slot. Converted to a virtual conference due to COVID19

   TESTING ENVIRONMENTS FOR LUNAR SURFACE PERCEPTION SYSTEMS; COMBINING INDOOR FACILITIES, VIRTUAL ENVIRONMENTS AND ANALOGUE FIELD TESTS.

   https://www.hou.usra.edu/meetings/isairas2020fullpapers/pdf/5078.pdf

   October 18–21, 2020

6. ISAIRAS 2020 - Pasadena, US

   Poster presentation with 15 minute speaking slot. Converted to a virtual conference due to COVID19

   Large scale realistic virtual environments for lunar robotics testing using real-time computer games engines

   https://www.hou.usra.edu/meetings/isairas2020/eposter/4071.pdf

   October 18–21, 2020

7. ISAIRAS 2020 - Pasadena, US

   Poster presentation with 15 minute speaking slot given by co-author. Converted to a virtual conference due to COVID19

   Absolute Localization for Surface Robotics in GPS-Denied Environments Using a

Neural Network.

https://www.hou.usra.edu/meetings/isairas2020/eposter/4032.pdf

October 18–21, 2020

8. Second AI and Data Science Workshop for Earth and Space Sciences - Pasadena, US

   Poster presentation with a 15 minute speaking slot. Converted to a virtual conference due to COVID19

   Absolute Localisation for surface robotics in GPS denied locations using a Neural Network https://datascience.jpl.nasa.gov/poster-27

   February 9–11, 2021

9. Space Resources Week - Luxembourg, LU

   Poster submission. Semi-virtual conference, unable to attend due to COVID19

   LunaLab lunar analogue testing facility

   https://cloud.list.lu/index.php/s/4jEnpqGPfodTo6o

   April 19–22, 2021

B.2 Awards

FNR Photo competitions 2019 & 2021

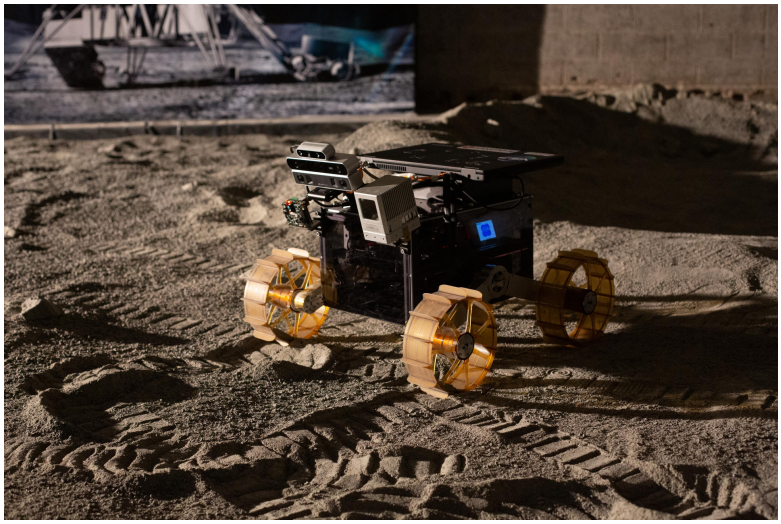

Figure B.1: 2019 1st prize - Category: Places and tools

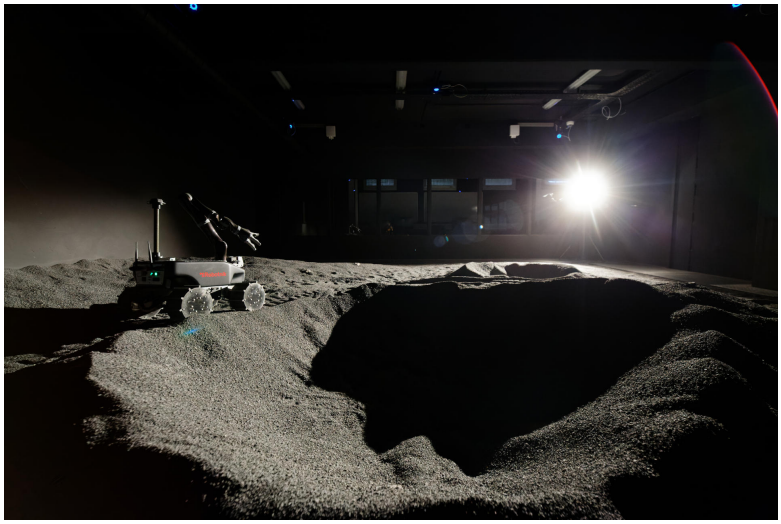

Figure B.2: 2021 1st prize - Category: Places and tools

Figure B.3: 2021 1st prize - Category: Scientists in action

B.3 Additional Grants

1. ESA-ISEB - IAC 2018 conference funding
2. NVIDIA - Applied Research Accelerator Program

B.4 Media and mentions

1. IEEE Spectrum

   https://spectrum.ieee.org/ai-trains-to-guide-planetary-rovers-without-gps

2. Intel chiptalk podcast

   https://connectedsocialmedia.com/17027/for-lunar-exploration-intel-ai-can-help-where-gps-

3. FNR article on research.

   https://www.fnr.lu/research-with-impact-fnr-highlight/lunar-rover-for-space-exploration/

4. FNR AR - Lunar rover in augmented reality.

   https://www.fnr.lu/letzscience-ar/

5. IGLUNA 2020 - Swiss space center

   https://www.youtube.com/watch?v=jcRb1sk_UYQ

6. NASA Frontier Development Lab

   https://www.youtube.com/watch?v=-ymnvxuCggw&feature=emb_title

7. FNR - What does it mean to be a researcher?

   https://youtu.be/N3lV_mOsjAs

8. ESERO - Ask a Space Expert:

   https://www.esero.lu/ask-a-space-expert/?lang=en

References

[1] C Acton, N Bachman, J Diaz Del Rio, B Semenov, E Wright, and Y Yamamoto. Spice: A means for determining observation geometry. In *meetingorganizer.copernicus.org*, 2011.

[2] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar. Yolo v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 687–694, 2020.

[3] DC Agle. News | NASA's Opportunity Rover Mission on Mars Comes to End. *www.nasa.gov*, 2019.

[4] Carolina Aiazzi, Abhinandan Jain, Aaron Gaut, Aaron Young, and Asher Elmquist. IRIS: High-fidelity Perception Sensor Modeling for Closed-Loop Planetary Simulations. In *AIAA SCITECH 2022 Forum*, pages 1–21, 2022.

[5] Carolina Aiazzi, Marco B. Quadrelli, Aaron Gaut, and Abhinandan Jain. Physics-based rendering of irregular planetary bodies for sensor modelling. In *i-SAIRAS 2020*, 2020.

[6] K.S. Ali, C.A. Vanelli, J.J. Biesiadecki, M.W. Maimone, A.M. San Martin, and J.W. Alexander. Attitude and Position Estimation on the Mars Exploration Rovers. *2005 IEEE International Conference on Systems, Man and Cybernetics*, 1:20–27, 2005.

[7] Mark Allan, Uland Wong, P. Michael Furlong, Arno Rogg, Scott McMichael, Terry Welsh, Ian Chen, Steven Peters, Brian Gerkey, Moraan Quigley, Mark Shirley, Matthew Deans, Howard Cannon, and Terry Fong. Planetary Rover Simulation for Lunar Exploration Missions. *IEEE Aerospace Conference Proceedings*, 2019-March, 2019.

[8] Da An, Alexander Woodward, Patrice Delmas, Georgy Gimel'farb, John Morris, and Jorge Marquez. Comparison of active structure lighting mono and stereo camera systems: Application to 3D face acquisition. *Proceedings of the Mexican International Conference on Computer Science*, pages 135–141, 2006.

[9] Yoshiyuki Anzai, Takehisa Yairi, Naoya Takeishi, Yuichi Tsuda, and Naoko Ogawa. Visual localization for asteroid touchdown operation based on local image features. *Astrodynamics*, 4(2):149–161, 2020.

[10] Georges Baatz, Olivier Saurer, Kevin Köser, and Marc Pollefeys. Large scale visual geo-localization of images in mountainous terrain. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7573 LNCS(PART 2):517–530, 2012.

[11] Bob Balaram, Timothy Canham, Courtney Duncan, Håvard F. Grip, Wayne Johnson, Justin Maki, Amelia Quon, Ryan Stern, and David Zhu. Mars Helicopter Technology Demonstrator. In *2018 AIAA Atmospheric Flight Mechanics Conference*, 2018.

[12] J. Balaram, Mi Mi Aung, and Matthew P. Golombek. The Ingenuity Helicopter on the Perseverance Rover. *Space Science Reviews*, 217(4):1–11, 2021.

[13] Mayank Bansal, Harpreet S. Sawhney, Hui Cheng, and Kostas Daniilidis. Geo-localization of street views with aerial image databases. *MM'11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops*, pages 1125–1128, 2011.

[14] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[15] HE Bender, MH Kahn, and TB Malone. An analysis of astronaut performance capability in the lunar environment. Volume 1-Performance problems and requirements for additional research. Technical report, NASA, 1969.

[16] Evangelos Boukas, Antonios Gasteratos, and Gianfranco Visentin. Localization of Planetary Exploration Rovers with Orbital Imaging: a survey of approaches. *Modelling, Estimation, Perception and Control of All Terrain Mobile Robots (WMEPC14), ICRA14*, 2014.

[17] Yannick Bukschat and Marcus Vetter. EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach. *arXiv preprint*, 11 2020.

[18] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[19] Carlos Campos, Richard Elvira, Juan J.Gomez Rodriguez, Jose M.M. Montiel, and Juan D. Tardos. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 7 2020.

[20] Patrick J.F. Carle, Paul T. Furgale, and Timothy D. Barfoot. Long-range rover localization by matching LIDAR scans to orbital elevation maps. *Journal of Field Robotics*, 27(3):344–370, 2010.

[21] Manuel Castillo-Lopez, Philippe Ludivig, Seyed Amin Sajadi-Alamdari, Jose Luis Sanchez-Lopez, Miguel A. Olivares-Mendez, and Holger Voos. A Real-Time Approach for Chance-Constrained Motion Planning with Dynamic Obstacles. *IEEE Robotics and Automation Letters*, 5(2):3620–3625, 2020.

[22] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. A Survey on Deep Learning for Localization and Mapping: Towards the Age of Spatial Machine Intelligence. *arXiv*, 2020.

[23] Ian Chen, Mark Allan, and Inc.) Inc.). Gazebo renders the moon. In *ROSCon Madrid 2018*, Mountain View, CA, 9 2018. Open Robotics.

[24] Jiale Chen, Lijun Zhang, Yi Liu, and Chi Xu. Survey on 6D Pose Estimation of Rigid Object. In *Chinese Control Conference, CCC*, volume 2020-July, pages 7440–7445. IEEE, 2020.

[25] Sebastiano Chiodini, Marco Pertile, Stefano Debei, Lorenzo Bramante, Enrico Ferrentino, Alfredo Giovanni Villa, Ivano Musso, and Marco Barrera. Mars rovers localization by matching local horizon to surface digital elevation models. In *4th IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace 2017 - Proceedings*, pages 374–379, 2017.

[26] Arup Roy Chowdhury, Manish Saxena, Ankush Kumar, S. R. Joshi, Amitabh, Aditya Dagar, Manish Mittal, Shweta Kirkire, Jalshri Desai, Dhrupesh Shah, J. C. Karelia, Anand Kumar, Kailash Jha, Prasanta Das, H. V. Bhagat, Jitendra Sharma, D. N. Ghonia, Meghal Desai, Gaurav Bansal, and Ashutosh Gupta. Orbiter high resolution camera onboard chandrayaan-2 orbiter. *Current Science*, 118(4):560–565, 2020.

[27] Anthony Colaprete. Volatiles Investigating Polar Exploration Rover. *ntrs.nasa.gov*, 2021.

[28] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

[29] Florian Cordes and Frank Kirchner. Heterogeneous Robotic Teams for Exploration of Steep Crater Environments. In *Planetary Rover Workshop, ICRA'10*, 2010.

[30] Fabio Cozman and Eric Krotkov. Robot localization using a computer vision sextant. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 1, page 106–111. IEEE, 1995.

[31] Fabio Cozman and Eric Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 3, pages 2452–2457, 1997.

[32] Matthew Cross, Kenneth A. McIsaac, Bryce Dudley, and William Choi. Negotiating corners with teleoperated mobile robots with time delay. *IEEE Transactions on Human-Machine Systems*, 48(6):682–690, 2018.

[33] Stan Darula, Richard Kittler, and Christian A. Gueymard. Reference luminous solar constant and solar luminance for illuminance calculations. *Solar Energy*, 79(5):559–565, 2005.

[34] Leonard David. NASA Moon Orbiter Fails to Spot India's Lunar Lander: Report | Space, 2019.

[35] Frank Dellaert. Factor Graphs and {GTSAM}. *Technical Report*, pages 1–27, 2012.

[36] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. BlenderProc. *arxiv.org*, 2019.

[37] Kaichang Di, Zhaoqin Liu, and Zongyu Yue. Mars rover localization based on feature matching between ground and orbital imagery. *Photogrammetric Engineering and Remote Sensing*, 77(8):781–792, 2011.

[38] Kaichang Di, Fengliang Xu, Jue Wang, Sanchit Agarwal, Evgenia Brodyagina, Rongxing Li, and Larry Matthies. Photogrammetric processing of rover imagery of the 2003 Mars Exploration Rover mission. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):181–201, 2008.

[39] Yassen Dobrev, Christoph Reustle, Tatiana Pavlenko, Florian Cordes, and Martin Vossiek. Mobile robot 6D pose estimation using a wireless localization network. In *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility, ICMIM 2016*, 2016.

[40] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. *proceedings.mlr.press*, 2017.

[41] Renaud Dube, Abel Gawel, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena. An online multi-robot SLAM system for 3D LiDARs. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2017-Septe, pages 1004–1011, 2017.

[42] Fabian Dubois, Louis Jerome Burtz, Oriol Gasquez, and Takahiro Miki. Finding the north on a lunar microrover: A lunar surface environment simulator for the development of vision-based navigation pipelines. *Proceedings of the International Astronautical Congress, IAC*, 2018-Octob:1–13, 2018.

[43] Martin N. Dunstan, Stephen M Parkes, Iain M Martin, Mark Mccrum, and Olivier Dubois-matra. Pangu : Planet and Asteroid Natural Scene Generation Utility. *4th International Conference on Astrodynamics Tools and Techniques*, pages 4–8, 2010.

[44] Allan R. Eisenman, Carl Christian Liebe, and Ramiro Perez. Sun sensing on the Mars exploration rovers. In *IEEE Aerospace Conference Proceedings*, volume 5, pages 2249–2262. IEEE, 2002.

[45] Richard C. Elphic, Jennifer L. Heldmann, Margarita M. Marinova, Anthony Colaprete, Erin L. Fritzler, Robert E. McMurray, Stephanie Morse, Ted L. Roush, Carol R. Stoker, Matthew C. Deans, and Trey F. Smith. Simulated real-time lunar volatiles prospecting with a rover-borne neutron spectrometer. *Advances in Space Research*, 55(10):2438–2450, 2015.

[46] Kynan Eng. Neuromorphic Vision in Space - EE Times Europe, 2021.

[47] John Enright, Tim Barfoot, and Marcela Soto. Star tracking for planetary rovers. In *IEEE Aerospace Conference Proceedings*, pages 1–13. IEEE, 3 2012.

[48] Epic Games. A first look at Unreal Engine 5, 2020.

[49] ESA. ESA - Galileo will help Lunar Pathfinder navigate around Moon, 2021.

[50] European Space Agency. Luna facility brings Moon to Earth, 2018.

[51] Matt Everingham and Nick Pelster. Lunar analog creation: Preparation and operation of a lunar regolith simulant testbed. In *AIAA Space 2009 Conference and Exposition*, pages 1–4, 2009.

[52] Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. A monocular pose estimation system based on infrared LEDs. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 907–913, 2014.

[53] Jinglun Feng, Chengjin Zhang, Bo Sun, and Yong Song. A fusion algorithm of visual odometry based on feature-based method and direct method. In *2017 Chinese Automation Congress (CAC)*. IEEE, 2017.

[54] Marcio A.A. Fialho and Daniele Mortari. Theoretical limits of star sensor accuracy†. *Sensors (Switzerland)*, 19(24):1–23, 2019.

[55] Terry Fong. Lunar Robotics Update. *NASA NTRS*, pages 1–32, 2018.

[56] Jordan Ford, Heather Jones, Khaled Sharif, and Red Whittaker. Technologies enabling the exploration of lunar pits. In *i-SAIRAS2020*, pages 1–6, 2020.

[57] Jordan Ford, Khaled Sharif, Heather Jones, and William Whittaker. Lunar Pit Exploration and Mapping via Autonomous Micro-Rover. In *2021 IEEE Aerospace Conference*, pages 1–7, 2021.

[58] Danilo Forte. *GNSS-based navigation in lunar environment and the Lunar Nearside Navigation Satellite System*. PhD thesis, POLITECNICO DI MILANO School, 2020.

[59] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part II: Matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, 19(2):78–90, 2012.

[60] Lawrence V. Fulton, Diane Dolezel, Jordan Harrop, Yan Yan, and Christopher P. Fulton. Classification of Alzheimer's Disease with and without Imagery Using Gradient Boosted Machines and ResNet-50. *Brain Sciences 2019, Vol. 9, Page 212*, 9(9):212, 8 2019.

[61] Paul Furgale, Pat Carle, John Enright, and Timothy D Barfoot. The Devon Island rover navigation dataset. *The International Journal of Robotics Research*, 31(6):707–713, 5 2012.

[62] Paul Furgale, John Enright, and Timothy Barfoot. Sun sensor navigation for planetary rovers: Theory and field testing. *IEEE Transactions on Aerospace and Electronic Systems*, 47(3):1631–1647, 7 2011.

[63] J. D. Gammell, Chi Hay Chi Hay Tong, P. Berczi, S. Anderson, T. D. Barfoot, and J. Enright. Rover odometry aided by a star tracker. In *2013 IEEE Aerospace Conference*, pages 1–10. IEEE, 3 2013.

[64] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 6 2014.

[65] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[66] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A Research Platform for Visual-Inertial Estimation. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4666–4672, 2020.

[67] David Gingras, Tom Lamarche, Pierre Allard, Natasha Jackson, Sébastien Gemme, Miranda Taylor, Chelsea Taylor, Chantelle Dubois, Guillaume Faubert, and Martin Picard. Lunar Exploration Analogue Deployment (Lead): Overview of the 2017-2019 Robotic Sample Return Mission Simulations. In *i-SAIRAS 2020*, 2020.

[68] Riccardo Giubilato, Sebastiano Chiodini, Marco Pertile, and Stefano Debei. An experimental comparison of ROS-compatible stereo visual SLAM methods for planetary rovers. *5th IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace 2018 - Proceedings*, pages 386–391, 2018.

[69] Riccardo Giubilato, Marco Pertile, and Stefano Debei. A comparison of monocular and stereo visual FastSLAM implementations. In *3rd IEEE International Workshop on Metrology for Aerospace, MetroAeroSpace 2016 - Proceedings*, pages 227–232, 2016.

[70] Joseph R. Guinn. Mars surface asset positioning using in-situ radio tracking. In *Advances in the Astronautical Sciences*, volume 108 I, pages 45–53, 2001.

[71] I Haase, J Oberst, F Scholten, M Wählisch, P Gläser, I Petrovna Karachevtseva, and M S Robinson. Mapping the Apollo 17 landing site area based on Lunar Reconnaissance Orbiter Camera images and Apollo surface photography. *Article in Journal of Geophysical Research Atmospheres*, 117:0–20, 2012.

[72] Antea Hadviger, Igor Cvišić, Ivan Marković, Sacha Vražić, and Ivan Petrović. Feature-based Event Stereo Visual Odometry. In *2021 European Conference on Mobile Robots (ECMR)*, 2021.

[73] Benjamin Hager. *Dynamic thermal modeling for moving objects on the Moon*. PhD thesis, Technische Universitat Munchen, 2013.

[74] Josh Handal, Karen Fox, and Katie Cousins. NASA's Artemis Rover to Land Near Nobile Region of Moon's South Pole, 2021.

[75] Bruce Hapke. Bidirectional reflectance spectroscopy: 1. Theory. *Journal of Geophysical Research: Solid Earth*, 86(B4):3039–3054, 1987.

[76] Andrew J. Hawkins. Waymo pulls back the curtain on 6.1 million miles of self-driving car data in Phoenix - The Verge.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, 12 2015.

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. *arXiv e-prints*, 3 2016.

[79] Christoph Heindl, Lukas Brunner, Sebastian Zambal, and Josef Scharinger. BlendTorch: A Real-Time, Adaptive Domain Randomization Library. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12664 LNCS:538–551, 10 2020.

[80] Robert A Hewitt, Evangelos Boukas, Martin Azkarate, Marco Pagnamenta, Joshua A Marshall, Antonios Gasteratos, and Gianfranco Visentin. The Katwijk beach planetary rover dataset. *International Journal of Robotics Research*, 37(1):3–12, 2018.

[81] Karl Hille. Obscured in the Lunar Highlands? | NASA, 2019.

[82] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[83] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.

[84] Ju Won Hwangbo, Kaichang Di, and Rongxing Li. Integration of orbital and ground image networks for the automation of rover localization. In *American Society for Photogrammetry and Remote Sensing Annual Conference 2009, ASPRS 2009*, volume 1, pages 344–356, 2009.

[85] Dmitri Ivanov and Domingos Fernandes. Thermal control of a light-weight rover system in the permanently shadowed regions of the lunar south pole. In *2020 International Conference on Environmental Systems*, pages 1–12. 2020 International Conference on Environmental Systems, 7 2020.

[86] A. Jäggi, U. Hugentobler, H. Bock, and G. Beutler. Precise orbit determination for GRACE using undifferenced or doubly differenced GPS data. *Advances in Space Research*, 39(10):1612–1619, 1 2007.

[87] Ramesh Jain, Rangachar Kasturi, and Brian G Schunck. *Machine vision*, volume 6. McGraw-Hill, 1995.

[88] Haidar Jamal. *Localization for Lunar Micro-Rovers*. PhD thesis, Carnegie Mellon University, 2021.

[89] JAXA. Space Exploration Experiment Building.

[90] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. Tsit: A simple and versatile framework for image-to-image translation. *Springer*, 2020.

[91] Andrew E. Johnson, Steven B. Goldberg, Yang Cheng, and Larry H. Matthies. Robust and efficient stereo feature tracking for visual odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 39–46, 2008.

[92] Heechul Jung, Min-Kook Choi, Jihun Jung, Jin-Hee Lee, Soon Kwon, and Woo Young Jung. ResNet-Based Vehicle Classification and Localization in Traffic Surveillance Systems, 2017.

[93] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Are we making real progress in simulated environments? measuring the sim2real gap in embodied visual navigation. *arXiv*, 2019.

[94] I. Karachevtseva, J. Oberst, F. Scholten, A. Konopikhin, K. Shingareva, E. Cherepanova, E. Gusakova, I. Haase, O. Peters, J. Plescia, and M. Robinson.

Cartography of the Lunokhod-1 landing site and traverse from LRO image and stereo-topographic data. *Planetary and Space Science*, 85:175–187, 2013.

[95] I. P. Karachevtseva, A. A. Kokhanov, N. A. Kozlova, and Zh F. Rodionova. Cartography of the Soviet Lunokhods' routes on the moon. *Lecture Notes in Geoinformation and Cartography*, 17(1971):263–278, 2019.

[96] Vadym Kaydash, Yuriy Shkuratov, Viktor Korokhin, and Gorden Videen. Photometric anomalies in the Apollo landing sites as seen from the Lunar Reconnaissance Orbiter. *Icarus*, 211(1):89–96, 2011.

[97] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154, 2004.

[98] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3:2149–2154, 2004.

[99] Dmitry Konovalov and Mangalam Sankupellay. Bird Call Recognition using Deep Convolutional Neural Network, ResNet-50 The Minke Whale Project View project Automatic Weight Estimation of Harvested Fish from Images View project Bird Call Recognition using Deep Convolutional Neural Network, ResNet-50. In *ACOUSTICS 2018*, 2018.

[100] Viktor Korokhin, Yuri Velikodsky, Yuriy Shkuratov, Vadym Kaydash, Urs Mall, and Gorden Videen. Using LROC WAC data for Lunar surface photoclinometry. *Planetary and Space Science*, 160(May):120–135, 2018.

[101] Eric Krotkov, Knud Henriksen, and Ralf Kories. Stereo Ranging with Verging Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1200–1205, 1990.

[102] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[103] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[104] Mathieu Labb and François Michaud. RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation. *Journal of Field Robotics*, 2018.

[105] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.

[106] Mathieu Labbé and François Michaud. Online global loop closure detection for large-scale multi-session graph-based SLAM. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2661–2666, 2014.

[107] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-view Multi-object 6D Pose Estimation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12362 LNCS, pages 574–591, 2020.

[108] S. Lacroix, A. De Maio, Q. Labourey, E. Paiva Mendes, P. Narvor, V. Bissonette, F. Souvannavong C. Bazerque, R Viards, and M. Azkarate. The Erfoud dataset: a comprehensive multi-camera and Lidar data collection for planetary exploration. In *ASTRA 2019*, 2019.

[109] Olivier Lamarre, Oliver Limoyo, Filip Marić, and Jonathan Kelly. The Canadian Planetary Emulation Terrain Energy-Aware Rover Navigation Dataset. *International Journal of Robotics Research*, 39(6):641–650, 3 2020.

[110] Olivier Lamarre, Oliver Limoyo, Filip Marić, and Jonathan Kelly. The Canadian Planetary Emulation Terrain Energy-Aware Rover Navigation Dataset. *International Journal of Robotics Research*, 39(6):641–650, 2020.

[111] John Lane, James Mantovani, Robert Mueller, M Nugent, A Nick, J Schuler, and I Townsend. Optical Extinction Measurements of Dust Density in the GMRO Regolith Test Bin. In *ASCE International Conference on Engineering, Science, Construction and Operations in Challenging Environments (Earth and Space 2016), Orlando, FL, USA, April*, volume 2011, pages 1–9, 2016.

[112] Jean Pierre Lebreton, Olivier Witasse, Claudio Sollazzo, Thierry Blancquaert, Patrice Couzin, Anne Marie Schipper, Jeremy B. Jones, Dennis L. Matson, Leonid I. Gurvits, David H. Atkinson, Bobby Kazeminejad, and Miguel Pérez-Ayúcar. An overview of the descent and landing of the Huygens probe on Titan. *Nature*, 438(7069):758–764, 2005.

[113] Pascal Lee. Mars on Earth: The NASA Haughton-Mars Project. *Ad Astra*, 2002.

[114] Sunny Leung and Oliver Montenbruck. Real-time navigation of formation-flying spacecraft using global-positioning-system measurements. *Journal of Guidance, Control, and Dynamics*, 28(2):226–235, 2005.

[115] Bin Li and Dimas Lima. Facial expression recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering*, 2:57–64, 6 2021.

[116] Chunlai Li, Wei Zuo, Weibin Wen, Xingguo Zeng, Xingye Gao, Yuxuan Liu, Qiang Fu, Zhoubin Zhang, Yan Su, Xin Ren, Fang Wang, Jianjun Liu, Wei Yan, Xu Tan, Dawei Liu, Bin Liu, Hongbo Zhang, and Ziyuan Ouyang. *Overview of the Chang'e-4 Mission: Opening the Frontier of Scientific Exploration of the Lunar Far Side*, volume 217. The Author(s), under exclusive licence to Springer Nature B.V. part of Springer Nature, 2021.

[117] Jimmy Li, Karim Koreitem, David Meger, and Gregory Dudek. View-Invariant Loop Closure with Oriented Semantic Landmarks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7943–7949, 2020.

[118] Rongxing Li, Shaojun He, Yunhang Chen, Min Tang, Pingbo Tang, Kaichang Di, Larry Matthies, Raymond E. Arvidson, Steven W. Squyres, Larry S. Crumpler, Tim Parker, and Michael Sims. MER Spirit rover localization: Comparison of ground image- and orbital image-based methods and science applications. *Journal of Geophysical Research E: Planets*, 116(2):0–16, 2011.

[119] C.C. Liebe. Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):587–599, 4 2002.

[120] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[121] Zhao Qin Liu, Kai Chang Di, Man Peng, Wen Hui Wan, Bin Liu, Li Chun Li, Tian Yi Yu, Bao Feng Wang, Jian Liang Zhou, and Hong Min Chen. High precision landing site mapping and rover localization for Chang'e-3 mission. *Science China: Physics, Mechanics and Astronomy*, 58(1):1–11, 2014.

[122] Zhao Qin Liu, Kai Chang Di, Man Peng, Wen Hui Wan, Bin Liu, Li Chun Li, Tian Yi Yu, Bao Feng Wang, Jian Liang Zhou, and Hong Min Chen. High precision landing site mapping and rover localization for Chang'e-3 mission. *Science China: Physics, Mechanics and Astronomy*, 58(1):1–11, 2014.

[123] Zhao Qin Liu et al. High precision landing site mapping and rover localization for Chang'e-3 mission. *Science China: Physics, Mechanics and Astronomy*, 58(1):1–11, 2014.

[124] M J Losekamm, S Barber, J Biswas, T Chupin, M Deiml, M Deremetz, A Evagora, M Glier, C Gscheidle, and K Kullack. Searching for Water Ice with the LUVMI-X Lunar Rover. In *The International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 1–5, 2020.

[125] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[126] Manolis Lourakis and Emmanouil Hourdakis. Planetary rover absolute localization by combining visual odometry with orbital image measurements. *Proc. of the 13th Symposium on Advanced Space Technologies in Automation and Robotics (ASTRA'15). European Space Agency*, 2015.

[127] Philippe Ludivig. Building a piece of the Moon: Construction of two indoor lunar analogue environments. In *71st International Astronautical Congress (IAC)*, pages IAC–20–A3.2B.3, 2020.

[128] Philippe Ludivig, Miguel Olivares-mendez, Abigail Calzada Diaz, Dmitry Ivanov, Holger Voos, and Julien Lamamy. Testing Environments for Lunar Surface Perception Systems ; Combining Indoor Facilities , Virtual Environments and Analogue Field Tests . In *i-SAIRAS 2020*, 2020.

[129] Philippe Ludivig, Holger Voos, and Julien Lamamy. Large scale realistic virtual environments for lunar robotics testing using real-time computer games engines. In *iSAIRAS, Virtual*, 2020.

[130] Lunar Reconnaissance Orbiter Project Lunar Geodesy and Cartography Working Group. A standardized lunar coordinate system for the Lunar Reconnaissance Orbiter and lunar datasets: LRO Project and LGCWG White Paper, 2008.

[131] You Qing Ma, Shao Chuang Liu, Bing Sima, Bo Wen, Song Peng, and Yang Jia. A precise visual localisation method for the Chinese Chang'e-4 Yutu-2 rover. *Photogrammetric Record*, 35(169):10–39, 2020.

[132] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.

[133] J. N. Maki, D. Gruel, C. McKinney, M. A. Ravine, M. Morales, D. Lee, R. Willson, D. Copley-Woods, M. Valvo, T. Goodsall, J. McGuire, R. G. Sellar, J. A. Schaffner, M. A. Caplinger, J. M. Shamah, A. E. Johnson, H. Ansari, K. Singh, T. Litwin, R. Deen, A. Culver, N. Ruoff, D. Petrizzo, D. Kessler, C. Basset, T. Estlin, F. Alibay, A. Nelessen, and S. Algermissen. The Mars 2020 Engineering Cameras and Microphone on the Perseverance Rover: A Next-Generation Imaging System for Mars Exploration. *Space Science Reviews*, 216(8), 2020.

[134] Iain M Martin, Martin N Dunstan, and Manuel Sanchez Gestido. Planetary Surface Image Generation for Testing Future Space Missions With Pangu. *2nd RPI Space Imaging Workshop*, 2019.

[135] Erwan Mazarico, · D D Rowlands, G A Neumann, · D E Smith, · M H Torrence, F G Lemoine, · M T Zuber, E Mazarico, and M H Torrence. Orbit determination of the Lunar Reconnaissance Orbiter. *Journal of Geodesy*, 86:193–207, 2011.

[136] M McCrum, S Parkes, I Martin, and M Dunstan. Mars Visual Simulation for ExoMars Navigation Algorithm Validation. *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 283–290, 2010.

[137] Alfred S. McEwen, Eric M. Eliason, James W. Bergstrom, Nathan T. Bridges, Candice J. Hansen, W. Alan Delamere, John A. Grant, Virginia C. Gulick, Kenneth E. Herkenhoff, Laszlo Keszthelyi, Randolph L. Kirk, Michael T. Mellon, Steven W. Squyres, Nicolas Thomas, and Catherine M. Weitz. Mars Reconnaissance Orbiter's High Resolution Imaging Science Experiment (HiRISE). *Journal of Geophysical Research*, 112(E5):E05S02, 5 2007.

[138] Benjamin J. McLoughlin, Harry A.G. Pointon, John P. McLoughlin, Andy Shaw, and Frederic A. Bezombes. Uncertainty characterisation of mobile robot localisation techniques using optical surveying grade instruments. *Sensors (Switzerland)*, 18(7), 2018.

[139] Paul Meacham, Nuno Silva, and Richard Lancaster. The Development of the Locomotion Performance Model (LPM) for the ExoMars Rover Vehicle. *ASTRA 2013 - 12th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2013.

[140] Benjamin J. Mellinkoff, Matthew M. Spydell, Wendy Bailey, and Jack O. Burns. Quantifying operational constraints of low-latency telerobotics for planetary surface operations. In *IEEE Aerospace Conference Proceedings*, volume 2018-March, pages 1–10, 2018.

[141] Alexey Merzlyakov and Steve Macenski. A Comparison of Modern General-Purpose Visual SLAM Approaches. *arXiv preprint*, 2021.

[142] Lukas Meyer, Michal Smíšek, Alejandro Fontan Villacampa, Laura Oliva Maza, Daniel Medina, Martin J. Schuster, Florian Steidle, Mallikarjuna Vayugundla, Marcus G. Müller, Bernhard Rebele, Armin Wedler, and Rudolph Triebel. The MADMAX data set for visual-inertial rover navigation on mars. *Journal of Field Robotics*, -(August 2020):1–21, 2021.

[143] A. H. Mishkin, J. C. Morrison, T. T. Nguyen, H. W. Stone, B. K. Cooper, and B. H. Wilcox. Experiences with operations and autonomy of the Mars Pathfinder Microrover. *IEEE Aerospace Conference Proceedings*, 2:337–351, 1998.

[144] Sherif A.S. Mohamed, Mohammad Hashem Haghbayan, Tomi Westerlund, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. A Survey on Odometry for Autonomous Navigation Systems. *IEEE Access*, 7:97466–97486, 2019.

[145] Rafael Muñoz-Salinas and R. Medina-Carnicer. UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. *Pattern Recognition*, 101:107193, 5 2020.

[146] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[147] Omkar Nikam. Opportunities Emerging from "New Space", 2018.

[148] David Nistér and James Bergen. Visual Odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004.

[149] Clark F. Olson, Habib Abi-Rached, Ming Ye, and Jonathan P. Hendrich. Wide-Baseline Stereo Vision for Mars Rovers. *IEEE International Conference on Intelligent Robots and Systems*, 2(October):1302–1307, 2003.

[150] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011.

[151] Luis E. Ortiz, Elizabeth V. Cabrera, and Luiz M. Gonçalves. Depth data error modeling of the ZED 3D vision sensor from stereolabs. *Electronic Letters on Computer Vision and Image Analysis*, 17(1):1–15, 2018.

[152] S M Parkes, M Dunstan, D Matthews, I Martin, and V Silva. LIDAR-based GNC for Planetary Landing: Simulation with PANGU. In *European Space Agency, (Special Publication) ESA SP*, pages 151–162, 2003.

[153] Sida Peng, Xiaowei Zhou, Yuan Liu, Haotong Lin, Qixing Huang, and Hujun Bao. PVNet: Pixel-wise Voting Network for 6DoF Object Pose Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[154] Andrew Petro. Surviving and Operating Through the Lunar Night. In *IEEE Aerospace Conference Proceedings*, 2020.

[155] Adrien POTIER, Toshinori KUWAHARA, Alperen PALA, Shinya FUJITA, Yuji SATO, Yoshihiko SHIBUYA, Hannah TOMIO, Pasith TANGHANAKANOND, Tomoyuki HONDA, Tomomasa SHIBUYA, Kamachi KOH, and Hiroki UTO. Time-of-Flight Monitoring Camera System of the De-orbiting Drag Sail for Microsatellite ALE-1. *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, 19(5):774–783, 2021.

[156] Tereza Pultarova. Europe unveils plans to bring 'GPS' and Skype to the moon with satellites | Space, 2021.

[157] Michael E. Purucker. A global model of the internal magnetic field of the Moon based on Lunar Prospector magnetometer observations. *Icarus*, 197(1):19–23, 9 2008.

[158] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors. *arXiv preprint arXiv:1901.03638*, 2019.

[159] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[160] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:779–788, 6 2015.

[161] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[162] Renuganth, Stoll Enrico, Jaekel Steffen, Katz Jacob, Saenz-Otero Alvar, and Varatharajoo. Planetary Monocular Simultaneous Localization and Mapping. *Journal of Field Robotics*, 2015.

[163] Stephan R. Richter, Hassan Abu AlHaija, and Vladlen Koltun. Enhancing Photorealism Enhancement. *arXiv*, 5 2021.

[164] M. S. Robinson, S. M. Brylow, M. Tschimmel, D. Humm, S. J. Lawrence, P. C. Thomas, B. W. Denevi, E. Bowman-Cisneros, J. Zerr, M. A. Ravine, M. A. Caplinger, F. T. Ghaemi, J. A. Schaffner, M. C. Malin, P. Mahanti, A. Bartels, J. Anderson, T. N. Tran, E. M. Eliason, A. S. McEwen, E. Turtle, B. L. Jolliff, and H. Hiesinger. Lunar reconnaissance orbiter camera (LROC) instrument overview. *Space Science Reviews*, 150(1-4):81–124, 1 2010.

[165] Zh F. Rodionova and E. A. Kozlova. Morphological analysis of the cratering of the south pole-aitken basin on the moon. *Solar System Research*, 34(5):390–397, 2000.

[166] Seth Roffe, Himanshu Akolkar, and Alan D George. on Neuromorphic Event-based Vision Sensor : A First Step Towards Space Applications. *arXiv*, 2017.

[167] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1689–1696, 2020.

[168] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: and efficient alternative to SIFT and SURF. In *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011.

[169] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152, 2001.

[170] Malak A. Samaan, Stephen R. Steffes, and Stephan Theil. Star tracker real-time hardware in the loop testing using optical star simulator. In *Advances in the Astronautical Sciences*, volume 140, pages 2233–2245, 2011.

[171] Lucimara C N Scaduto, Erica G Carvalho, Lucas F Santos, F M Yasuoka, Mário A Stefani, and Jarbas C Castro. Baffle design and analysis of stray-light in multispectral camera of a Brazilian satellite. *Annals of Optics, XXIX ENFMC*, 2006.

[172] Davide Scaramuzza and Friedrich Fraundorfer. Tutorial: Visual odometry. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[173] Christoph Schmitt, Johannes Both, and Florian Kolb. RVS3000-3D : LIDAR meets Neural Networks. *isairas2018*, 2018.

[174] Martin J. Schuster, Marcus G. Muller, Sebastian G. Brunner, Hannah Lehner, Peter Lehner, Ryo Sakagami, Andreas Domel, Lukas Meyer, Bernhard Vodermayer, Riccardo Giubilato, Mallikarjuna Vayugundla, Josef Reill, Florian Steidle, Ingo Von Bargen, Kristin Bussmann, Rico Belder, Philipp Lutz, Wolfgang Sturzl, Michal Smisek, M. Moritz, Samantha Stoneman, Andre Fonseca Prince, Bernhard Rebele, Maximilian Durner, Emanuel Staudinger, Siwei Zhang, Robert Pohlmann, Esther Bischoff, Christian Braun, Susanne Schroder, Enrico Dietz, Sven Frohmann, Anko Borner, Heinz Wilhelm Hubers, Bernard Foing, Rudolph Triebel, Alin O. Albu-Schaffer, and Armin Wedler. The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration. *IEEE Robotics and Automation Letters*, 5(4):5315–5322, 2020.

[175] Martin J. Schuster, Korbinian Schmid, Christoph Brand, and Michael Beetz. Distributed stereo vision-based 6D localization and mapping for multi-robot teams. *Journal of Field Robotics*, 36(2):305–332, 2019.

[176] Gerald Schweighofer and Axel Pinz. Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, 2006.

[177] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. *Artificial Intelligence, Robotics and Automation in Space, Proceedings of the Fifth International Symposium, ISAIRAS '99*, pages 621–635, 2018.

[178] Aashish Sheshadri, Kevin M. Peterson, Heather L. Jones, and William L. Red Whittaker. Position estimation by registration to planetary terrain. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 432–438. IEEE, 9 2012.

[179] Bruno Siciliano and Oussama Khatib. Springer handbook of robotics. *Springer Handbook of Robotics*, pages 1–2227, 1 2016.

[180] Asif A. Siddiqi. Deep Space Chronicle: A Chronology of Deep Space and Planetary Probes 1958-2000. Technical Report 24, National Aeronautics and Space Administration, 2002.

[181] Deborah A. Sigel and David Wettergreen. Star tracker celestial localization system for a lunar rover. *IEEE International Conference on Intelligent Robots and Systems*, pages 2851–2856, 2007.

[182] O Sikorski, D Izzo, and Gabriele Meoni. Event-Based Spacecraft Landing Using Time-To-Contact. In *CVPR 2021 workshop paper*, 2021.

[183] Nuno Silva, Richard Lancaster, and Jim Clemmet. ExoMars Rover Vehicle Mobility Functional Architecture and Key Design Drivers. *ASTRA Conference 2013*, 2013.

[184] S Slojkowski. Lunar Reconnaissance Orbiter Orbit Determination Accuracy Analysis. Technical Report 1, NASA, 2018.

[185] Solar MEMS Technologies. NanoSSOC | Space Equipment | Solar MEMS Solar Sensor.

[186] Emanuel Staudinger, Dmitriy Shutin, Christoph Manß, Alberto Viseras Ruiz, and Siwei Zhang. Swarm Technologies for Future Space Exploration Missions. In *i-SAIRAS 2018*, 2018.

[187] Braden Stefanuk, Melissa Battler, Kaizad Raimalwala, Adam Deslauriers, Michele Faragalli, Krzysztof Skonieczny, Evan Smal, and Ewan Reid. Lunar Analogue Dataset for Traversability Assessment and Novelty Detection. In *ASCEND 2021*, 2021.

[188] Hauke Strasdat, J M M Montiel, and Andrew J Davison. Real-time Monocular SLAM: Why Filter? In *2010 IEEE International Conference on Robotics and Automation*, pages 2657–2664, 2010.

[189] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[190] Guodong Sun, Yan Wang, Lin Gu, and Zhenzhong Liu. An improved ICP algorithm for point cloud registration. *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics, ICARM 2021*, pages 582–585, 2021.

[191] Tomoji Takasu and Akio Yasuda. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. *International Symposium on GPS/GNSS*, pages 4–6, 2009.

[192] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:10691–10700, 5 2019.

[193] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787, 2020.

[194] Guoliang Tang, Zhijing Liu, and Jing Xiong. Distinctive image features from illumination and scale invariant keypoints. *Multimedia Tools and Applications*, 78(16):23415–23442, 8 2019.

[195] Catherine L Thornton and James S Border. *Radiometric Tracking Techniques for Deep-Space Navigation*. John Wiley & Sons, 2003.

[196] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 3 2002.

[197] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming Down to Earth: Satellite-to-Street View Synthesis for Geo-Localization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3 2021.

[198] Chi Hay Tong, David Gingras, Kevin Larose, Timothy D. Barfoot, and Érick Dupuis. The Canadian planetary emulation terrain 3D mapping dataset, 2013.

[199] Daniel Townson, Mark Woods, and Stuart Carnochan. Exomars Visloc – the Industrialised , Visual Localisation System for the Exomars Rover. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2018)*, page 7, 2018.

[200] Tzutalin. LabelImg, 2015.

[201] Kentaro UNO, Louis-Jerome BURTZ, Marc HULCELLE, and Kazuya YOSHIDA. Qualification of a Time-of-Flight Camera as a Hazard Detection and Avoidance Sensor for a Moon Exploration Microrover. *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, 16(7):619–627, 2018.

[202] Mallikarjuna Vayugundla, Florian Steidle, Michal Smisek, Martin J. Schuster, Kristin Bussmann, and Armin Wedler. Datasets of long range navigation experiments in a moon analogue environment on Mount Etna. In *50th International Symposium on Robotics, ISR 2018*, pages 77–83, 2018.

[203] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[204] Richard Volpe. Mars Rover Navigation Results Using Sun Sensor Heading Determination. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, volume 1, pages 460–467. IEEE, 1999.

[205] John Walker. Flight System Architecture of the Sorato Lunar Rover. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2018)*, 2018.

[206] Jia Wang, Yu Zhang, Kaichang Di, Ming Chen, Jianfeng Duan, Jing Kong, Jianfeng Xie, Zhaoqin Liu, Wenhui Wan, Zhifei Rong, Bin Liu, Man Peng, and Yexin Wang. Localization of the chang'e-5 lander using radio-tracking and image-based methods, 2021.

[207] Benjamin P. Weiss and Sonia M. Tikoo. The lunar dynamo. *Science*, 346(6214), 2014.

[208] Brianna Wessling. CMU's Iris moon rover prepares for launch in 2022, 12 2021.

[209] David S. Wettergreen, Greydon Foil, Michael Furlong, and David R. Thompson. Science Autonomy for Rover Subsurface Exploration of the Atacama Desert. *AI Magazine*, 35(4):47–61, 12 2014.

[210] Wikipedia. Odometry, 2021.

[211] J. P. Williams, D. A. Paige, B. T. Greenhagen, and E. Sefton-Nash. The global surface temperatures of the moon as measured by the diviner lunar radiometer experiment. *Icarus*, 283:300–325, 2017.

[212] Paweł Wittels, Grzegorz Gawdzik, Marlena Gołofit, Piotr Jaroszek, Jakub Kra, Sławomir Kurdziel, Mateusz Macia, and Sarmad Aziz. Racer : Determination of the Maximum Speed of the Fast Teleoperated Rover for Lunar Exploration. In *isairas2018*, 2018.

[213] Natalie Wolchover. NASA Gives Up On Stuck Mars Rover Spirit. *www.space.com*, 2011.

[214] Uland Wong, Ara Nefian, Larry Edwards, Xavier P. Buoyssounouse, Michael Furlong, Matt Deans, and Terry Fong. Polar Optical Lunar Analog Reconstruction (POLAR) Stereo Dataset, 2017.

[215] Uland Wong, Ara Nefian, Larry Edwards, Michael Furlong, Xavier Bouyssounouse, Vinh To, Matthew Deans, Howard Cannon, and Terry Fong. Characterization of Stereo Vision Performance for Roving at the Lunar Poles, 2016.

[216] Benjamin Wu, Potter Ross W. K., Philippe Ludivig, Andrew S. Chung, and Timothy Seabrook. Absolute Localization Through Orbital Maps and Surface Perspective Imagery: A Synthetic Lunar Dataset and Neural Network Approach. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3262–3267, 2020.

[217] Martin Wudenka, Marcus G Müller, Nikolaus Demmel, Armin Wedler, Rudolph Triebel, Daniel Cremers, and Wolfgang Stürzl. Towards Robust Monocular Visual Odometry for Flying Robots on Planetary Missions. *arXiv preprint arXiv:2109.05509*, 2021.

[218] Kaan Yilmaz, Baris Suslu, Sohini Roychowdhury, and L. Srikar Muppirisetty. AV-SLAM: Autonomous vehicle SLAM with gravity direction initialization. In *Proceedings - International Conference on Pattern Recognition*, pages 8093–8100. IEEE, 2020.

[219] Herbert Zettl. The Three-Dimensional Field: Detph and Volume. In *Sight, sound, motion: Applied media aesthetics*, pages 153–170. Wadsworth Publishing Company, 2005.

[220] Ji Zhang and Sanjiv Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems*, 2014.

[221] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 5816–5824, 2017.

[222] Maria T Zuber, David E Smith, Ronald S Zellar, Gregory A Neumann, · Xiaoli, Sun · Richard, B Katz, Igor Kleyner, Adam Matuszeski, Jan F Mcgarry, Melanie N Ott, Luis A Ramos-Izquierdo, David D Rowlands, Mark H Torrence, Thomas W Zagwodzki, M T Zuber, D E Smith, G A Neumann, · X Sun, · J F Mcgarry, · D D Rowlands, T W Zagwodzki, R S Zellar, · R B Katz, · I Kleyner, · A Matuszeski, · M N Ott, L A Ramos-Izquierdo, and M H Torrence. The Lunar Reconnaissance Orbiter Laser Ranging Investigation. *Space Sci Rev*, 150:63–80, 2010.