INDIVIDUAL LAB REPORT #6

AWADHUT THUBE

Team G - The Pit Crew

Team Members: Alex Withers, Justin Morris

Individual Progress

For the first progress review, our team had planned to complete two tasks. One was to add a stereo camera into the Webots simulator and the other one was to develop a heuristic to estimate the distance of the pit edge from the rover. However, my work at the beginning of the semester was not related to the previously mentioned deliverables. Instead, I had to complete the task of obtaining point cloud data from the simulator and publishing it on a ROS topic.

I started off by setting up the simulation environment. The webots simulator allows us to load a custom model as the simulation world. We used the pit model acquired from Utah as our simulation world. The process of setting up the webots simulator and interfacing it with ROS nodes is well documented. Alex also had documented the entire process of setting up the simulator for the purpose of our project.

The next step for me was to familiarize myself with the code that was developed by Alex over the summer. The current working code can be found at https://github.com/awadhutthube/Pit-Navigator-Utah.git. The Github repository consists of multiple ROS packages. The various packages handle various subsystems that operate during the mission. There exists a dedicated package for navigation and control (webots_control), rover path planning (teb-local-planner), mission state machine (smach), etc. The code for adding the camera into the simulation and polling data from it was already developed earlier.

Initially, two camera modules were added to the simulation with the idea to replicate what was achieved during the spring semester. During the SVD, we had generated a dense point cloud from stereo images. However, we realized that following the same approach could fail when working with the simulator. The reason is that the images polled from the individual cameras in the simulator could not be used to generate a point cloud because they lacked features/texture that we would observe in natural images. Thus, we had to think of something else to obtain a point cloud from the simulator.

I learned that two different types of sensors namely the Rangefinder and the Lidar will allow us to obtain a dense point cloud. The process for achieving the same varied in both cases. When using a lidar, we can directly get a 3D point cloud. The lidar scans multiple points in one plane and can return similar results from multiple planes in space. On the other hand, the rangefinder outputs a depth image of the environment. This depth image has to be further processed to obtain a dense point cloud.

The depth image can be processed to output a point cloud using the depth_image_proc package which contains nodelets to obtain different point cloud formats. The only required part of this method was to remap all the topics output by the rangefinder to match those accepted by the point cloud nodelet. This nodelet subscribes to two topics. One that contains the depth image and the other that contains the camera information of the rangefinder.

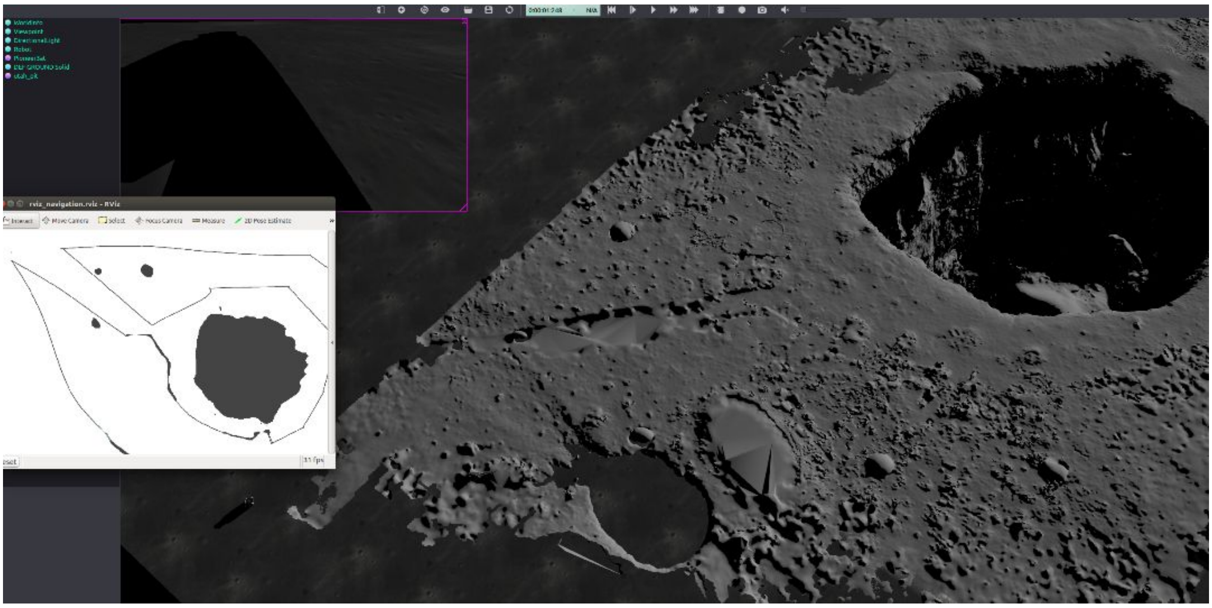

Fig 1: Simulated Lunar Environment

Challenges:

The most challenging part of this phase of the project was to familiarize myself with the simulator and the code developed by Alex. I had to test small modules to understand the operation of the code. Each time I had to test a small module, I had to restart the simulation which took a long time to load. Apart from this, I also had to solve some intermediate compatibility issues of the software and my personal machine.

Team Work:

Alex Withers:

Alex was responsible for identifying the simulator during the last semester. He had the most experience of working with the simulator and had developed most of the planning and navigation code. For this progress review, Alex managed to include a camera module in the simulator. He also documented the entire process of setting up the environment and interfacing it with ROS for the purposes of our project. This made it very easy for me to install and operate the simulator.

Justin Morris:

Justin was responsible for developing the heuristic for measuring the pit edge distance from the rover. He came up with a strategy where we would divide the point cloud obtained from the simulation into smaller parts and fit a plane through each part. The parts which are farthest away from the rover will be used to inform the rover about the pit edge distance. Justin also worked towards setting up the simulation on his own machine so that the entire team can work on the simulation.

Future Plan:

Next up, as a team, we have to focus on more complex terrain patterns and work with different types of slopes. For that purpose, we have to use the point cloud output of the rangefinder data and use the discussed heuristic to obtain information about the environment. Personally, I will be working on the initial processing / transformation of the point cloud. The idea is to use most of the code developed during the previous semester and modify it in a fashion so that it is compatible with the webots simulator.