Eötvös Loránd University

Faculty of Informatics

Erasmus Mundus Joint Master in

Intelligent Field Robotic Systems

# Vision aided Autonomous Power Descent Guidance for Planetary Soft-Landing

*Author:*

Ulugbek Alibekov

*Internal supervisor:*

Istenes Zoltán

Associate Professor

*External supervisor:*

Sumeet Gajanan Satpute

Postdoctoral Researcher

*Lulea, 2023*

STATEMENT

OF THESIS SUBMISSION AND ORIGINALITY

I hereby confirm the submission of the Master Thesis Work on the IFRoS MSc course with author and title:

Name of Student: Ulugbek Alibekov

Code of Student: R2TUAR

Title of Thesis: Vision aided Autonomous Power Descent Guidance for Planetary Soft-Landing

Supervisor: Istenes Zoltán

at Eötvös Loránd University, Faculty of Informatics.

In consciousness of my full legal and disciplinary responsibility I hereby claim that the submitted thesis work is my own original intellectual product, the use of referenced literature is done according to the general rulesof copyright.

I understand that in the case of thesis works the following acts are considered plagiarism:

• literal quotation without quotation marks and reference;

• citation of content without reference;

• presenting others' published thoughts as own thoughts.

Budapest, 2023.

..........................................

student

**EÖTVÖS LORÁND UNIVERSITY**

**FACULTY OF INFORMATICS**

# MASTER THESIS TOPIC DECLARATION FORM

**Name of Student**: **Ulugbek Alibekov**

Neptun code: R2TUAR

Training: Full-time

Technical major: Erasmus Mundus Joint Master Degree in Intelligent Field Robotic System

Email Address: ulugbek.alibekov@nu.edu.kz

Phone Number: +36-70-557-29-36

**Name of ELTE supervisor: Istenes Zoltán**

Department: Faculty of Informatics

Contact (email, phone): istenes@inf.elte.hu

**Name of Industrial supervisor: Sumeet Gajanan Satpute**

Department: Computer Science, Electrical and Space Engineering

Contact (email, phone): sumeet.gajanan.satpute@ltu.se

**Information about the internship**

Name of the company: Luleå University of Technology

Starting date of internship: 1st February 2023

Closing date of internship: 15th May 2023

Weekly schedule: 24 hours per week

---

*The purpose of the admission declaration is to certify that the student of the MSc in Intelligent Field Robotics System at ELTE Faculty of Informatics may complete the mandatory internship in the selected institution within the framework detailed hereby and in accordance with the learning outcomes required by the program.*

**Title of the thesis**

**Vision aided Autonomous Power Descent Guidance for Planetary Soft-Landing**

**Topic of the thesis (1 – 1,5 pages)**

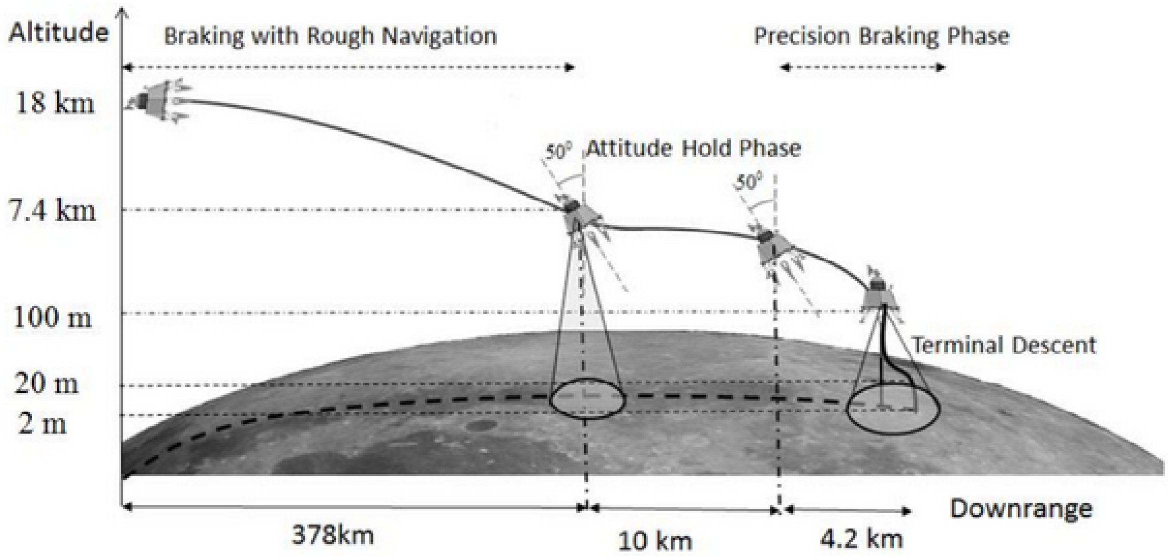



Figure 1: Schematic of a vision aided multiphase planetary soft-landing

Aiming towards the development of a long-term research base for scientific exploration of futuristic objectives, autonomous safe landing of a space vehicle over an unknown territory of a celestial body is a fundamental prerequisite. With every new planetary mission there has been a constant progress in innovative and efficient autonomy stack for performing soft- landing with pinpoint accuracy.

Here in this master thesis, the goal will be :

- To develop a framework in physics-based simulation engine with necessary sensor suites, that emulates and demonstrate the capability of terrain based autonomous visual navigation

- To develop a vision-based algorithm that can process the captured images of the local terrain to progressively localise and identify a relatively safe landing sites as the spacecraft approach during the descent phase

- To develop an autonomous guidance algorithm that drives the spacecraft to enable soft-landing with touchdown velocity at designated landing sites.

An outcome of this thesis will be a functional autonomy stack to demonstrate vision aided landing scenarios in physics-based simulation platforms.

Important background knowledge needed are engineering mathematics, familiarity with Linux, ROS, Gazebo/Unity, MatLab/Python.

Encryption of the topic is necessary: **YES/NO**

I ask for the acceptance of my thesis topic.

Budapest, 09/01/2023

.....................................................

Student

I approve of the suggested topic of the Master's Thesis:

Budapest, 19/1/2023 .....................................................

ELTE supervisor

I approve of the suggested topic of the internship, and in the name of the institution (organization, company) above I agree, that the named student will carry out his/her internship along the conditions detailed above:

Luleå , 11/01/2023 .....................................................

Industrial supervisor

The topic of the thesis and internship is approved by the Dean of Faculty of ELTE Informatics

Budapest, …./…./2023

.....................................................

**Dr. Tamás Kozsik**

Dean of Faculty of Informatics

# Contents

# Acknowledgements

I am grateful to the IFRoS committee for granting me the opportunity to become a part of the vibrant robotics community at the University of Girona, Eötvös Loránd University, and Luleå University of Technology. These two years have been filled with remarkable experiences, both in terms of academic pursuits and personal growth.

I would like to say thank you to my professors for their support and for providing me with the necessary resources to enhance my learning and research. Their guidance and expertise have been invaluable in shaping my academic journey.

I am also thankful to my groupmates for their continuous encouragement throughout this journey. Their collaborative spirit and passion for robotics have been a constant source of inspiration.

I would like to express my gratitude to my supervisors, Istenes Zoltán and Sumeet Gajanan Satpute, for their guidance, patience, and mentorship during the course of my master's thesis.

Lastly, I am grateful to my family and friends for their constant support, understanding and encouragement.

I am truly happy to have opportunity to be a part of this academic community and to be surrounded by such incredible individuals. Thank you all for being a part of this chapter in my life.

# Chapter 1

# Introduction

Have you ever looked up at the sky and wondered at the vastness of space? If so, you are not alone. Space exploration has long been a fascinating and mysterious subject, and it continues to captivate the imaginations of people all around the world to this day. Traditionally, telescopes have been the primary tool for exploring the objects closest to humans in the cosmos. However, due to their short working distance, they can only provide a limited amount of information [1]. To gain a deeper understanding of celestial bodies and their environments, planetary exploration missions have become an essential component of space exploration. By deploying spacecraft to conduct close-up observations and measurements, scientists can gather more detailed and precise data that cannot be obtained from telescopes alone.

Navigating and choosing a safe landing spot during a planetary landing mission can be one of the most challenging tasks. Many recent planetary missions have relied on either operator control or predefined maps to accomplish this. For instance, the Japanese Hayabusa2 sample return mission's initial landing stage was manually controlled by an operator [2], while the Mars rover Perseverance, launched in July 2020, relied on satellite images to identify a safe landing spot on the Martian surface [3]. Despite the success of both missions, these methods may not be suitable for future missions to distant space objects. Therefore, the implementation of an autonomous system for navigation and hazard avoidance is essential.

## 1.1 Background and context

Starting from the first human step taken on a celestial body, significant progress has been made in space exploration and research aimed at discovering new worlds. However, a question arises as to why studying and exploring space objects is important. The answer lies in several reasons, including scientific curiosity, resource exploration, human survival, and the search for extraterrestrial life. Regardless of the underlying motive, it is evident that humanity has been and will continue to be driven by the discover secrets held within the vast expanse of space

One recent project related to space exploration is the launch of the Mars rover Perseverance, which aims to study the Martian surface and gather geological samples. Equipped with a drone capable of investigating terrain from a high altitude, the mission seeks to gain valuable information about the planet's geology and climate,laying the foundation for future projects aimed at making Mars habitable for human life [4]. Another exciting project in the works is NASA's planned mission to Europa, one of Jupiter's icy moons, scheduled for 2027 [5]. The mission is focused on landing on the moon's surface to study its icy crust and potential life signs, providing valuable insights into the conditions and potential for habitability on other celestial bodies.

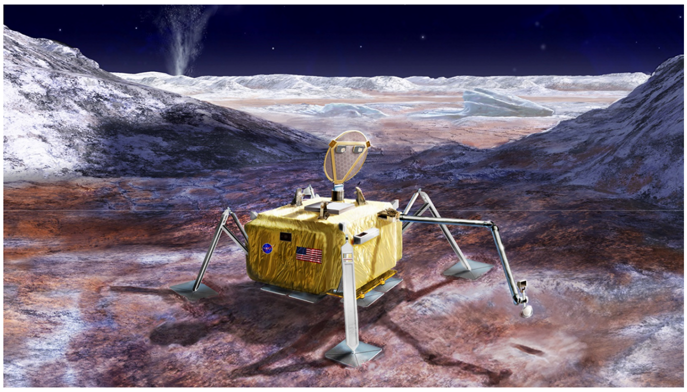


Figure 1.1: Europa mission lander.

Both the Mars rover mission and the planned Europa lander are unmanned spacecraft that must operate autonomously throughout their missions. To successfully land on a space object, such as a planet or moon, a lander must complete several critical stages, including exiting the Earth's atmosphere, navigating to the celestial body, entering the space object's atmosphere, descending through the atmosphere, landing safely on the surface, and performing surface operations. While each of these stages is important, this master thesis will focus specifically on the challenges and techniques associated with the critical stages of descending and landing on a space object.

Safe planetary landing is a critical aspect of robotic missions in space exploration, and one of the most challenging tasks involved is hazard avoidance. To ensure a safe landing, a spacecraft must be equipped with sophisticated navigation and hazard detection systems, which can identify potential obstacles such as rocks, cliffs, craters, and hills, and select the best possible landing site. This task becomes even more challenging when attempting to land on distant objects, as there is often limited information available about the terrain and surface conditions.

Scientists have proposed various methods to address the challenges of safe planetary landing, including the use of a wide variety of cameras and sensors. However, there is still room for improvement as some of these approaches rely on predefined maps or are not adaptable to quickly changing atmospheric conditions [6]. As a result, this master thesis aims to propose a novel method for safe planetary landing that can be simulated and tested in different atmospheric conditions using the Unity game engine.

## 1.2 Problem statement

The autonomous landing phase is a crucial step in future unmanned planetary missions, wherein the spacecraft should make independent decisions based on environmental data received from its sensors. Previous landing missions relied on predefined maps or human control from Earth, but these methods are ineffective for exploring distant objects. As the mission distance increases, the quality of obtained maps for the terrain deteriorates, leading to potential obscurity of surface hazards such as craters, mountains, and hills. In addition, the signal propagation worsens between the command center and the spacecraft. Thus, the development of autonomous guidance and hazard avoidance systems is necessary to enable the spacecraft to effectively choose the optimal landing spot using real-time visual information from the planetary surface.

## 1.3 Research questions

For this master's thesis, the following research questions were formulated, with an aim to explore the field of visual processing, guidance and simulation environments for space missions:

1. How can visual information be used to extract features and gradients from planetary surfaces and therefore aiding in safe landing and hazard avoidance during robotic missions? This question will explore the various methodologies and algorithms that can be utilized to effectively process visual data, identifying key obstacles that can impact the success of a mission. It will also involve a study on of how these extracted features can be integrated into the decision-making process of the landing and navigation systems of robotic spacecraft.

2. How can the Unity environment be utilized to simulate planetary surfaces and analyze the performance of vision-aided landing and hazard avoidance systems? This question will lead to the environment simulation and testing using Unity, a powerful real-time 3D development platform, for creating realistic simulations of different planetary terrains.

3. How the lighting conditions on different planetary surfaces can affect the accuracy the system? This question will investigate the influence of various inclination of the light source on the performance of the system.

The answers to these research questions will contribute to the ongoing development of autonomous planetary landing systems for space exploration, particularly in terms of improving their reliability, adaptability, and overall performance.

## 1.4 Objectives and scope

The objectives of this master's thesis are as follows:

1. Investigate the utilization of visual information obtained by an RGB camera from a planetary surface, specifically focusing on the extraction of gradient information. This investigation will examine the methodologies and algorithms that are effective in processing the visual data from a planetary surface.

2. Execute guidance for the spacecraft to reach the designated safe landing positions. This task involves the computation of a series of waypoints which creates the trajectory from the initial to the final position. This trajectory planning will

incorporate odometry data, characteristics of the landing site, and the proposed timeframe, to ensure a controlled and safe descent towards the identified landing zone.

3. Develop a two-stage visual-aided hazard avoidance system that combines gradient information and LIDAR scans to discern safe landing sites. This involves creating an integrated system that effectively combines information from different sources, enhancing the accuracy and reliability of hazard detection and landing site selection.

4. Test the system under varying lighting conditions using the Unity simulation environment. This testing process will involve the creation of diverse lighting conditions in Unity to evaluate the robustness and adaptability of the system, as well as refine its operation under a wide range of real-world scenarios.

The scope of this project includes the design, development, and implementation of a visual-aided landing and hazard avoidance system for space exploration missions. This system will utilize gradient extraction techniques applied to images captured by an onboard RGB camera and data from LIDAR scans. Together, these will be utilized to identify and avoid potential landing hazards on planetary surfaces. This project will also cover the use of the Unity environment for realistic simulation of various planetary conditions, as well as for performance testing under different light conditions.

## 1.5 Methodology and approach

For solving the autonomous safe landing problem, a comprehensive methodology was developed and implemented. Initially, a virtual environment replicating the terrain of a celestial body was created using the Unity game engine. Subsequently, a simulated spacecraft was custom-designed, equipped with an RGB camera sensor and LIDAR technology for terrain analysis. A connection between ROS2 and Unity was established to enable the execution of computational nodes that were used to identify a safe landing spot and guide the spacecraft to the designated location. Detailed information regarding the process and technical implementation can be seen from the 'Methodology' chapter.

## 1.6 Contribution of the thesis

This thesis offers the following contributions:

1. The design and development of an efficient system for safe planetary landing and hazard detection using visual data. This includes the creation algorithms and methodologies that effectively process and interpret visual information from an RGB camera, as well as LIDAR scans. The system is intended to aid in gradient extraction and obstacle recognition on planetary surfaces, thereby enhancing the safety and success rates of space exploration missions.

2. Evaluation of the system's performance under various lighting conditions within a simulated environment. By leveraging the Unity platform for realistic planetary surface simulations, this project not only tests the robustness of the system under various conditions but also provides opportunities for iterative improvements and adjustments. This ensures the system's readiness and adaptability before its real-world implementation.

3. Establishing an open-source environment that supports other researchers and enthusiasts interested in planetary landing and exploration problems. This contribution will provide a collaborative space where individuals can learn from, contribute to, or extend the present work.

## 1.7 Overview of the thesis structure

This master's thesis is organized into six chapters. The first chapter, the introduction, presents the background by providing a broad overview of the topic, stating the problem, defining the objective, and posing the research questions. In the second chapter, a detailed literature review is conducted, bringing a summary of current knowledge in the field. This review not only compares recent publications works but also identifies the gaps that this research aims to fill. Subsequently, the methodology chapter provides a thorough explanation of the system proposed to solve the presented problem. In the following chapter, results from the implemented methodology are presented. The next two chapters, discussion of the results and the following conclusions are made.

# Chapter 2

# Literature Review

One of the initial steps of the master's thesis involved conducting literature review. This included an in-depth exploration and analysis of contemporary publications within the topic of planetary safe landing to provide an up-to-date understanding of the field.

## 2.1 Overview of relevant literature

In the past two decades, the field of planetary exploration has seen a significant increase in scholarly publications. Initial strategies for landing spacecraft primarily relied on predetermined landing sites, which were selected based on satellite imagery obtained from space missions [6]. However, as the scope of space exploration broadens, this technique will become insufficient. For example, recent space mission such as Japan's Hayabusa2 was still operator-dependent [2], which could result in a potential failure due to slow signal feedback speed and human factors.

Given these challenges, today's research focus has shifted towards the development of autonomous landing and hazard avoidance systems. These systems commonly employ sensors onboard the spacecraft, particularly RGB cameras, to capture visual data of the planetary surface. The images are then processed using computer vision techniques such as feature extraction and object detection to identify potential hazards and safe landing spots [7]. The Mars Curiosity Rover provides a real-world application of vision-based landing systems where it utilized a Mars Descent Imager (MARDI), a fixed-focus color camera, to precisely obtain its position in relation to Mars' surface [4].

Furthermore, the research has expanded to include machine learning methodologies for hazard detection and avoidance. These techniques train artificial intelligence models on large datasets, enabling them to detect various hazards, such as rocks, craters, and other obstacles [8]. However, applying these methods to missions on distant objects could lead to significant challenges due to the scarcity of terrain data imagery.

Overall, researchers have proposed a diverse range of strategies, including vision-aided systems and machine learning models. The future development of these technologies would allow safer and more efficient space exploration missions on distant planets and moons.

## 2.2 Theoretical framework

The theoretical framework of the master's thesis includes integration of multiple fields such as computer vision and guidance systems, in order to successfully achieve safe landing. The use of Unity as a simulation environment and ROS2 for communication further enhances the flexibility and adaptability of the system.

## 2.3 Related work and previous research

Recent advancements in hazard avoidance mostly rely on the application of deep learning models. Among these, Downes, Steiner, and How introduced a model named LunaNet, which utilizes convolutional neural networks (CNNs) to identify craters from images obtained from onboard cameras. Identified craters are subsequently matched with known lunar craters, and these matched results serve as landmarks for localization. Despite the authors claiming that LunaNet can detect twice as many craters compared to other existing solutions, it is should be mentioned that this CNN model necessitates a significant amount of data. However, such extensive datasets are often inaccessible during remote space missions. This limitation induces the need for a more autonomous approach to hazard detection.

One of those relevant literature contributions comes from Villa, McMahon, and Nesnas, who identified rocks and small objects on terrestrial planets' surfaces using a shadow imagery approach. By treating shadows as random measurements of light-occluding objects and employing an algorithm to calculate a probabilistic occupancy grid of surface landmarks, the researchers achieved good results. It is important to note that this technique assumes prior knowledge of the sunlight direction relative to the camera frame. However, in unfamiliar environments with varying lighting conditions, this method may encounter challenges and potential limitations.

Mango, Opromolla, and Schmitt made another contribution by developing a hazard detection system that leverages LIDAR information. Their approach involves a two-phase technique, wherein the initial stage generates a coarse hazard map to identify and exclude the most dangerous regions. Subsequently, in the second stage, a fine map is produced to determine the optimal landing spot. The hazard level is calculated based on measurements of slope, roughness, and surface height. However, it is worth noting that their proposed method has a limitation in terms of the testing stage, as they utilized the RVS3000-3D LIDAR - Pose Estimator for Satellite Servicing. Although this LIDAR system has a working range of approximately 3 km, its weight of nearly 10 kg poses a significant challenge, making it too heavy for integration into a spacecraft mission.

Gao and Zhou introduced a terrain hazard detection method that utilizes statistical information extracted from images, represented by feature density. In their study, various feature extraction algorithms were evaluated, and the Sobel operator was selected as the optimal method for extracting feature points. By analyzing the density of these features in the image terrain, a surface description is generated to identify safe landing locations. However, a key drawback of their approach is the lack of consideration for important factors such as solar elevation, intensity, and other environmental conditions, which may limit its effectiveness in real-world scenarios.

Ciabatti, Daftry, and Capobianco introduced an example of a simulation environment in their study, where they utilized the features of Deep Reinforcement Learning and Transfer Learning to solve the challenge of safe planetary landing in unknown terrains. To achieve this, they constructed a simulation model utilizing the Bullet/PyBullet library. Despite the promising results, the simulation environment contains limitations in its flexibility, particularly when compared to platforms like Unity.

## 2.4 Gaps in the literature and research questions

As mentioned previously, researchers have explored various approaches to address the challenge of ensuring safe landings in planetary missions. Some of these methods involve utilizing RGB images for feature extraction and density calculation, while others employ shadow analysis to identify obstacles. These techniques are suitable for high-altitude operations as they rely solely on camera sensors. However, a significant gap exists in testing these image processing techniques under diverse lighting conditions.

In addition, some researchers have incorporated LIDAR sensors to create 3D digital elevation maps and evaluate hazard levels. However, due to current technological limitations, space missions can only accommodate small and lightweight LIDARs, which often come with a restricted working range. Consequently, a decision has been made to combine both approaches to develop a viable system for real planetary missions.

Moreover, there is a noticeable absence of a proposed simulation environment specifically designed for planetary landing missions to facilitate testing. Therefore, this study aims to address this gap by providing 3D terrain models using the Unity 3D physics engine integrated with ROS 2. This integration will enable other researchers to evaluate and test their methods in a realistic environment.

# Chapter 3

# Methodology

The focus of this study is to implement and evaluate a vision aided autonomous power descent guidance system for planetary soft-landing. The objective was to create a simulation environment involving a spacecraft and terrain, develop a complete framework for safe-landing problem and analyze the performance under different conditions.

## 3.1 Software and Hardware setup

To validate the proposed method, a simulation environment was set up, utilizing Unity's game engine and establishing its connection with the Robot Operating System 2 (ROS2) via the ROS TCP connector, a resource provided by Unity Technologies.

Unity, first released in June 2005, has since become a robust platform for creating interactive experiences, particularly due to its comprehensive variety of features that include physics simulation, 3D rendering, and collision detection mechanisms. It employs C# as its main programming language, allowing the running of scripts and the definition of environment variables. A key advantage of using Unity game engine over alternatives like Gazebo is its better rendering capabilities. Unity allows the re-rendering of images coming from the camera, thus enabling the generation of high-resolution images at various altitudes which is an important factor for this project. Therefore, Unity was chosen as the core of the simulation environment, which guarantees consistent, high-resolution images captured by spacecraft sensors from different altitudes.

ROS2, an evolution of ROS (Robot Operating System), is an advanced tool used widely in robotics applications to facilitate communication between different sensors by creating nodes. ROS2 improves upon its predecessor with enhanced features like increased security, the absence of a ROS master, multi-platform support, and overall better performance. This makes ROS2 a final choice for integrating various sensors and computational nodes in this project.

The ROS TCP Connector [13], a development by Unity Technologies, serves as a crucial link between the Unity environment and ROS2 (Robot Operating System 2). It facilitates a seamless exchange of messages between ROS2 and Unity, thus integrating the Unity game engine more deeply into the wider robotics ecosystem.

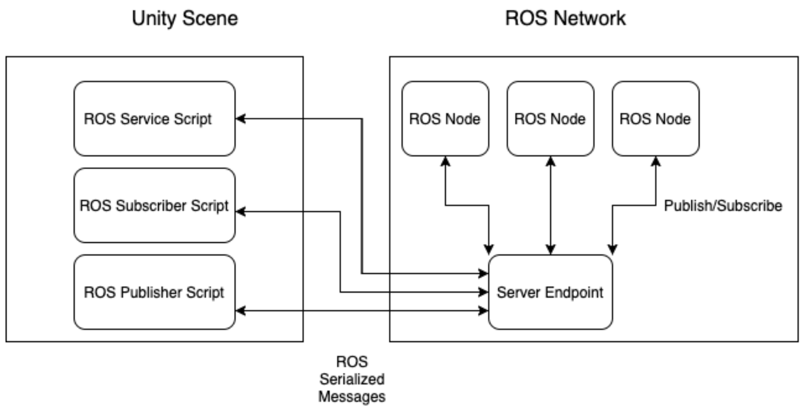



Figure 3.1: ROS–Unity Communication

The connector operates using Transmission Control Protocol (TCP), a standard that controls the transmission of data over network connections. TCP ensures the reliable and orderly delivery of messages across the network, thereby providing a robust communication system which is important factor for accurate and effective simulation. By establishing Unity-ROS integration, the simulation of complex robotic systems and environments within the Unity platform is now available, which can provide photorealistic rendering and physics simulation.

## 3.2 Simulation and Modeling

The simulation scene was created from scratch to emulate the landscape of a distance object. A terrain environment of 4km by 4km was designed, taking inspiration from images of the moon's surface. The simulated surface, including the various craters, hills, and plains, was manually created utilizing Unity's paint terrain tool.

The material for the terrain was selected to mimic the moon's texture and color. Unity's default material was utilized for that purpose, with a grayish color, set to 0 for both smoothness and metallic properties.

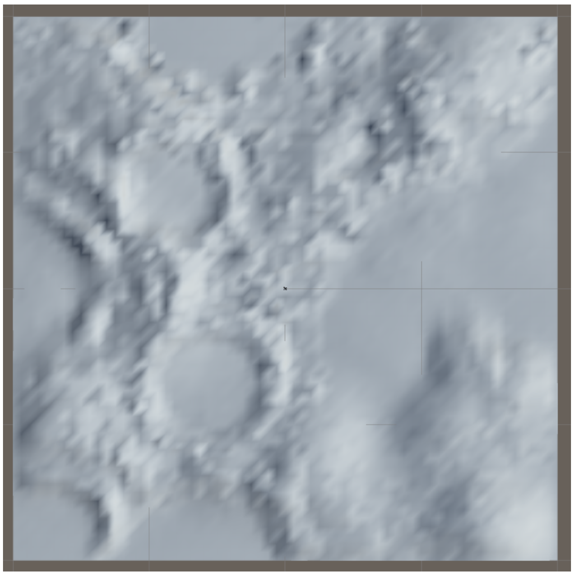


Figure 3.2: Simulated terrain in Unity (top view).

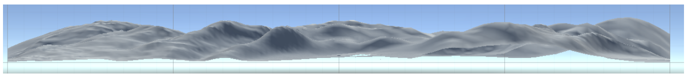


Figure 3.3: Simulated terrain in Unity (side view).

A spacecraft was modeled within Unity using a combination of its built-in 3D objects, including a cube, four capsules, and a sphere. The structure involved placing the four capsules along the vertical edges of the cube to form the spacecraft's body, with the sphere attached beneath the cube representing the sensor's location. To aid in visualization and scale, the cube was configured to be 10 meters, with the capsules being 0.8 meters each. The spacecraft's material was selected to be a default dark-colored material with a metallic value of 1 and smoothness set to 0.7, which offers a distinct contrast with the terrain.

For the visual and guidance systems, an RGB sensor was mounted underneath the spacecraft 0.6 meter below its center, oriented downwards. This sensor is a model of the Logitech-C910 and was obtained from the UnitySensorsROSAssets repository developed by Field Robotics Japan [14]. This sensor offers high-resolution imaging capabilities with its 1920x1080 resolution. In addition to the RGB sensor, a 3D LiDAR (Velodyne VLP-16) was taken from the same repository, providing highly accurate spatial data.

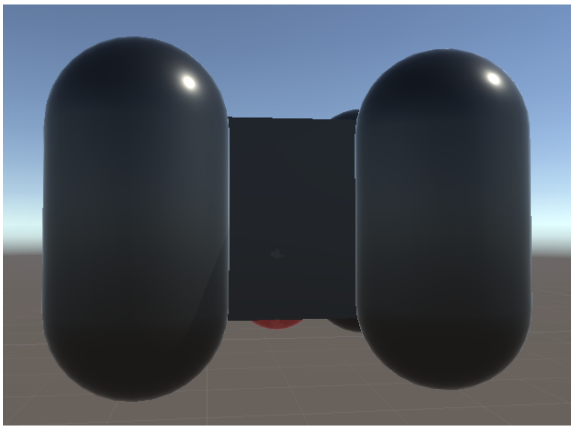


Figure 3.4: Simulated spacecraft in Unity.

The simulation's lighting conditions are controlled by manipulating the position of a directional light source. This light source simulates a large, distant illumination source, similar to the sun or moon. Its location outside the range of the game world ensures consistent and realistic lighting across the entire simulated terrain.

## 3.3 Nodes and Communication

To solve the problem of ensuring a safe landing, a series of specialized nodes were designed to accurately identify suitable landing spots. An illustration of the communication schema among these nodes is presented in the figure 3.5.

**safe_landing_node** - it uses the OpenCV library for processing incoming compressed images, which are presumably obtained from a spacecraft RGB camera sensor. It identifies potential safe landing spots based on gradient analysis of the terrain in the images, and publishes the centroid of the most promising landing region. This is achieved by calculating gradient magnitudes and directions, labeling each connected region of low gradient magnitude, and then computing the areas and centroids of these regions. The largest region's centroid is then published, and the region's outline along with the centroid are visually marked on the image.

**pixel_to_world_node** - this C++ node is designed to convert 2D pixel coordinates (in the image frame) to 3D world coordinates, based on known camera intrinsics and depth information. It subscribes to the "*/centroids*" topic to get the 2D pixel coordinates and to the "*/odom*" topic to obtain the depth information (distance from the spacecraft to the surface). The code employs a pinhole camera model for this conversion process, using predefined intrinsic parameters specific to the Logitech C910 camera. After the conversion, the 3D point in the world frame is published on the "*/centroid_world*" topic.

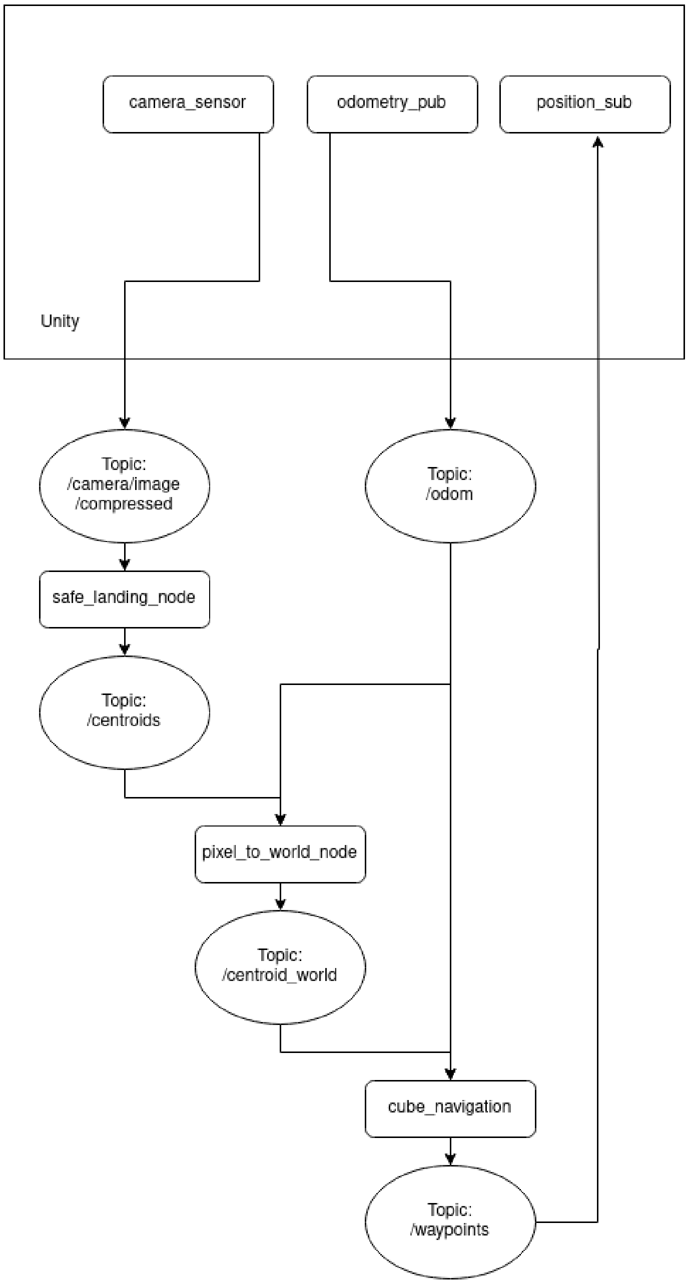



Figure 3.5: Node communication diagram.

**cube_navigation** - is responsible for computing and publishing 3D waypoints given the initial and final positions of a robot. Initial position and velocity are gathered from the "*/odom*" topic, while the final position is obtained from the "*/centroid_world*" topic. Once the positions are obtained, the node calculates a series of waypoints using a polynomial trajectory generation method that ensures smooth motion of the robot by considering initial and final velocities and accelerations. This polynomial interpolation also takes into account gravitational effects. The waypoints are calculated at a frequency of 1 Hz and published on the "*/waypoints*" topic.

The aforementioned nodes perform gradient analysis, feature computation, and visualization tasks by using the extensive capabilities of the OpenCV library. This library is a comprehensive resource, offering a wide range of tools for computer vision and machine learning applications.

As for the nodes inside Unity, there are 3 main components:

**camera_sensor** - This node utilizes a C# sensor script designed for the Logitech-C910 RGB camera to publish images to the "*/camera/image/compressed*" topic. The existing script has not been modified for this purpose.

**odometry_pub** - This node serves the purpose of publishing the position and orientation of the spacecraft to the "*/odom*" topic, which is configured to receive messages of the odometry type. The frame_id for this setup is established as "*world*".

**position_sub** - This node constantly modifies the spacecraft's position based on waypoint values. It's worth noting the differing coordinate systems utilized by ROS and Unity. ROS uses a "FLU" (forward, left, up) frame where the X-axis points forward, the Y-axis points left, and the Z-axis points upward. On the other hand, Unity uses a "RUF" (right, up, forward) coordinate frame where the X-axis points to the right, the Y-axis points upward, and the Z-axis points forward. Due to these differences, necessary transformations to the position values were implemented.

## 3.4 Gradient Analysis, Coordinate Frame Transformation and Polynomial Guidance

In determining the optimal landing spot for a spacecraft, an image processing technique known as gradient analysis is employed. The process begins with the conversion of the received image into grayscale, then the Sobel operator is applied to calculate the image's gradients in both the x and y-directions.

This transformation shows areas of significant change in pixel intensity, potentially representing geographical features such as hills, mountains, craters, or other terrain irregularities. The gradients are converted into polar coordinates, providing the magnitude and direction of each pixel's gradient. Then, a thresholding opera-

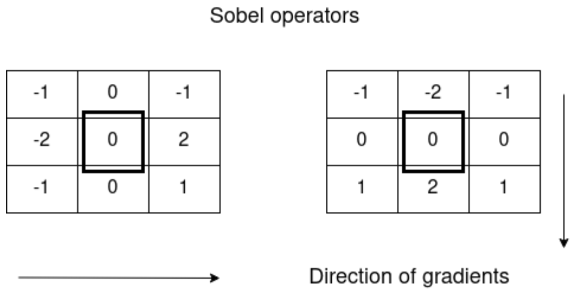



Figure 3.6: Kernels used for sobel gradient method.

tion is executed on the magnitude array, effectively segmenting regions based on the difference of their gradient magnitudes. After experimentation, a threshold value of 5 was found to give the best results.

Following the thresholding operation, the connected components algorithm from the OpenCV library is utilized. This algorithm scans the thresholded image, identifying and labeling connected regions, which are potentially safe or unsafe landing spots based on their gradient characteristics.

Once the regions have been identified, the code computes the area and centroid for each one. The area of a region is computed by the pixel count, and the centroid—representing the geometric center of a region—is computed by summing up the x and y coordinates of its pixels and taking the average. As a result, this process generates potential landing spots based on the terrain's gradient properties. However, it's important to note that these positions are in the image coordinate frame. To be used for navigation, these coordinates must be converted into the world coordinate frame, allowing for the accurate guidance of the spacecraft to the determined safe landing position.

To translate pixel coordinates into world frame coordinates, the camera's intrinsic parameters are required. However, the developers of the Logitech C910 camera simulation in Unity didn't provide these specific parameters. Therefore, the following assumptions were made: the horizontal field of view is set at 70 degrees, while the vertical field of view is set at 43 degrees. Given that the camera offers HD quality, intrinsic parameters are obtained as follows:

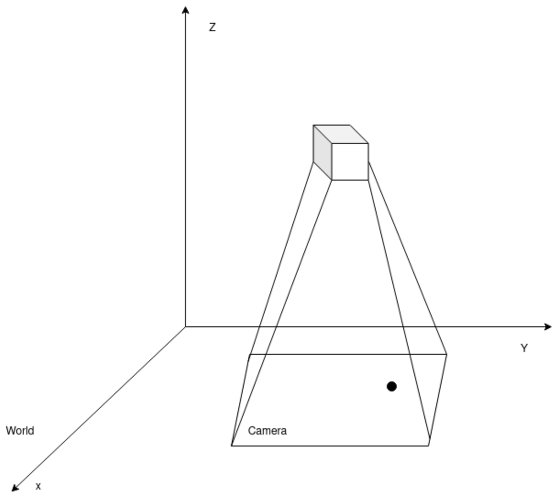



Figure 3.7: Pixel to world coordinate frame.

$$f_x = \frac{w}{2 * tan(HorizontalFOV/2)} \tag{3.1}$$

$$f_y = \frac{h}{2 * tan(VerticalFOV/2)} \tag{3.2}$$

$$c_x = \frac{w}{2} \tag{3.3}$$

$$c_y = \frac{h}{2} \tag{3.4}$$

where $f_x$ - focal length in the x-direction, $f_y$ - focal length in the y-direction, $c_x$ - optical center x-coordinate, $c_y$ - optical center y-coordinate.

With this information, it is possible convert from pixel coordinate frame $(u, v)$ to world coordinate frame $(x, y, z)$.

$$x = (u - c_x) * \frac{D}{f_x} \tag{3.5}$$

$$y = (v - c_y) * \frac{D}{f_y} \tag{3.6}$$

$$z = 300 \tag{3.7}$$

The z-value is configured to 300, marking the altitude at which the spacecraft transitions to utilizing LIDAR data. The camera's depth (D) is calculated by subtracting 0.6 from the current altitude. This adjustment is necessary as the camera sensor is positioned 0.6 meters lower on the z-axis from the spacecraft's center.

Once a secure landing site is identified, it will serve as the final destination for the spacecraft. This process is performed using a polynomial guidance algorithm. The starting position and velocity are obtained by subscribing to the "/odom" topic. The gravitational constant has been set to match that of Mars, registering at $3.71 m/s^2$ and it acts only in the negative $z$ direction. The final acceleration is designed to be equal to this gravitational constant.

System dynamic are represented as:

$$\dot{x} = v \tag{3.8}$$
$$\dot{v} = a - g \tag{3.9}$$

Boundary conditions:

$$r(0) = r_0 \tag{3.10}$$
$$v(0) = v_0 \tag{3.11}$$

Position, velocity and acceleration constraints:

$$r(t_f) = r_f \tag{3.12}$$
$$v(t_f) = v_f \tag{3.13}$$
$$a(t_f) = a_f \tag{3.14}$$

Knowing that:

$$a = a_0 + a_1 t + a_2 t \tag{3.15}$$
$$v = v_0 + (a_0 - g)t + \frac{t^2}{2} a_1 + \frac{t^3}{3} a_2 \tag{3.16}$$
$$r = r_0 + v_0 t + \frac{t^2}{2}(a_0 - g) + \frac{t^3}{6} a_1 + \frac{t^4}{12} a_2 \tag{3.17}$$

By combing these equations, it is possible to represent acceleration as a function of initial position and velocity, final position, velocity, and acceleration, final time and gravitational constant:

$$a_0 = \frac{12}{t_f^2}(r_f - r_0 - v_0 t + \frac{t_f^2}{2}g) - \frac{6}{t_f}(v_f - v_0 + gt_f) + a_f \tag{3.18}$$

$$a_1 = -\frac{48}{t_f^3}(r_f - r_0 - v_0 t + \frac{t_f^2}{2}g) + \frac{30}{t_f^2}(v_f - v_0 + gt_f) - \frac{6}{t_f}a_f \tag{3.19}$$

$$a_2 = \frac{36}{t_f^4}(r_f - r_0 - v_0 t + \frac{t_f^2}{2}g) - \frac{24}{t_f^3}(v_f - v_0 + gt_f) + \frac{6}{t_f^2}a_f \tag{3.20}$$

It should be mentioned that since the movement is in 3D each acceleration value will have 3 components in x,y, and z directions.

# Chapter 4

# Results

This chapter outlines the results obtained from the conducted study. Initial testing focused on the gradient analysis algorithm, which was evaluated by altering the spacecraft's location within the simulation environment. The spacecraft's elevation was set at a maximum of 2000 meters and a minimum of 300 meters (along the z-coordinate), while its positioning along the x and y coordinates varied within a range of -1000 to 1000 meters.

## 4.1 Presentation of empirical findings

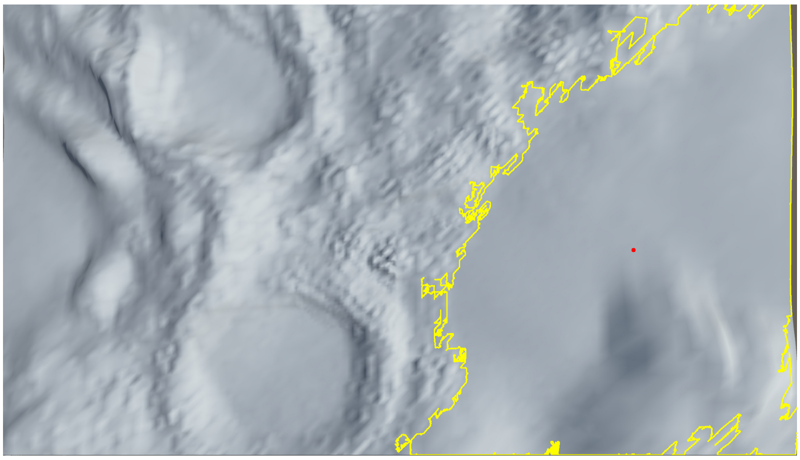


Figure 4.1: Centroid (red point) calculation for spacecraft at x= 0m, y= 0m, z= 2000m.

As it can be seen in the figure, the algorithm successfully identified the largest region with a uniform gradient value, indicating the flattest surface lacking of hazard objects. However, it's worth noting a minor discrepancy where a small hill located at the bottom right of the image was inaccurately classified as a flat surface. This misinterpretation could be attributed to the minimal change in pixel intensity representing the hill, causing the algorithm to falsely perceive it as a safe region. A potential solution to this issue could involve incorporating LIDAR scan data, which is capable of discerning more detailed geometric changes across the terrain.

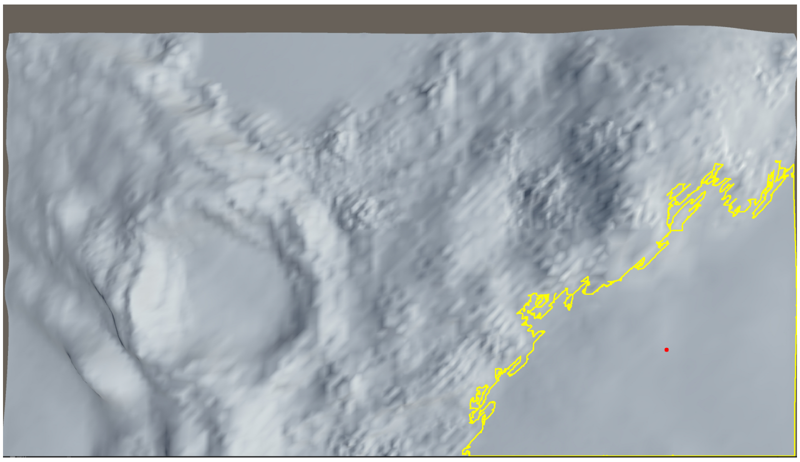


Figure 4.2: Centroid (red point) calculation for spacecraft at x = 0m, y= 1000m, z= 2000m.

As the surface texture grows increasingly complex with an uneven distribution of rocks and mountains, the algorithm demonstrates robust performance in accurately identifying a safe landing spot.

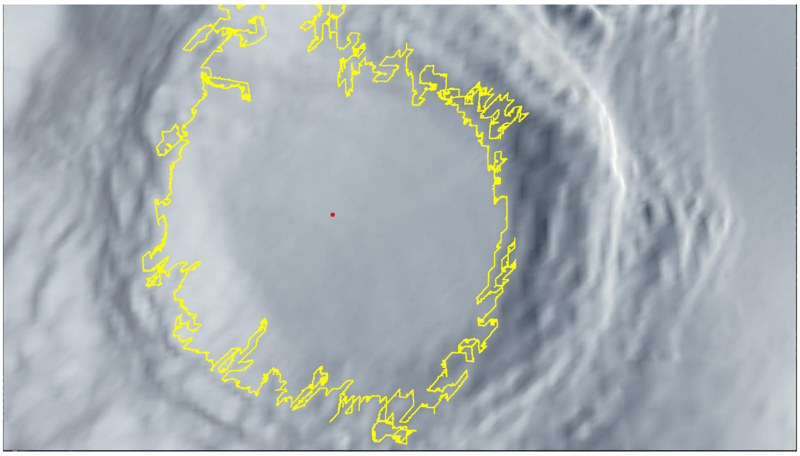


Figure 4.3: Centroid (red point) calculation for spacecraft at x= -500m , y= -700m, z= 1000m.

As the spacecraft approached the crater, the algorithm correctly identified the flat surface within the crater and calculated its centroid location, demonstrating its ability to detect suitable landing zones even within challenging topographical features. The sharp edges around the highlighted region correspond to the high contrast in intensity values attributed to the rocky perimeter of the crater.

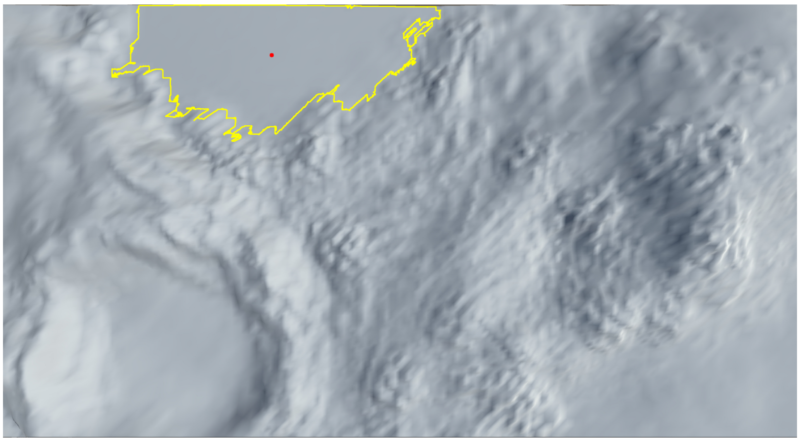


Figure 4.4: Centroid (red point) calculation for spacecraft at x= 0m , y= 1200m, z= 1500m.

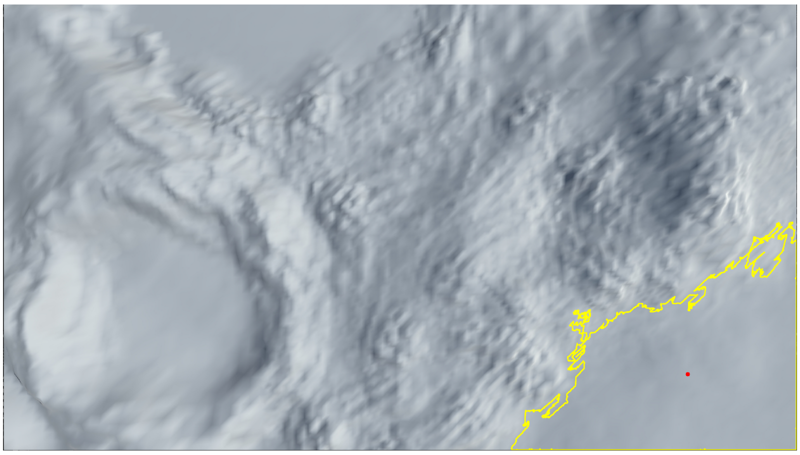


Figure 4.5: Centroid (red point) calculation for spacecraft at x= 0m , y= 1000m, z= 1500m.

The above images illustrate that the centroid's location is not fixed but dynamic in nature. As the spacecraft navigates its path, it constantly captures images of the surface and executes gradient analyses on them. Consequently, should it encounter a region that encompasses a larger area than the previous one, the centroid's position is updated to reflect this change. This behaviour is clearly seen in Figure 4.4 and Figure 4.5, wherein the spacecraft is moving along its Y-axis direction from 1200m to 1000m. During this movement, the spacecraft recalculates the gradients, subsequently identifying new regions of superior area coverage.

## 4.2 Evaluation of the influence of various lighting conditions

Another important component of the testing phase is the evaluation of the algorithm's performance under various lighting conditions. To accomplish this, Unity's directional light tool was used. This tool does not have a detectable source position and therefore the variations of sunlight rays were created only through rotation along the x-axis.

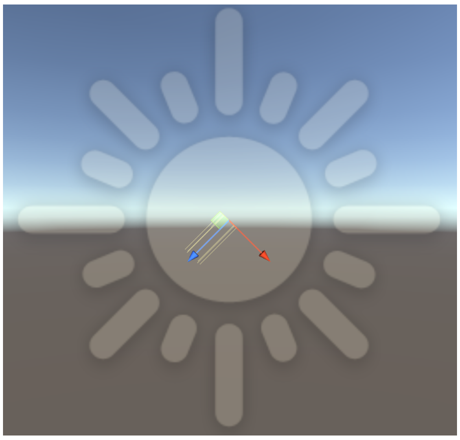


Figure 4.6: Directional light tool with 45 degree rotation along x-axis

To ensure a comprehensive evaluation, it was decided to test the algorithm using 10 different rotation variations, ranging from 0 to 90 degrees. These variations were implemented in increments of 10 degrees. The approach was designed to simulate a variety of lighting conditions, similar to what the spacecraft might encounter in different times of the day or under diverse weather conditions.

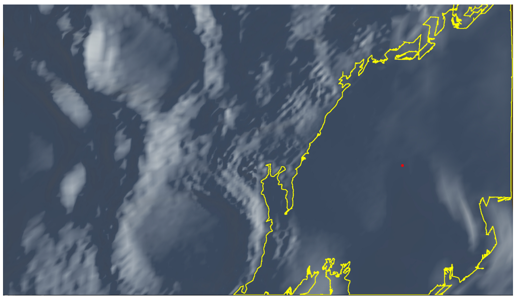
In all experimental trials, the spacecraft remains stationary at a fixed location set at the geometric center of the terrain, specifically at coordinates x=0m, y=0m, z=2000m.

The results indicate that the algorithm operates optimally when the rotation of the sun along the x-axis is minimal. This effect is primarily due to the increased shadow cast by the unevenness of the terrain as the sun nears the horizon. However, as the sun approaches toward its zenith position, the variations in intensity values diminish, thereby thwart computation of safe landing spots.

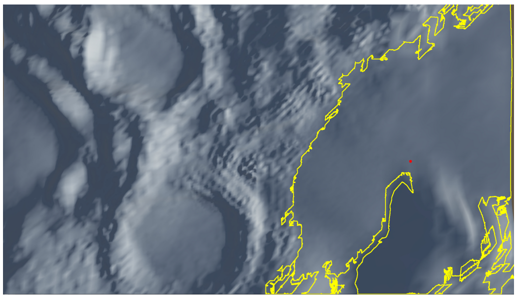
For instance, at angles of 80 or 90 degrees, the algorithm mistakenly identifies the safe landing area. This is likely because the overhead sun minimizes shadow casting, causing a decrease in contrast and texture information that the algorithm relies upon to differentiate between various surface characteristics.

Another observation is that, regardless of sun inclination, the algorithm persistently interprets the small hill region at the bottom right corner as a flat surface. This could be due to the algorithm's tendency to smooth out minor terrain irregularities, which can result in the misclassification of small hills as flat areas.

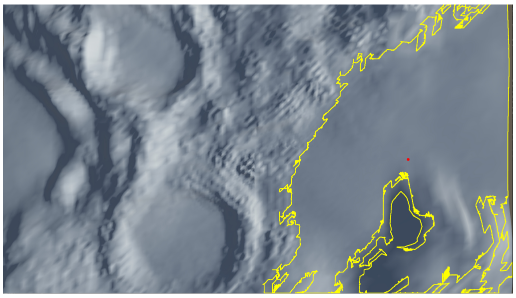
Such misinterpretations suggest the importance of considering multiple sources of information such as LIDAR scans.



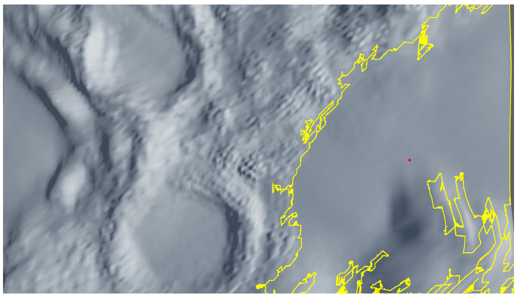
(a) 0 degrees



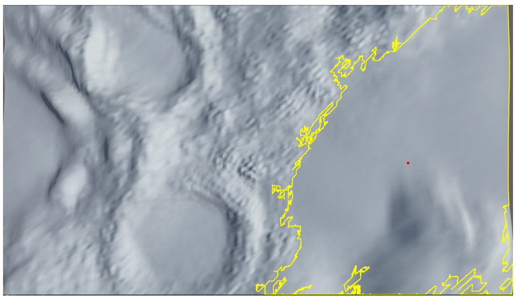
(b) 10 degrees



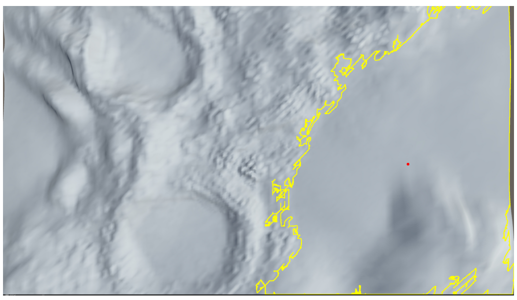
(a) 20 degrees



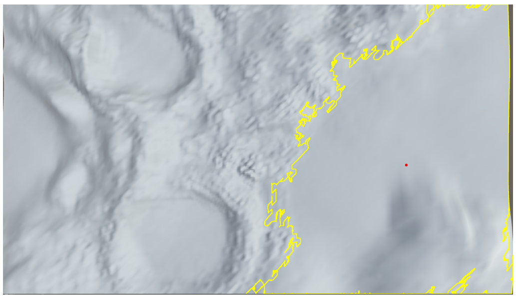
(b) 30 degrees



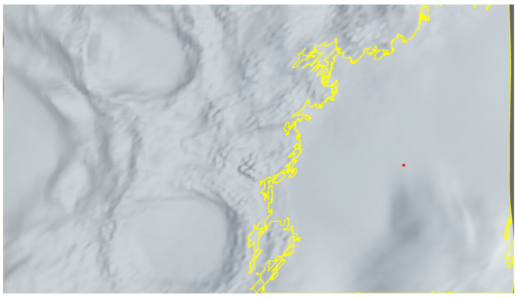
(a) 40 degrees



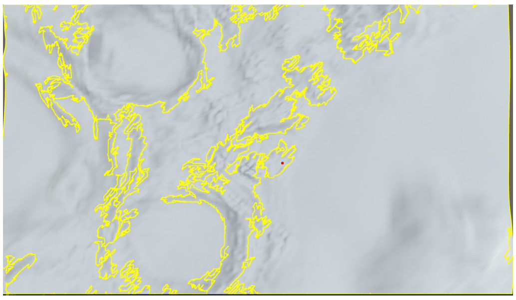
(b) 50 degrees



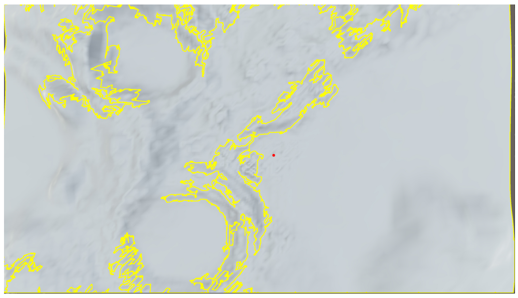
(a) 60 degrees



(b) 70 degrees



(a) 80 degrees



(b) 90 degrees

Figure 4.11: Gradient analysis under different rotations of the light source.

The initial concept involved blending gradient information with feature extraction. However, after employing the ORB feature extractor and detector for keypoint computation, it was found that the number of keypoints was insufficient to accurately determine a safe landing spot.

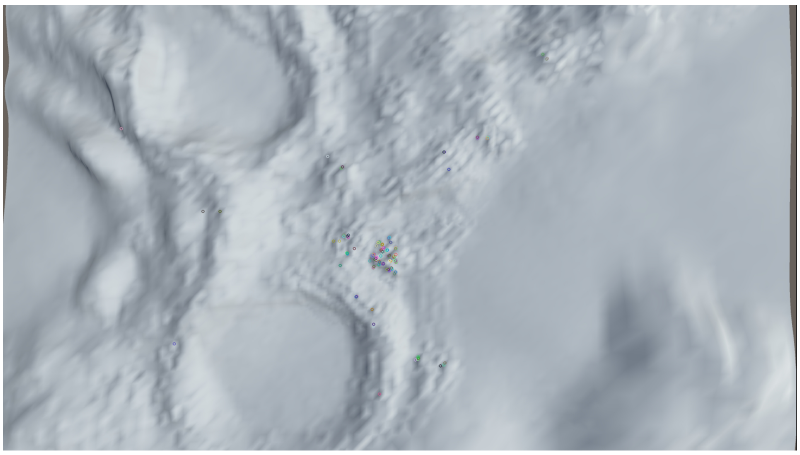


Figure 4.12: Keypoints calculation using ORB feature detector and extractor

## 4.3 Polynomial guidance

After the centroid location was obtained and correctly transformed into world coordinate frame, the spacecraft starts to move the final position in accordance with polynomial guidance.

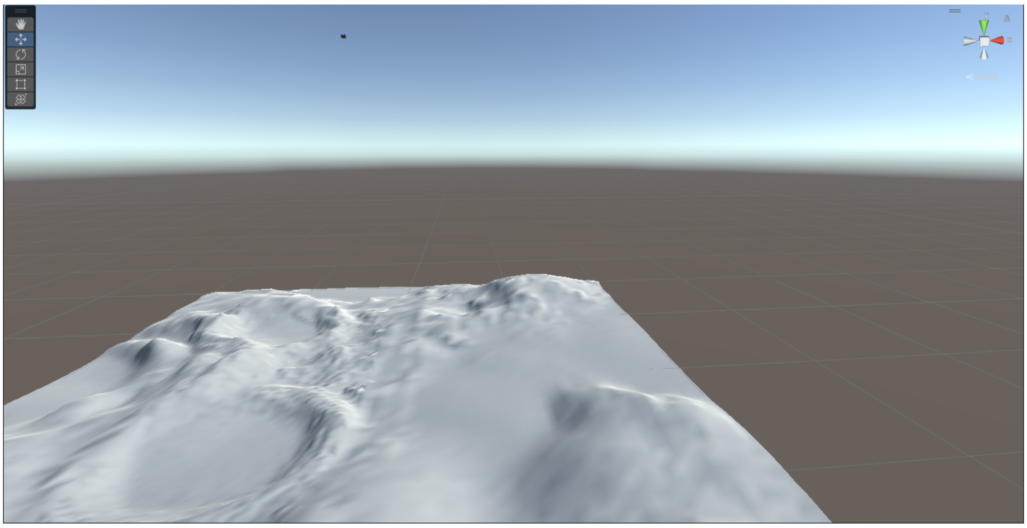


Figure 4.13: Spacecraft apporaching the surface under polynomial guidance

A short demonstration of the movement can be seen by following this link.

In the simulation provided, the allotted time for the final touchdown was equal to 10 minutes. This time frame was chosen to allow for a smooth and controlled descent trajectory. The results clearly indicate that the spacecraft, under the guidance of polynomial algorithms, can successfully and accurately navigate to a safe landing position.

## 4.4 LIDAR test failure

To counteract the disadvantage of the gradient analysis method, where it incorretly identifies hilly terrains as flat surfaces, the adoption of LIDAR sensor measurements was decided upon. LIDAR, an acronym for Light Detection and Ranging, is a versatile instrument that operates by projecting a 360-degree laser beam. It calculates the time taken for the reflected light to return to the receiver, thus generating a 3D point cloud.

One of the primary advantages of LIDAR is its ability to provide precise geometric data about the terrain. For the purposes of this simulation, the Velodyne VLP-16 was employed. The accompanying C# script allows for the adjustment of various LIDAR parameters, such as the number of layers, increments, maximum and minimum vertical angles, and range, among others.

During the testing phase, it was observed that there exists dispersion of the point clouds which didn't accurately reflect the terrain. Even after calibration of LIDAR parameters, this issues still persists. The reasons behind this discrepancy are assumed to be due to mistakes in the provided sensor script.

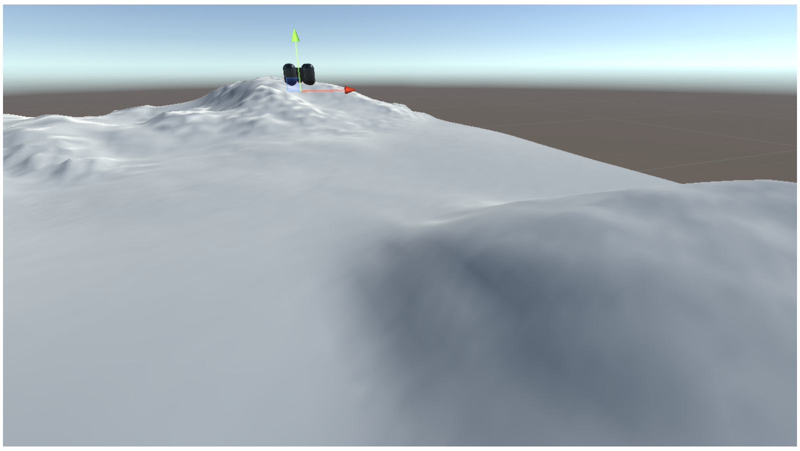


Figure 4.14: Spacecraft approaching a hill.

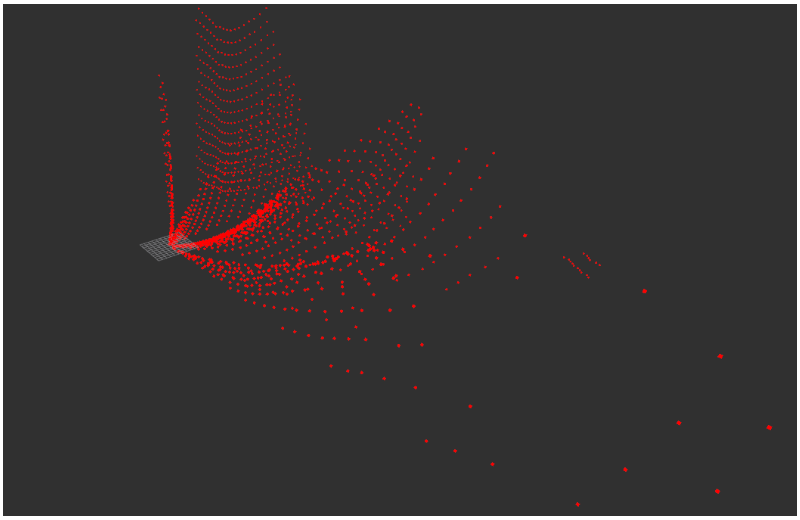


Figure 4.15: Point cloud dispersion coming from LIDAR sensor.

## 4.5 Summary of key findings

The gradient analysis method proves to be highly effective in accurately identifying safe landing regions and determining their centroid position. As the spacecraft moves inside simulated the environment, the results continuously update, adapting to the changing landscape. Nevertheless, there are certain challenging scenarios where hills with minimal variations in height can be incorrectly classified as safe regions.

During the testing phase, it was observed that the gradient calculation is influenced by changes in intensity. However, for the majority of sun inclinations, the resulting safe landing regions remained appropriate, with inaccuracies occurring only at 80 and 90-degree rotation angles.

Initially, the integration of feature points alongside the gradient analysis was considered. However, after conducting several tests, it became evident that the number of features available was insufficient for making reliable decisions regarding safe landing regions.

By implementing polynomial guidance algorithms, the spacecraft achieved successful navigation to the final position, following an optimal path.

Lastly, LIDAR scan measurements were evaluated in the simulated environment. Although the point cloud data was successfully published in Rviz2, it contained sparsity and did not represent the actual terrain.

# Chapter 5

# Conclusion

## 5.1 Summary of research questions and objectives

This master's thesis focuses on the exploration of visual processing, guidance, and simulation environments for space missions. The research questions formulated are as follows:

- How can visual information be used for extracting features and gradients from planetary surfaces to aid in safe landing and hazard avoidance during robotic missions?

- How can the Unity environment be utilized to simulate planetary surfaces and analyze the performance of vision-aided landing and hazard avoidance systems?

- How do lighting conditions on different planetary surfaces affect the system's accuracy?

The objectives of this thesis are:

- Investigate the utilization of visual information, specifically gradient extraction, obtained from an RGB camera on a planetary surface.

- Execute spacecraft guidance to reach designated safe landing positions by planning a trajectory based on odometry data, landing site characteristics, and the proposed timeframe.

- Develop a two-stage visual-aided hazard avoidance system that combines gradient information and LIDAR scans to identify safe landing sites.

- Test the system under varying lighting conditions using the Unity simulation environment.

## 5.2 Conclusions drawn from the research

The gradient analysis method is highly effective in accurately identifying safe landing regions and their centroid position. Results continuously update as the spacecraft moves in the simulated environment, adapting to the changing landscape. However, challenges arise when hills with minimal height variations are incorrectly classified as safe regions. In addition, during testing phase, changes in intensity were found to influence the gradient calculation. Despite this, the resulting safe landing regions remained appropriate for most sun inclinations, with inaccuracies only occurring at 80 and 90-degree rotation angles. Initially, integrating feature points with the gradient analysis was considered, but insufficient features were available to make reliable decisions about safe landing regions. The spacecraft achieved successful navigation to the final position by implementing polynomial guidance algorithms, following an optimal path. In the simulated environment, LIDAR scan measurements were evaluated, and although the point cloud data was successfully published in Rviz2, it exhibited sparsity and did not accurately represent the actual terrain. Although not all objectives of the master's thesis were achieved, this study successfully established a simulated framework for visual-aided planetary safe landing and guidance.

## 5.3 Contributions and implications

The study has made significant contributions in the following areas:

- Implementation of a gradient method for addressing the challenge of safe landing. This method effectively groups pixels with low fluctuations in gradients and calculates the centroid of the region with the highest area, providing a representative landing position.

- Integration of a polynomial guidance algorithm into the system, enabling the spacecraft to navigate to the desired location with precision.

- Development of a Unity-ROS2 framework with simulated terrain and a spacecraft equipped with cameras and sensors. This framework can serve as a valuable tool for other researchers to explore and test their own solutions for the problem of planetary safe landing.

- Experimental analysis of the influence of lighting conditions on gradient calculation. The study revealed that while the gradient approach is affected by lighting conditions, misclassification occurs primarily at high rotational angles of the sun.

## 5.4 Recommendations for further research

Future work could include the following enhancements:

- Implementation of the LIDAR sensor script from scratch and its integration into the existing system. The inclusion of LIDAR will significantly enhance the accuracy of the proposed method, as it provides precise 3D representation of the terrain.

- Expansion of the study's focus beyond guidance to incorporate a control system that takes into consideration the spacecraft's propulsion system. This addition is essential for simulating real-life scenarios and ensuring comprehensive control over the spacecraft's movements.

# Bibliography


[1] Woowon Byun et al. "Performance Assessment of the KASI-Deep Rolling Imaging Fast-optics Telescope Pathfinder". In: *Publications of the Astronomical Society of the Pacific* 134.1038 (2022), p. 084101.

[2] Makoto Yoshikawa et al. "Hayabusa2-the next asteroid sample return mission of Japan". In: *Transactions of The Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan* 12.ists29 (2014), Tk_29–Tk_33.

[3] JN Maki et al. "The Mars 2020 Engineering Cameras and microphone on the perseverance rover: A next-generation imaging system for Mars exploration". In: *Space Science Reviews* 216 (2020), pp. 1–48.

[4] Kenneth A Farley et al. "Mars 2020 mission overview". In: *Space Science Reviews* 216 (2020), pp. 1–41.

[5] Samuel M Howell and Robert T Pappalardo. "NASA's Europa Clipper—a mission to a potentially habitable ocean world". In: *Nature Communications* 11.1 (2020), p. 1311.

[6] Ralph D Lorenz. "Planetary Landings with Terrain Sensing and Hazard Avoidance: A Review". In: *Advances in Space Research* (2022).

[7] Marco B Quadrelli et al. "Guidance, navigation, and control technology assessment for future planetary science missions". In: *Journal of Guidance, Control, and Dynamics* 38.7 (2015), pp. 1165–1186.

[8] Lena Downes, Ted J Steiner, and Jonathan P How. "Deep learning crater detection for lunar terrain relative navigation". In: *AIAA SciTech 2020 Forum*. 2020, p. 1838.

[9] Jacopo Villa, Jay McMahon, and Issa Nesnas. "ROBUST LANDMARK AND HAZARD DETECTION ON SMALL BODY SURFACES USING SHADOW IMAGERY". In: ().

[10] Davide Mango, Roberto Opromolla, and Christoph Schmitt. "Hazard detection and landing site selection for planetary exploration using LIDAR". In: *2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE. 2020, pp. 392–397.

[11] Ai Gao and Shibo Zhou. "Feature Density Based Terrain Hazard Detection for Planetary Landing". In: *IEEE Transactions on Aerospace and Electronic Systems* 54.5 (2018), pp. 2411–2420.

[12] Giulia Ciabatti, Shreyansh Daftry, and Roberto Capobianco. "Autonomous planetary landing via deep reinforcement learning and transfer learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2031–2038.

[13] Unity Technologies. *ROS-TCP-Connector*. `https://github.com/Unity-Technologies/ROS-TCP-Connector`. 2021.

[14] Field Robotics Japan. *UnitySensorsROSAssets*. `https://github.com/Field-Robotics-Japan/UnitySensorsROSAssets`. 2022.