INDIVIDUAL LAB REPORT #9
10/29/2020

AWADHUT THUBE

Team G - The Pit Crew

Team Members: Alex Withers, Justin Morris

Individual Progress:

For this week's progress review our team had the following 3 goals:

1. Detecting sheer brinks using the triangle mesh structure
2. Generate a triangle mesh from our reconstructed pointcloud
3. Test waypoint navigation on blue

I was responsible for the detection of sheer brinks using the triangle mesh structure in simulation. First, I observed the shape of the generated mesh. On most occasions, the shape of the mesh looked like a trapezium. We discussed among ourselves different ways to use this data to inform our system of the presence of an edge in the vicinity of the rover. Our brainstorming session yielded a couple of ideas out of which, I ended up implementing one.

The approach that I took was to extract points and edges on the far side of the mesh. This was a non-trivial task as the mesh consisted of many points which were stored in a matrix in an unordered fashion. I used Pyvista to extract information about the boundary of the mesh. This included all edges and points that formed the mesh boundary. Once all boundary points were obtained, a simple threshold was used to determine all points lying on the far side of the mesh (points farthest from the base_link of the rover). I observed that the points belonging to the far side of the mesh were roughly half the total number of points that formed the mesh boundary. This extra bit of information helped improve the heuristic. After having all points at the far side, the average distance of the edge was computed from the base link of the rover. Fig 1. Shows the triangle mesh when the rover is close to the brink. The red color of the points highlights that the rover has identified the presence of an edge in front of itself.

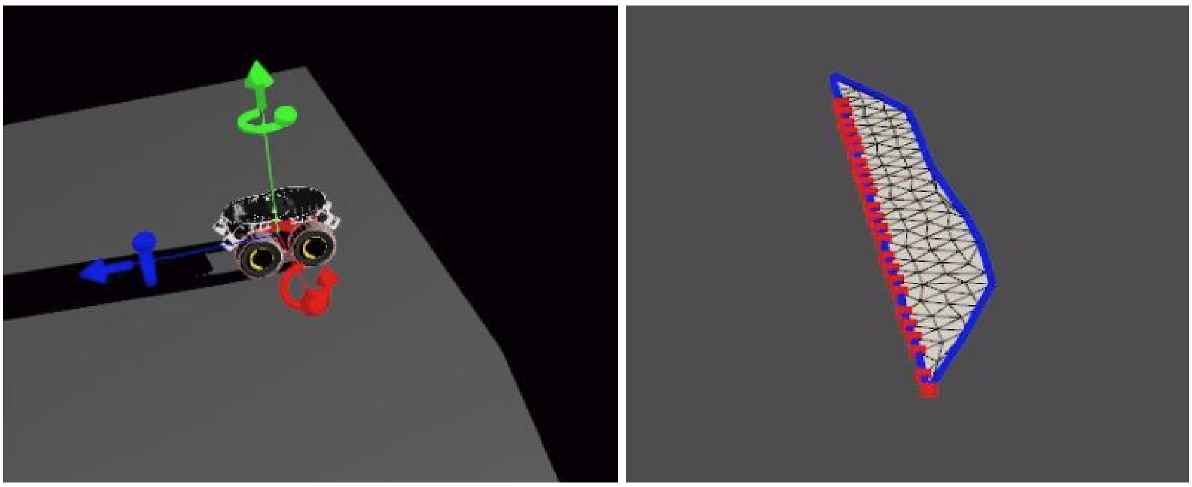

Fig: (Left) Rover close to sheer brink, (Right) Corresponding Mesh

In the process of extracting boundary edges, boundary points and determining the edge distance, I ended up developing a small python library. In addition to the processing functions, I also added some utility functions in this library. These functions will allow us to better visualize our output. Basically, they allow us to color the mesh edges and points with any desired color. Additionally, I have also added code to enable non-blocking visualization of the mesh. This allowed us to capture a video recording while the rover was moving instead of showing only images.

Apart from working on the first goal of detecting sheer brinks, I also supported Justin to complete our second goal which required us to generate a mesh from real data. My work in this area was mainly related to installing the realsense dependencies on the rover computer and setting up cv-bridge in ROS to convert the image messages published by the realsense ROS node into OpenCV images. We then recorded rosbags with raw images from the realsense and converted them into point clouds using OpenCV's block matching algorithm. However, we later decided to use the pyrealsense2 python wrapper to obtain images from the realsense instead of using the realsense ROS package. I had developed some code snippet during the spring semester to use the pyrealsense library to obtain stereo images from the camera. We reused a fair bit of that code for our purposes of unit testing this mesh generation part.

Challenges:

One of the major challenges I faced this time was mainly related to the Pyvista documentation. The Pyvista documentation is pretty sparse and I had to use pythons helper functions like dir() and help() to all attributes and methods related to different Pyvista objects. The Pyvista community is also fairly small as Open3D and MeshLab are the more widely used tools. Another challenge was to figure out the right approach to use the mesh data to estimated edges near the vicinity of the rover. As mentioned earlier, we eventually had a brainstorming session to decide on the correct approach.

Finally, this time I had to collaborate with Justin to achieve our objectives. Collaborating on zoom is difficult as there was a significant amount of lag when we shared our screens with our programs and the simulation running. This made us spend more time on the project than we rather would have. We also had to go to campus to get LiPo batteries as the one we were using was discharged below acceptable thresholds.

Teamwork:

Alex Withers - Alex was working towards our third goal for PR10. He did an amazing job integrating the wheel odometry and the IMU sensor on the real rover to perform waypoint navigation. He faced many problems while trying to accomplish his goals and solved most of them. We were not able to demonstrate perfect navigation during the actual presentation, but Alex figured it out shortly after that. He followed John's advice to use relative orientation from the IMU to obtain better results.

Justin Morris - Justin worked on our second objective as mentioned earlier. He also, started the rover and made sure that all the hardware on the rover was functioning. He supported Alex with waypoint navigation and they both did all of the tests at Justin’s apartment. Justin has also been serving as our project manager and is keeping track of the team’s overall targets.

Future Plan:

We have an upcoming field test on Friday, 30th October, 2020. We plan to go to gascola to survey the site and decide on an appropriate location for performing our FVD. We will also be testing our subsystems as much as possible and will be focusing on collecting some good quality real world data which we could use to further improve our system. Our main focus during this field test would be to thoroughly test waypoint navigation at the site and collect real world data to test mesh generation and brinkmanship later.

Personally, I will be keeping our unit codes ready for use during the field test. Towards the buildup of PR11, my main responsibility will be to integrate brinkmanship with real world navigation. Integrating the complete system in simulation will also be one of our major targets.