Team F: FALCON EYE

Individual Lab Report 6

Progress Review 7

Rahul Ramakrishnan

Team
Pulkit Goyal
Yuchi Wang
Pratibha Tripathi
Danendra Singh

February 1, 2018

1 Individual Progress

My primary contribution to the progress was to perform obstacle localization and map them in Rviz along with Pulkit. I was also responsible for understanding the package structure of the pointcloud in order to extract the desired location information using Point Cloud library.

1.1 Obstacle Localization

Now that we were able to detect the obstacles in pointcloud in real-time using Velodyne, we started working on localizing these obstacles with respect to the Lidar frame. We could visualize the pointcloud as shown in Fig.1 without any noticeable lag between the display in Rviz and the changes in the environment as we have updated our mini-pc with higher RAM and SSD drive instead of hard drive. We will now be able to process these pointclouds in the same PC as the husky enabling a modular system.

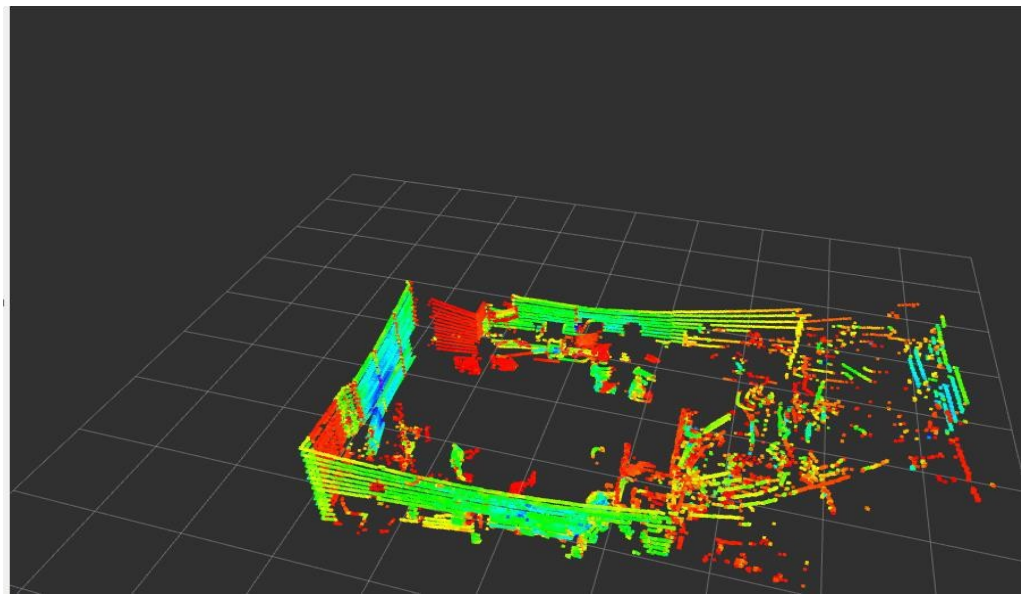

Fig.1 Pointcloud visualization of Tele-supervised autonomous robot’s lab

For the obstacle localization, we obtained the points from Velodyne_heightmap. This package considers every object in the Velodyne’s field of view, above the height specified by us as an obstacle. The detected obstacle's positions were obtained by subscribing to the topic velodyne_obstacles, published by the heightmap. We wrote a subscriber node which subscribes to this topic, extracts the desired coordinates from the pointcloud message structure and displays them on the terminal. It was also visualized in Rviz as in Fig.2.

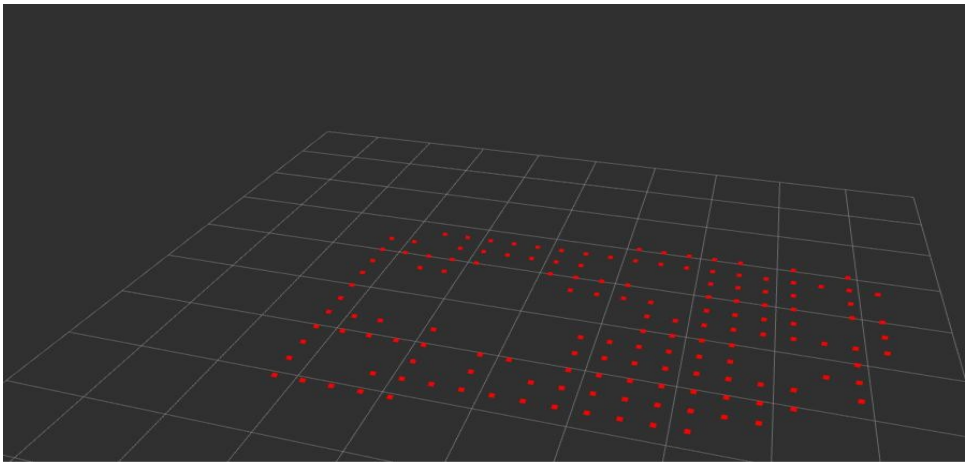

Fig.2 Map of obstacles detected in Fig1

In accordance with the obstacles map, for the navigation purpose it is essential to know the availability of clear path. This is obtained by subscribing to velodyne_clear topic produced by the same height map. Also, the boundaries of the clear path have also been visualized as shown in Fig.3.

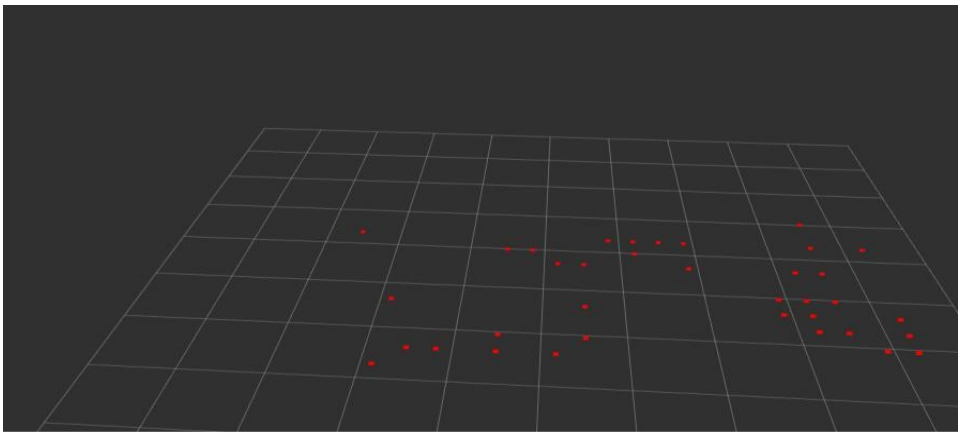

Fig.3 Map of clear path boundaries detected in Fig1

1.2 Point cloud Packet Structure

It is essential to understand the packet structure of pointcloud in order to perform the desired manipulation on them. The point cloud structure has several fields including width (total number of points published), x,y,z positions and several other data members in different format. Our requirements was to obtain the field x and y, which were all float32 type of those points corresponding to obstacles. This will further be essential to our next goal of registering the point cloud corresponding to each obstacle.

2 Challenges

Figuring out a way to extract the coordinates of the obstacles from the published pointcloud was quite a challenge. It was initially unclear of the packet structure of the pointcloud in order for us to write the code for subscriber node. Getting hold of these took a while but hopefully it wouldn’t be a challenge anymore in the near future.

Also, as expected the course works were immense and a lot to handle. But since we had already gone through that for a semester now, it was no longer a major challenge.

3 Teamwork

Yuchi took up the work of connected the localized april tags in form of a graph. This is a vital part of our project. Danendra and Pratibha worked on the calibration of IMU and acquiring that data. hey successfully visualized them in Rviz. Pulkit and I worked on the obstacle localization. Pulkit also coordinated with Danendra for figuring out the intricacies of wi-fi module that we are planning to buy.

4 Future plans

I will work alongside Pulkit on perfecting the localization of obstacles and their registration using PCL library. Pratibha and Danny will continue to work on IMU, getting the data into Husky node. This is a major part of our SVE and hopefully will be completed by next progress review. Yuchi will continue to improve the accuracy of graph localization and also on figuring out a solution to make the drone fly higher than its current limit set by us due to it's inability to detect the april tags.