Individual Lab Report - 4

Lunar ROADSTER

Team I

Author: **Simson D'Souza**
Andrew ID: sjdsouza
E-mail: *sjdsouza@andrew.cmu.edu*

Teammate: **Ankit Aggarwal**
ID: ankitagg
E-mail: *ankitagg@andrew.cmu.edu*

Teammate: **Deepam Ameria**
ID: dameria
E-mail: *dameria@andrew.cmu.edu*

Teammate: **Bhaswanth Ayapilla**
ID: bayapill
E-mail: *bayapill@andrew.cmu.edu*

Teammate: **Boxiang (William) Fu**
ID: boxiangf
E-mail: *boxiangf@andrew.cmu.edu*

Supervisor: **Dr. William "Red" Whittaker**
Department: Field Robotics Center
E-mail: *red@cmu.edu*

April 2, 2025

Carnegie Mellon University
Robotics Institute

Contents

1 Individual Progress

1.1 Enhancing the Occupancy Grid Map for Improved Navigation

The previously created occupancy grid map was inaccurate, as it incorrectly classified too many regions as obstacles. To improve its accuracy, I worked alongside Deepam to flatten the Moon Yard surface, as shown in Figure 1, and rescanned the area using a FARO laser scanner. The Moon Yard was modified by creating craters of various diameters and depths. To capture a comprehensive point cloud dataset, we conducted scans from three different locations, ensuring blind spots were covered. These scans were then stitched together using Autodesk ReCap Pro.

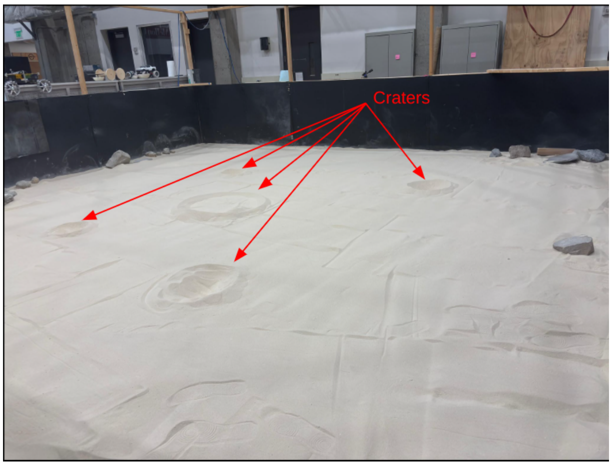

Figure 1: Flattened Moon Yard Terrain with Craters of Varying Sizes and Depths

Next, I developed an algorithm to process the point cloud data. Using the RANSAC (Random Sample Consensus) method, I fitted a plane to the data. Since the Moon Yard surface was flattened beforehand, the detected plane was highly accurate. With this plane as a reference, I defined thresholds to differentiate obstacles:

- **Lower Threshold**: Identified craters deeper than 0.1 meters as obstacles.
- **Upper Threshold**: Used to detect rocks and other objects on the Moon Yard. Since wooden planks were positioned above the ground at a certain height, I set an upper range to exclude them from being classified as obstacles while still detecting relevant objects.

To further refine the occupancy grid map and eliminate unnecessary noise, I fine-tuned RANSAC parameters such as distance threshold and the number of iterations to achieve the best ground plane fit. Additionally, I optimized the lower and upper thresholds to improve obstacle classification accuracy.

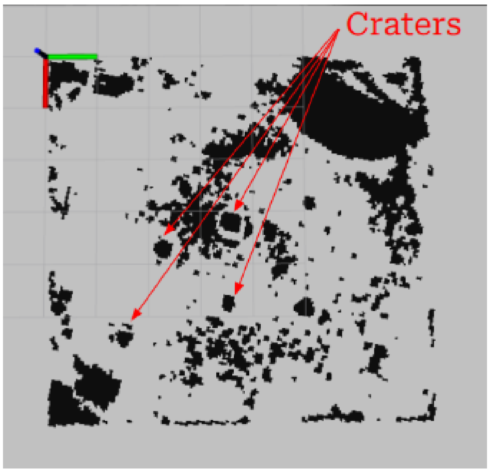

Figure 2: Old Occupancy Grid Map

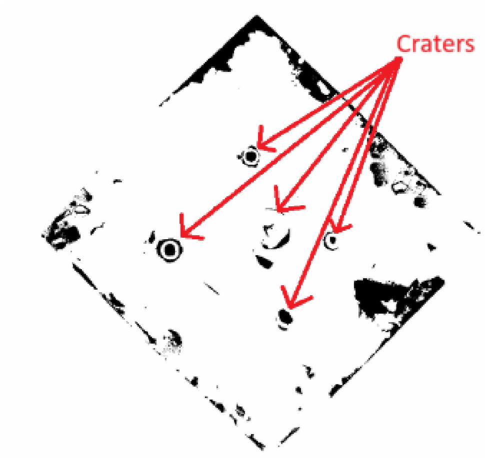

Figure 3: Refined Global Cost Map

Figure 2 and 3 compares the original and refined maps, illustrating the improvements. Since navigation requires .pgm and .yaml files, I converted the refined occupancy grid map into a 2D costmap using the nav2 map server package. The final version of global costmap, suitable for navigation, is shown in Figure 3.

1.2 Detection and Classification of Gradable Craters

The robot must avoid craters with a diameter greater than 0.5 meters and a depth exceeding 0.1 meters, while smaller craters should be groomed. To achieve this, I developed an algorithm to identify regions where crater depths are below 0.1 meters and then determine their diameters based on soil distribution at the rim.

Using OpenCV, I processed the image data to detect black regions representing craters. Given a resolution of 0.01 meters, the 2D costmap was generated, and the coordinates of craters meeting the criteria (diameter $\leq$ 0.5 meters and depth $\leq$ 0.1 meters) were identified, their coordinates were extracted and converted into meters, as shown in Figures 4 and 5.

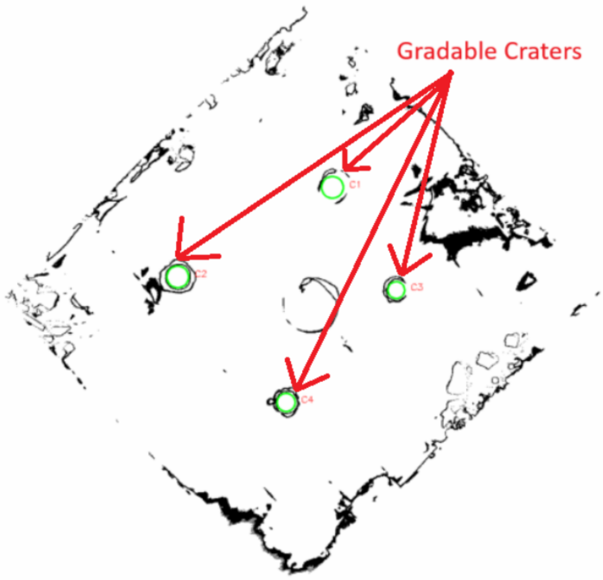

Figure 4: Identified Gradable Craters

```
Centroids of Gradable Craters (World Coordinates):
Crater 1: X = 10.298 m, Y = 3.360 m
Crater 2: X = 7.726 m, Y = 4.857 m
Crater 3: X = 11.359 m, Y = 5.070 m
Crater 4: X = 9.526 m, Y = 6.944 m
```

Figure 5: Gradable Craters Location

To validate accuracy, the extracted coordinates were verified with the actual locations in RViz when the map was loaded. The results confirmed that the coordinates were correct. These verified coordinates will be provided to the navigation stack, serving as waypoints for the robot's movement.

1.3 Implementation and Configuration of the Navigation Stack

For navigation, we are utilizing the ROS2 Nav2 stack. In collaboration with Bhaswanth, I successfully set up Nav2 within an NVIDIA Jetson Docker container. As part of the setup, I modified the launch file to integrate our robot's URDF and configure the costmap. The costmap was then successfully loaded in RViz.

Initially, the URDF only defined the base link, which led to issues with transformations, as frames like odom and others were missing. However, after running the localization stack, the transformations functioned correctly. Currently, I am working on modifying the Nav2 configuration files to enable proper robot spawning.

2 Challenges

One of the key challenges I faced was determining the ground plane before applying upper and lower thresholding. When the terrain had significant variations in elevation, it was difficult to accurately define a plane using RANSAC. To resolve this, we flattened the sand in the Moon Yard to a certain level while maintaining craters of various diameters and depths. This preprocessing step significantly improved the accuracy of ground plane estimation.

Another challenge arose while building the workspace for the Nav2 stack. Some packages had dependencies on the gazebo ROS package, which could not be installed on the NVIDIA Jetson due to its incompatibility with the ARM processor. Since simulation in Gazebo was not required for our project, I resolved this issue by removing gazebo ROS related files from the Nav2 stack. This allowed for a successful workspace build.

Currently, we are encountering transformation issues in the localization stack. Since the navigation stack depends on accurate localization, ensuring the localization stack functions correctly and provides precise robot position data is crucial for further integration and successful navigation.

3 Teamwork

Given my contributions outlined in the Individual Progress section, the following are the contributions of my team members.

1. **Ankit**: His primary focus was on developing the tool planner methodology, incorporating input from the team to ensure seamless integration. He worked with me and Deepam to establish a manufacturing plan for the E-Box. Additionally, he assisted William in debugging wheel odometry and collaborated with Deepam to address the rover's breakdown issue caused by a worn-out rear drive axle.

2. **Deepam**: His main task involved testing actuators with different gear ratios to determine the most suitable option for the project. He collaborated with Bhaswanth to enable teleoperation for the tool, although oscillations at intermediate positions still need to be resolved. Working alongside me, he helped flattening the Moon Yard and forming craters of varying shapes and sizes. Additionally, he collaborated with Ankit to address an issue where the rover was breaking down due to a worn-out rear axle. By sourcing spare parts from a twin rover, they successfully replaced the damaged axle on ROADSTER. He also used Ankit's E-Box design to laser-cut the walls of the E-Box at TechSpark.

3. **Bhaswanth**: He collaborated with William to test the localization stack in the Moon Yard. During testing, they identified that the current issue lies with global localization and are actively working on debugging it. He also assisted Deepam in implementing teleoperation for the dozer. Additionally, he worked with me on the initial setup of the navigation stack on the Nvidia Jetson board.

4. **William**: He collaborated with Bhaswanth to debug the localization stack for the rover. While the local localization (odom to base link transform) was successfully implemented using the IMU and wheel encoders, the global localization (map to base link) still requires further debugging. Additionally, he worked on the sensor stack, integrating the RealSense camera with the docker container and ensuring it correctly publishes point cloud data over a ROS topic.

4 Plans

Until the next lab demo, my focus will be on completing the navigation stack. I plan to spawn the robot at a defined position in RViz and remap the necessary localization topics to align with Nav2 topics. Additionally, I will verify whether the Nav2 generated commands are correctly received by the hardware interface node on the robot. Another key task is configuring the local costmap parameters to integrate point cloud data from the RealSense camera while fine-tuning Nav2 parameters to optimize performance, ensuring accurate robot localization and real-time updates in RViz.

William and Bhaswanth will continue testing the localization stack, refining both global and local positioning to enhance navigation and ensure that transformations between frames are correct. In collaboration with Bhaswanth, I will work on integrating the navigation and localization stacks. Once integrated, we will test the robot's navigation by specifying waypoints in RViz and verifying whether the robot accurately reaches the goal location. Meanwhile, Deepam and Ankit will focus on developing the tool planner, ensuring precise control over the dozer mechanism.