ABSTRACT

The planning of a collision-free trajectory on a planetary surface is one of the issues that becomes more and more important as planetary robot autonomy improves. Operators on Earth used to plan trajectories up until a few years ago. However, this consumes limited bandwidth and communication resources and shortens the rover's useful activity time. In this study, we focus on autonomous guiding and navigation, in particular on the function of path planning and its application to robotic autonomous planetary platforms.

This framework has been divided into the following stages: Image stitching, 3D image Reconstruction, Object Detection, Top-Down View to Map, Finding Shortest path. The Rover Captures the images, the captured images are stitched, which are then converted into 3D point clouds, while the panorama is analyzed for Hazards. Top-Down View of the Map is generated in the form of an occupancy grid and the Shortest path between two points are found using the well-known A* algorithm.

Using all these methodologies our main aim is develop an algorithm that can generate its path autonomously without human intervention. The system should allow the rover to detect hazards in its path and estimate between multiple paths to find the optimum path without human commands.

[**Computer systems organization**]: Architectures—Other Architectures—Neural Networks; Embedded and cyber-physical systems—Robotics—Robotic Autonomy

[**Computing methodologies**]: Computer graphics—Image manipulation

Contents

List of Tables

List of Figures

ABBREVIATIONS

LDA	:	Latent Drichlet Allocation
API	:	Application Programming Interface

NOTATIONS

α	:	Smoothing factor for words
β	:	Smoothing factor for topics

Chapter 1

Introduction

1.1 Introduction

The planning of a collision-free trajectory on a planetary surface is one of the issues that becomes more and more important as planetary robot autonomy improves. Operators on Earth used to plan trajectories up until a few years ago. However, this consumes limited bandwidth and communication resources and shortens the rover's useful activity time. Depending on the position of the planet, the communication between mars and earth can take 5 to 20 minutes [1]. This time delay isn't ideal in an unknown terrain like Mars. This also leads to a wastage of time, severely hindering the efficiency of the rover. Due to these reasons, a completely autonomous rover would be beneficial which could not only analyse the terrain but also but also decide its own path. In this study, we focus on autonomous guiding and navigation, in particular on the function of path planning and its application to robotic autonomous planetary platforms.

1.2 Motivation

The main intention of this research work is to propose a framework that allows the rover to save time by cutting the time wasted during its wait sequence. In current planetary missions, a sequence of commands are sent from the ground station to the rover twice on every Martian day. This is done because real-time communication between the ground station and the rover is infeasible due to large distance between earth and mars. A rover autonomously navigating through an unknown terrain, will not have to depend on commands from ground station, saving time.

1.3 Problem Definition

To develop a framework of algorithms that is capable of generating its path autonomously without human intervention. The system should allow the rover to analyze its surrounding terrain, detect hazards in its path, estimate paths to its destination and compare between multiple paths to find the optimum path without human intervention.

1.4 Objective

The main objectives of the research work is as follows:

1. To develop a framework which enables the rovers to perform autonomously without waiting for commands from the earth.
2. To develop an algorithm that is capable of detecting hazards like craters, dead volcanoes etc to help the rover navigate.
3. To enable the rover to map its surrounding onto a 2D map in the form of an occupancy grid.
4. To develop a robust algorithm that is capable of estimating various paths, and choose an optimal path to the destination.

1.5 Outline

The work will be divided into 5 phases:

1. Image Stitching: In this phase, the images taken from a camera are stitched together to form a equirectangular panorama. The image stitching uses SIFT feature matching to find features and match them between two images.

2. Detect hazards like craters, pits and dead volcanoes based on the image stitched. This recognition will done with a ML algorithm called as YOLO, which is a real-time object detection algorithm. YOLO is a single-stage real time object detection algorithm that scans through the dense sampling of location to detect objects. YOLO is chosen because of its robust and speed of computation without sacrificing performance.

3. The panorama with the object detected is used for the 3D reconstruction phase. Wherein a 3D is scene is reconstructed by extracting the depth features in the form of depth maps

4. Based on the rovers movement, a top down view of the map is generated. This view will generate a map, which will be stored in memory. This is done so that for every revisit, the rover doesn't have to rescan the terrain.

5. The rover will be able to navigate through terrains based on the map generated in the previous phase. If multiple paths exist to reach a point, each path is evaluated based on minimum fuel/energy used to navigate to the destination. After the evaluation, the most optimal path can be identified. The well established A* algorithm will be used for this.

Chapter 2

Literature Review

Several trajectory planning algorithms have been developed to tackle the problem of rover navigation. One of the most renowned algorithms is the Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT), which was utilized by the Mars Exploration Rovers (MER). A new method has been suggested that utilizes shadows to extract and map obstacles in a Viking terrain [2].

NASA scientists integrated an AutoNav system into the MER systems, which was a modified version of the global path planning algorithm known as Field D*. By incorporating Field D* into the rover's capabilities, it became significantly more proficient at navigating around hazards. This enhanced system ensures that the rover is less prone to getting stuck and can successfully overcome complex obstacles [3].

During a study, researchers developed an enhanced version of the A* algorithm by incorporating factors such as rover traversability and environmental parameters (such as surface slope and hardness) as constraint criteria. The suggested A* algorithm was then compared to the original A* method using the MATLAB platform to evaluate its performance [4].

The application of Generative Adversarial Networks (GAN) has enabled the creation of lifelike 3D environments by utilizing remotely sensed images of landscapes obtained from satellites or drones. This involves the synthesis of a plausible RGB satellite image and the generation of a corresponding Height Map in the form of a 3D point cloud. This point cloud serves as an accurate representation of the landscape and is suitable for meshing purposes [5].

MiDAS is a depth estimation model that utilizes monocular input and has been trained on a wide range of datasets. The system proposed in this study demonstrates invariance to variations in depth range and scale. It emphasizes the use of principled multi-objective learning to effectively combine data from various sources. Additionally, the study highlights the significance of pretraining encoders on auxiliary tasks as a crucial aspect of the approach [6].

GndNet is a cutting-edge end-to-end method that achieves real-time estimation of ground plane elevation information and simultaneous segmentation of ground points. This approach utilizes the PointNet and Pillar Feature Encoding network to extract features and predict the ground height for individual cells within a grid. The network is trained using the augmented SemanticKITTI dataset, specifically tailored for this objective [7].

A study was conducted to investigate a fast and optimal algorithm for smooth path planning in point cloud environments, particularly for robots like autonomous vehicles. This approach aims to find obstacle-free paths that are computationally efficient, smooth, and dynamically feasible. It achieves this by analyzing a point cloud representation of the target environment and employing a modified bi-directional and RRT-connect-based path planning algorithm. Additionally, a k-d tree-based obstacle avoidance strategy and a three-step optimization process are incorporated into the algorithm [8].

In a study, researchers proposed a control architecture for autonomous navigation, which improves the safety and traversing capabilities of the rover. The proposed approach builds upon the existing ExoMars navigation method and efficiently implements autonomous features. It primarily relies on the optical Localization Cameras stereobench, a sensor commonly found in rovers, and has the potential to enable cost-effective long-range autonomous navigation in moderately challenging terrains [9].

Chapter 3

Methodology

3.1 Image Stitching

Rover by taking itself as the center, captures a set of images covering 360° view of its surroundings. Due to the nature of the camera, the FOV of the image captured is limited [10]. Hence, these set of images are stitched together to form a seamless panorama. The methodology involved in the image stitching is as follows. The images are first resized into low resolution images. Next, a keypoint features detection algorithm called as SIFT is used to identify features of an object in an image. These features are conspicuous elements found in the image that can matched in other images. Figure 3.1 shows the SIFT keypoint features detected by the algorithm.

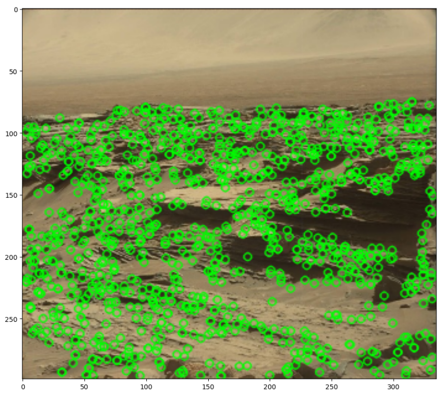

Figure 3.1: SIFT features detected on an input image

This process of feature detection is done on all the images of interest. Next, in a process of feature matching the SIFT features detected on all the images are matched on all combinations of images and a confidence score is derived from the feature matching. These confidence scores are mapped on to a matrix called confidence matrix. In this way the algorithm is robust to mismatched ordering of images. This process is shown in Figure 3.2, wherein the green lines connecting features represents the feature matching.

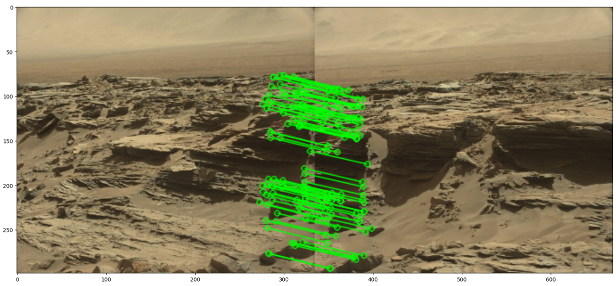

Figure 3.2: Results of feature matching between two images

The images are aligned together based on the feature matching. This process is called as Warping. But because all images exist in a separate coordinate system, they are transformed into a common frame (the common frame chosen is the center image). This is done by the homography matrix. The aligned images further contain black borders where information isn't present. Hence, the largest rectangular image is cropped out of the images. This image is the panorama of interest. Figure 3.3 depicts the warped panorama of 3 images

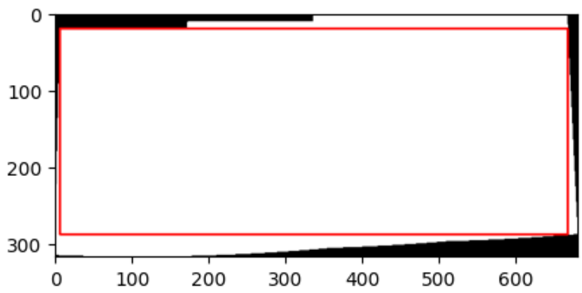

Figure 3.3: The warped images with black backgrounds and the cropped region.

after performing the homographic transformation. The region in red depicts the region of interest of the panorama.

3.2 Hazard Detection

For the hazard detection phase, hazards such as rocks, pits, craters need to be identified. This is important so that these hazards don't obstruct the path of the rover. The panorama generated from phase 1, needs to be detected for these hazards. For this task, the YOLO algorithm was chosen due to its nature of being a single pass algorithm and while also being fast and robust. YOLO can process 155 frames per second and at the same time maintain double the mAP compared to other real time object detectors [11].

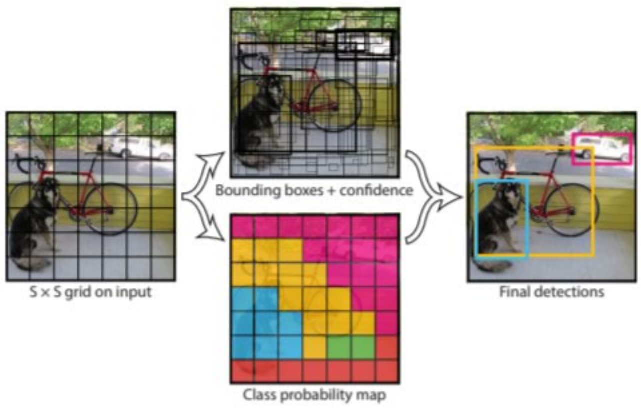

Figure 3.4: General Working of the YOLO algorithm

Figure 3.4 shows the general framework of the YOLO algorithm. The YOLO algorithm uses training images with bounding boxes as input for training the model. The object surrounded by these boxes are labelled based on their classes. These images are fed to the neural network defined by the algorithm. The network divides the image into grid of regions where each region predicts the bounding boxes and probabilities of classes. These bounding boxes are weighted by the predicted probabilities. Figure 3.5 depicts a sample of the training data with bounding boxes defined for rocks.

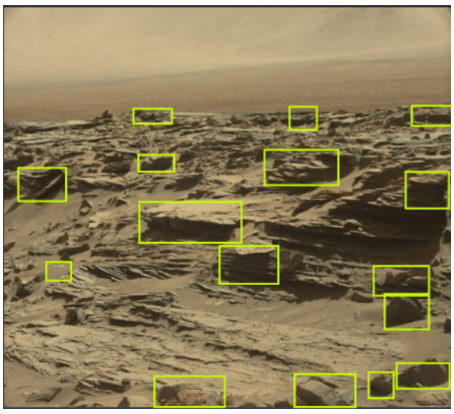

Figure 3.5: A sample training image

The model for our hazard detection phase was trained over 1000 images of marsian and lunar surface images. YOLO makes a predictions with confidence ranging from 1 to 100. Lower the confidence of the prediction, more likely the prediction is going to be a background noice. Hence, an optimal confidence threhsold needs to be chosen, such that legitimate predictions are not filtered out. A confidence threshold of 40% was chosen based on trial and error. Due to unavailability of a suitable data set, a data set containing low resolution images were chosen for the object detection task.

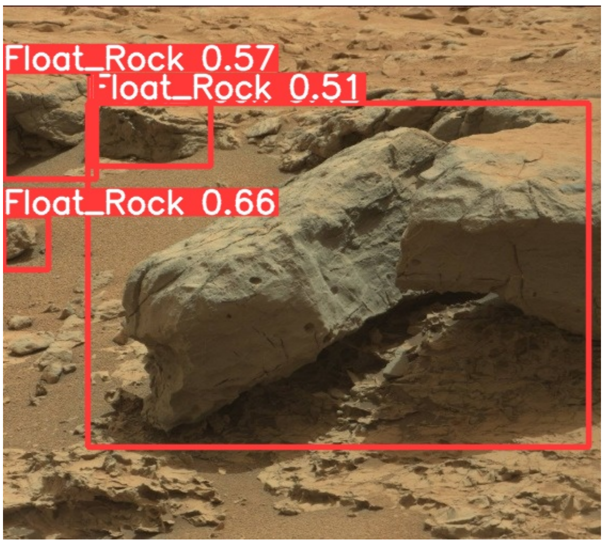

Figure 3.6: Results obtained for sample prediction

3.3 3D Reconstruction

In the 3D reconstruction phase, the panorama image with object detected are reconstructed into 3D scenes in the form of point clouds. When a scene is projected onto an image (when an image is captured), the depth information is lost. This depth information is what distinguishes a 2D image from a 3D scene. Without this depth information, a rover cannot estimate the accurate distance of an hazard from the it. Hence, we regain this information in the form of depth maps. A pre-trained model offered by the transformers library is used for the generation of these maps. Figure 3.7 depicts the depth map generated for the earlier panorama. The intensity of the colors in the depth maps represent the depth of the pixel from the camera.

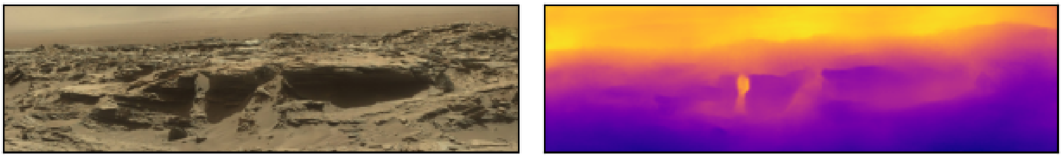

Figure 3.7: The depth map generated for the panorama

With the newly gained depth information, we can use it to reconstruct a 3D scene from the input image. We used the Open3D library to perform a RGBD transformation on the input image to generate a 3D point cloud of the scene. These generated 3D point clouds are vizualized using MeshLab.

3.4 Occupancy Grid Generation

3D point clouds are not suitable medium to be considered as maps. This is because point clouds consume significantly more space than 2D images. To overcome this, the 3D point clouds are projected down onto a 2D plane such that the depth information is retained. This is done in the form of occupancy grids. Occupancy grids are techniques used for map generation in path planning problems. Occupancy grids are grid matrix of the map represented in the form binary values. If a cell in the occupancy grid is 1, then this means that a robot can traverse in that cell. If cell has a value of zero, then the cell is an hazard/non-traversable. While path estimation, the rover will generate the path in the free region (white region) and avoid the obstacle region (black region).

Occupancy grids in this phase are generated by marking the hazards detected during object detection. After a 3D model has been reconstructed, the region within the bounding boxes predicted by YOLO are mapped on to the 3D model with a distinct color. Fig 3.8 represents the hazards mapped on to the 3D model from the captured scene.

Figure 3.8: Projected bouning boxes from 2D images onto the 3D model

Next, the modified 3D model is projected onto a 2D plane in the form of an 2D occupancy grid. While projecting, the 3D pixels within the hazard region are given the value of 0. The rest of the region are considered as traversable region (value of 1).

3.5 Path Generation

Between two points, a path can generated using the A-Star algorithm on an occupancy grid. A-star algorithm is a search algorithm commonly used in path-estimation problems. Given two points (start and destination) a shortest path is generated between them. At every stage, the algorithm will decide its next node, based on the lowest value of a variable 'f', where f is the sum of variable g and h. g is the distance required to travel along the path we created from the starting point to a certain square on the grid and the heuristic, abbreviated h, is an estimate of how far it will take to get from that grid square to the finish line. A suitable start cell is chosen and a destination cell is chosen for the rover. The rover occupies the size of one cell (1m x 1m). The shortest path is mapped onto the occupancy grid in the form of the blue line. Fig 3.9. shows the path generated between two points.

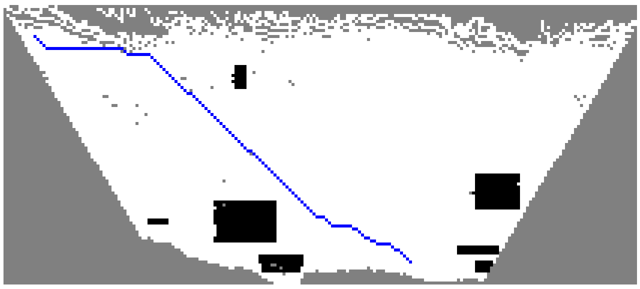

Figure 3.9: The path generated between two points

Chapter 4

Results

The results of each phases are split into subsections. The results for each phases is as follows:

4.1 Image Stitching

As as example, three images were chosen to be stitched together. The results of SIFT feature detection is shown in represented in image 4.1.

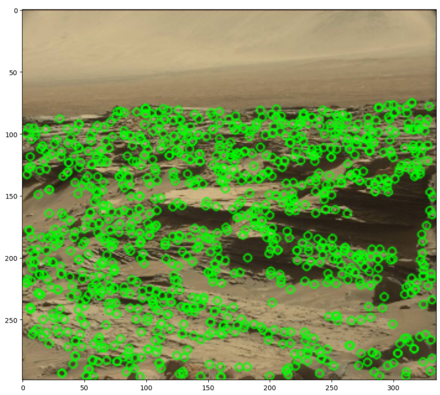

Figure 4.1: SIFT features detected on an input image

The final seamless panorama generated after the image stitching process is shown in image 4.2. The images used for testing the image stitching process was captured by NASA's Curiosity rover.

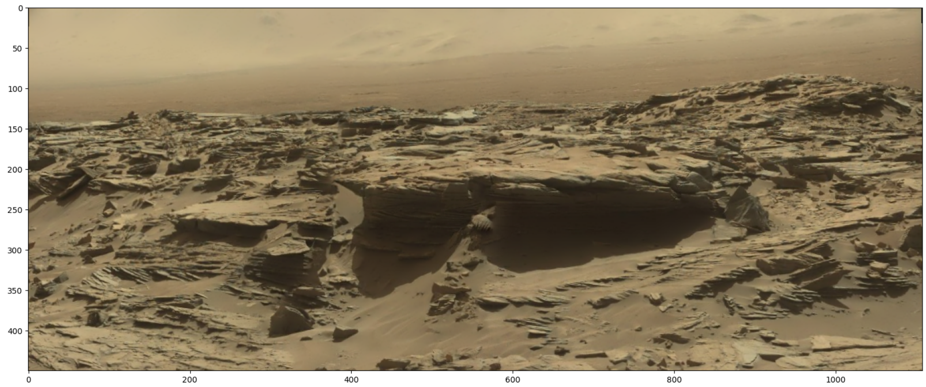

Figure 4.2: The final panorama stitched from the input images

4.2 Hazard Detection

The hazard detection was performed using YOLOv5 on an Nvidia T4 series GPU. The mAP value of the model was 83.9%, with the precision being 80% and 78% recall. Figure 4.3 shows mAP, Precision and Recall graphs.

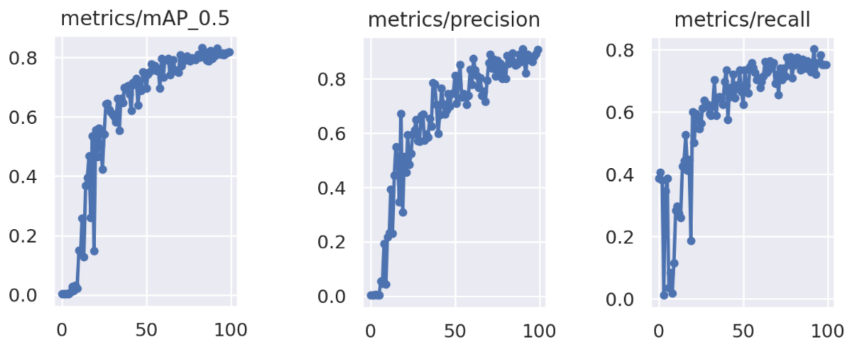

Figure 4.3: The mAP, precision, recall metrics of the model

A sample of the testing data is shown in the following figure 4.4. For the sample testing case, a confidence threshold of 40% was chosen based on trial and error. All the predictions below this threshold have been filtered out, due to their noisy nature.

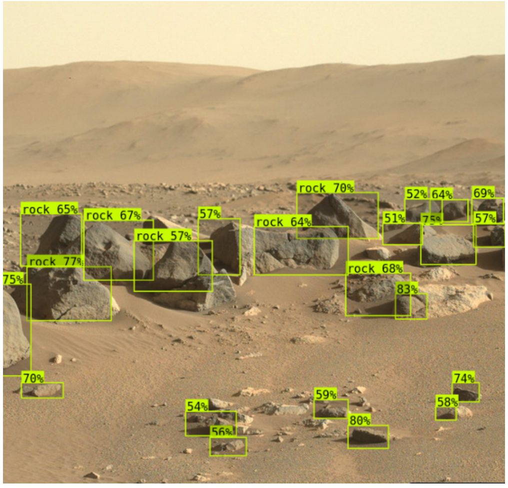

Figure 4.4: A sample of hazard detection by YOLO

4.3 3D Reconstruction

An uncalibrated image captured by the Chinese Yutu rover was used for the 3D reconstruction phase. Figure 4.5 shows the input image and the corresponding screenshot of the 3D model viewed from the top. The resulting point-cloud is visualized in the MeshLab software.

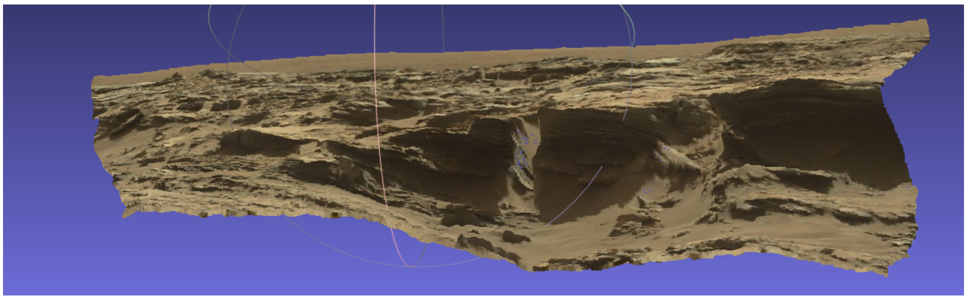

Figure 4.5: 3D point cloud view generated from the 2D image

4.4 Occupancy grid generation

An example of an occupancy grid generated by projecting the 3D point cloud onto a 2D image is shown in figure 4.6. The region in black represent the hazard region. The region in white are the free-traversable region. The grey area represents the uncertain region. Grey areas are high cost region.

Figure 4.6: 3D top down view generated from the 2D image

4.5 Path Planning

Figure 4.7 shows the path generated using the A-star algorithm. A starting point and destination was manually chosen on the occupancy grid (represented by 1 and 4 on figure 4.7). Between the starting and destination region, two region of interest were chosen (represented by 2 and 3 in figure 4.7). These region of interest are to be traversed by the rover before reaching the destination.

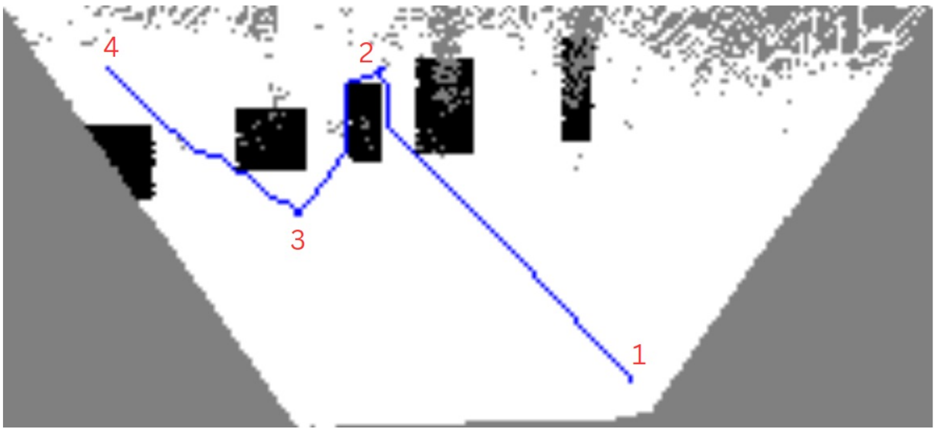

Figure 4.7: Path generated by A-star algorithm

Chapter 5

Conclusion and Future Scope

Path planning is a significant problem in planetary missions. Due to harsh nature and minimum hardware resources available, the rovers on planetary missions need to be efficient in responding to its surroundings. Current methods are susceptible to time delays, gaps in communication and resource wastage. Hence, an autonomous navigation system was devised which is capable of analyzing its surroundings and planning its path accordingly. The proposed system is capable converting images captured by it, into maps through which it can navigate. Various technologies like Image stitching, 3D reconstruction, occupancy grids, YOLO object detection and A-star algorithm path planning was used. The proposed system will allow the rover to navigate autonomously without need for human intervention, saving significant amount of time wasted on waiting for human commands.

References

[1] "Rover ground station communication delay," https://mars.nasa.gov/mars2020/spacecraft/rover/communications/#:~:text=It%20generally%20takes%20about%205,(DSN)%20antennas%20on%20Earth., accessed: 2023-04-30.

[2] D. F. Joseph Carsten, Arturo Rankin and A. Stentz, *Global Path Planning on Board the Mars Exploration Rovers.* Carnegie Institute of Technology, Jan. 2007.

[3] E. C. J. Graciano, *Autonomous Rover Path Planning And Reconfiguration.* Gilching, Germany: AEVO, 2017.

[4] W. Wu, X. Xie, M. Wei, N. Liu, X. Chen, P. Yan, O. Mechali, and L. Xu, *Planetary Rover Path Planning Based on Improved A* Algorithm*, 08 2019, pp. 341–353.

[5] E. Panagiotou and E. Charou, "Procedural 3d terrain generation using generative adversarial networks," 2020.

[6] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2020.

[7] A. Paigwar, Erkent, D. Sierra-Gonzalez, and C. Laugier, "Gndnet: Fast ground plane estimation and point cloud segmentation for autonomous ve-

hicles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2150–2156.

[8] Z. Zheng, T. R. Bewley, F. Kuester, and J. Ma, "Bto-rrt: A rapid, optimal, smooth and point cloud-based path planning algorithm," 2022.

[9] L. Gerdes, M. Azkarate, J. R. Sánchez-Ibáñez, L. Joudrier, and C. J. Perez-del Pulgar, "Efficient autonomous navigation for planetary rovers with limited resources," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1153–1170, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21981

[10] 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*, vol. 216, no. 8, p. 137, Nov 2020. [Online]. Available: https://doi.org/10.1007/s11214-020-00765-9

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.