

# INDIVIDUAL LAB REPORT #8

10/14/2020

## AWADHUT THUBE

Team G - The Pit Crew  
Team Members: Alex Withers, Justin Morris

## Individual Progress

Our team had 2 main goals for this progress review. One of them was to generate a triangle mesh from the raw point cloud that we previously obtained from the webots simulator. The other goal built on top of the previous one where we had to use the generated mesh to detect the presence of unsafe terrain near the vicinity of the rover. The idea we had in mind was that we wanted to use the normals of each of the triangles in the mesh to decide whether a portion of the terrain was sloped enough to be deemed unsafe.

Before our team could work towards achieving these goals, we had to fix a pending issue that was related to the simulation speed. While running the simulation on our personal computers, we were able to achieve a real-time factor of 0.2. We decided to use one of the lab computers and set up remote access for everybody. I was responsible for configuring the machine so that other people would be able to access it without any supervision. I used the NoMachine software on ubuntu which is developed for fast transfer of multimedia for remote access. I set up our simulation environment and all its dependencies on this new system.

I started off with accessing the raw point cloud and processing it to limit its range. After cropping the point cloud to the required dimensions, I tried to transform it based on the rover's orientation with respect to the global reference frame. This was done by using the orientation data published by the IMU mounted on the rover. Alex added the IMU to the simulation and published its data which I used to compute the appropriate transformation matrix. Transforming the cloud into the global reference frame allows us to distinguish if the rover itself is driving on sloped terrain. In the absence of this transformation, the cloud would be represented in the base\_link reference frame and there would be no information with respect to the absolute value of the terrain slope.

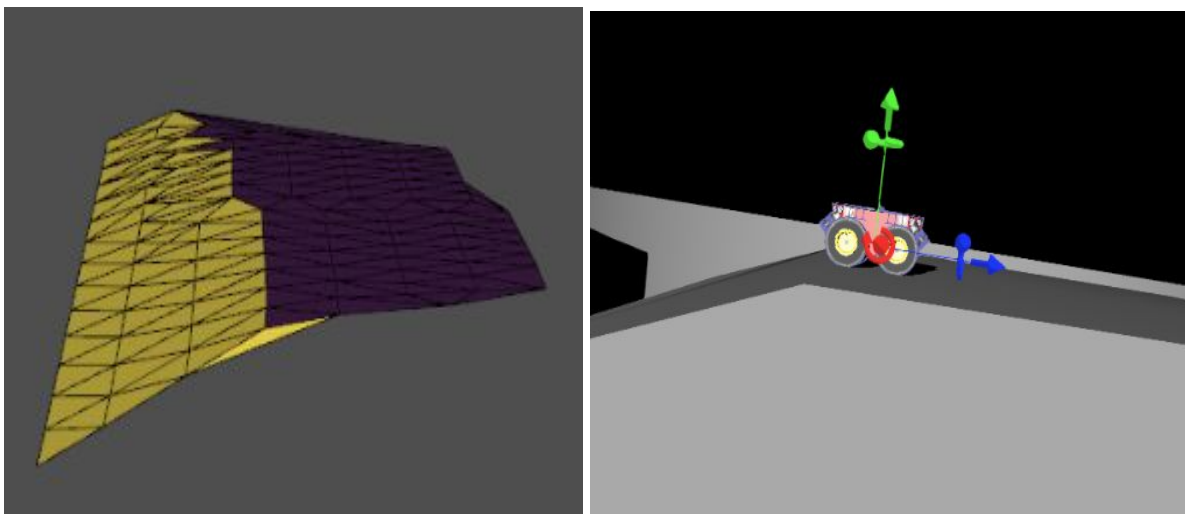


Fig 1: (Left) Generated triangle mesh, (Right) Rover pose in the simulation

Once the cloud was transformed appropriately, I moved forward to discretize it to reduce the mesh computation time. The raw cloud was very dense and in most cases consisted of more than 200,000 points. This cloud was then divided into a fixed number of cells contained in the dominant plane of the cloud. Each cell was represented by the average values of the x, y, and z coordinates of the points contained inside it. Thus, the cloud was converted to a sparse representation that contained these average cell points. This significantly improved the performance of the mesh generation function with regards to the computation time. Later, we ended up using the discretization function provided by the Pyvista library as it allowed us to remove outliers from the point cloud. Pyvista was used as an alternative to Open3D. We initially intended to use Open3D as it has quickly become one of the most used libraries for dealing with 3D data and its visualization. After I obtained this mesh in the required format, Justin took on the work of using its normals to identify unsafe terrain near the vicinity of the rover. The images shown in Fig. 1 demonstrate the generated mesh and the difference in color distinguishes the safe terrain from the unsafe terrain. The cells colored yellow are deemed unsafe for traversing.

## Challenges

This time around, I faced a couple of major challenges as shown below.

1. The first challenge was related to setting up a new computer for remote access. This was the first time I set up a computer for remote desktop access. I found out that my router had to be configured appropriately to accept certain inbound connections. Generally, routers are configured to allow all outbound connections but restrict inbound connections. I spent a lot of time trying to understand the process and figure out good software that would easily allow remote access.
2. The second challenge was finding an alternative library that supported our legacy code and generated a triangle mesh from a point cloud. We had to scrap our initial plan of using Open3D as it only supported Ubuntu 18.04 and we were using Ubuntu 16.04. Migrating to Ubuntu 18.04 would have meant that we would have to migrate from ROS Kinetic to ROS Melodic and we wanted to avoid that at this stage of the project. Hence, I looked up other options for generating the triangle mesh and eventually settled with Pyvista.

## Team Work

Alex Withers - Alex was responsible for setting up an environment that would allow us to test the proposed heuristic of determining unsafe terrain. He created an environment that consisted of different types of sloped terrain. These sloped varied in their angles with the vertical and also in their characteristics. For eg., one slope type was a direct sheer brink. One was a gradual slope. One was built on a curved segment of the terrain. Alex also added an IMU on the simulated rover and published

its angles on an appropriate ROS topic. Alex has also been working towards some of the goals for future progress reviews.

Justin Morris - Justin was working to use the normals of individual cells of the triangle mesh and determining whether a triangle is safe or unsafe for traversal. He also helped me debug one issue in the cloud discretization code. Towards the end, Justin was also working towards obtaining a non-blocking visualization of the generated mesh which would allow us to see changes in the mesh structure as the rover moved in the simulation. Justin is also serving as the project manager and is updating the team with the project requirements short term goals on a timely basis.

## Future Plan

Our goals for the next progress review are as follows:

1. Identify gaps in the triangular mesh which represent sheer drop-offs in the terrain and incorporate these in the planning map.
2. Using the surrogate rover Blue, demonstrate path-following behaviors between multiple waypoints in a physical environment.

These goals build up to our fall validation demo and we would be looking to test most of our simulation code on the real rover.