

# INDIVIDUAL LAB REPORT #3

AWADHUT THUBE

Team G - The Pit Crew

Team Members : Alex Withers, Justin Morris

## Individual Progress

My goal for this progress review was to get our stereo reconstruction pipeline working. This required two parts, building and running the brinkmanship code using point cloud data from the realsense and converting the stereo reconstruction data in a format which can be accepted as a point cloud by the brinkmanship code.

I started with acquiring the brinkmanship code and building it using ros. The code involves 2 parts, the first part being running the realsense camera node which publishes all the camera data on the appropriate ros topics. This part functioned smoothly because I had previously set up and tested the realsense sdk after interfacing it with ros. The second part was the source code for rover brinkmanship. This code was written in C++ and accepted point cloud data from the realsense. The program is written to segment the incoming point cloud into parts which represent points at similar depths. The idea behind doing this is that we expect to find points at two different depths when viewing a pit. The area between the segments would be empty and would represent the pit.

The brinkmanship source code reads preset parameters from the ros parameter server. These include the topic on which to subscribe to obtain the point cloud, the rover dimensions, minimum and maximum depth for cloud filtering purposes, and camera parameters. Running the realsense camera ros node and the brinkmanship source code individually does not work as these parameters are set in a launch file. Running the launch file (realsense\_brinkmanship.launch) starts both the nodes and loads the required parameters on the parameter server. The brinkmanship node takes in the raw realsense point cloud and generates a filtered and transformed cloud. The cloud is transformed taking into consideration the position of the camera on the rover and the rover imu reading. This filtered and transformed cloud is published on a rostopic called /ranger\_brinkmanship/points.

The second part of my work involved getting the reconstructed points from the stereo images and converting them into the cloud format accepted by the brinkmanship code. The reconstruction that was performed earlier dropped the intensity information of the pixels and retained the spatial information (x, y, z position in the 3D space). Before passing this reconstructed cloud for filtering and segmentation, the data format must be appropriate and must have 6 dimensions for each point (x, y, z, r, g, b). I successfully converted the data to a point cloud format using the sensor\_msgs package in ros. However, while publishing the cloud data there is a mismatch in the values being published and the values seen through rostopic echo. The published values are floats (range 0-5) and the values read are integers from (range 100-300).

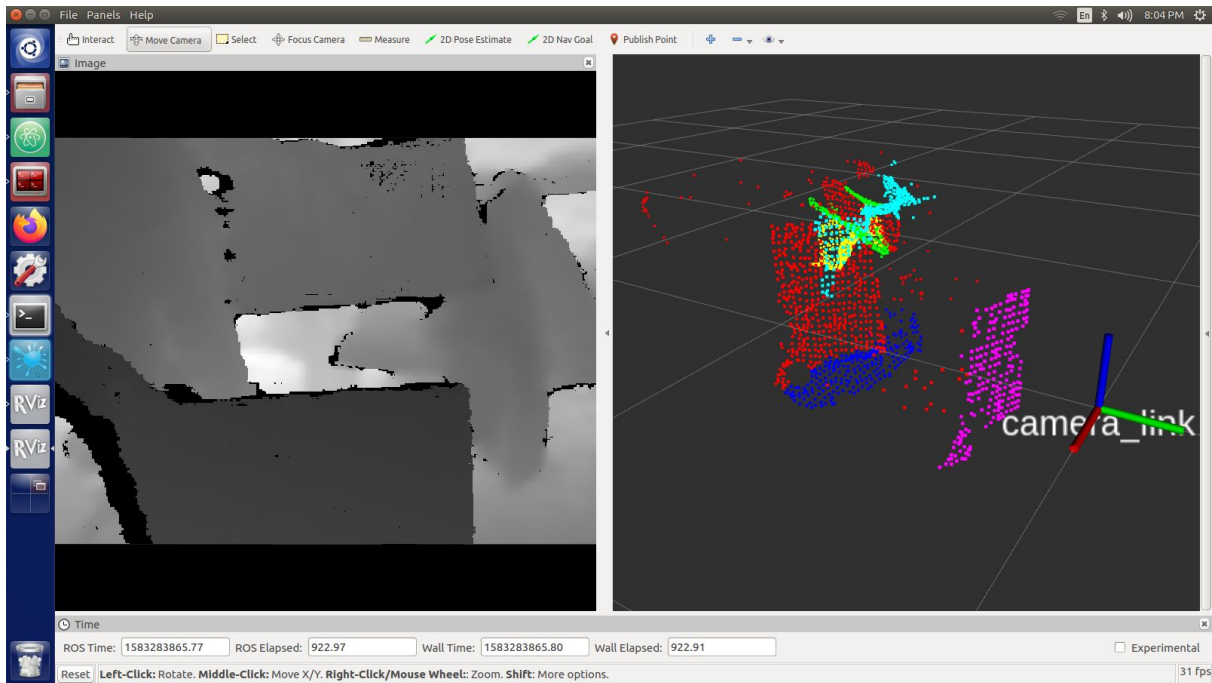


Fig: Depth Segmented Cloud

Apart from the work done on the primary task, I was also responsible for the procurement of items for building a new robot. This would be an exact copy of what is being currently used for field testing.

## Challenges

There were a few challenges which I faced while working on the project during the last two weeks. One challenge I faced in my work was with regards to the data format mismatch between the constructed pointcloud and format accepted by the brinkmanship code. On publishing the reconstructed cloud through a ros node running python, the float values are mapped to integers at a different scale.

Another non-technical challenge was managing all the procurement of items required for building a new rover. One of the items was customizable and hence there was some confusion with regards to the exact order. This delayed the order by a couple of days.

## Team Work:

Alex Withers - Alex worked on getting a realistic pit model to load on the webots simulator. He was able to load a model onto the existing simulation. The model was available readily and he would be looking to replace it with a better pit mesh.

Justin Morris - Justin had brought rover Blue to the MRSD Lab and worked towards getting the rover to run. Justin and I ran the rover using keyboard teleop. We also recorded a rosbag and moved the robot on a large table to visualize the camera data. The space on the lab table was far too less for us to capture any significant data. We could not take the robot out for the test as one of the motors was not working.

### Future Plan:

We have decided to build a robot that could be used for testing purposes. We wish to have full time access to a robot which would allow us to perform quick tests and changes to our system. Accordingly we have started procuring items required to build an exact copy of robot Blue. Some of these items have been delivered to us and we are waiting on the other items. We hope to finish building the robot over the spring break.

We also plan to direct our efforts to getting the complex components of our system running like planning and risk assessment in simulation.