INDIVIDUAL LAB REPORT 9

Progress Review - 10

Shubham Garg

TEAM H (PHOENIX)

Teammates:

Akshit, Parv, Steve

October 24, 2019

Individual Contribution:

1. Opening detection algorithm using point cloud data
2. Debugging, testing and running long missions with the sensor fusion output

Opening detection algorithm using point cloud data:

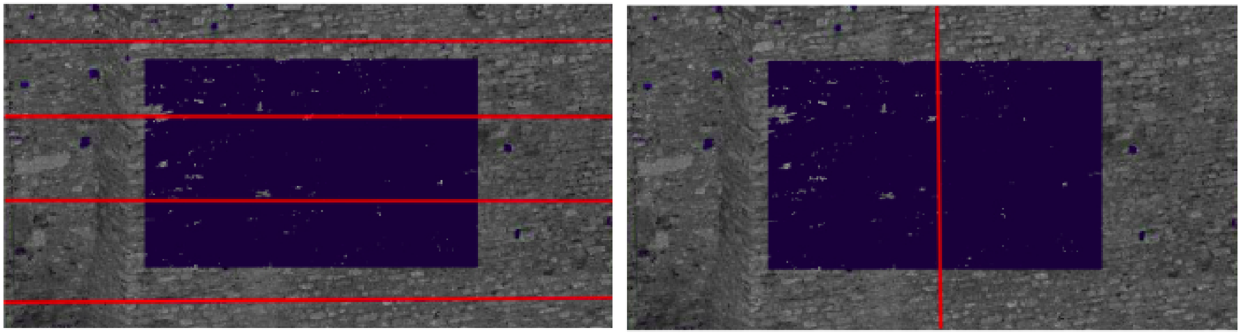
The existing window detection algorithm is based on the depth image which doesn't perform well in an outdoor environment and featureless environment. Moreover, it works for the door like opening where we have a boundary for all four sides. So, our idea is to write a generic opening detection algorithm which can be used for both windows and doors.

Last week I tried multiple algorithms to detect opening which can be used with both platforms. First few days, I tried removing using a convex hull to find the opening. In this method, we need to remove all the dominant plane from the point cloud and just get the plan with the opening. Here, the major difficulty was to remove the dominant plane as it requires a lot of fine-tuning (for RANSAC) and also it might not work in all the environments.

Then we used a different approach inspired by the line scanning method. Now, instead of doing the scan in one direction, we are using a scan in both x & y-direction as shown in Figure 1. First, in the point cloud, we scan along the y-axis (by sliding window mechanism) to find the horizontal edges of the window. Here I have briefly discussed our sliding window algorithm implementation:

1. First, we compute the sum of first k elements out of n terms using a linear loop and store the sum.
2. Then we graze linearly over the array till it reaches the end and simultaneously keeps track of maximum sum.
3. To get the current sum of the block of k elements we just subtract the first element from the previous block and add the last element of the current block.

We slice the point cloud in the y-axis in an interval of 10 cm. Using the sliding window algorithm we find the rising and falling edge along the z-axis in the x-direction. It will give us some start and end world coordinates of the opening. We compute the centroid in the x-direction so that we limit our search in the y-direction. Now, we will slice point cloud in y-direction around the opening center in the x-direction. Now we repeat the above steps of finding rising and falling edge along the z-axis in the y-direction. Using these steps we calculate the center coordinate of the opening.



Scan in y direction

Scan in x direction

Figure 1: Opening detection algorithm

Currently, the z value from the opening detection algorithm is noisy. So, we are getting z after computing the dominant plane in the scene.

Debugging, testing and running long missions with the sensor fusion output:

Last week, I successfully fused the T265 tracking camera, wheel odometry & IMU (3DM-GX5-45 GNSS/INS) but I observed drift in the pose output for longer runs. So, last week I mainly worked on debugging the root cause of the drift and test longer mission on the Husky. Figure 2. shows the trajectory of the husky only with the wheel odometry and with the EKF2 output (T265 tracking camera and wheel odometry). As can be seen clearly in the figure that pose is drifting a lot with the wheel odometry alone because of wheel skidding and unevenness of the surface.

As I mentioned in my earlier ILR that robot localization package gives us the flexibility to select different parameters from $(X, Y, Z, \theta, \phi, \psi, \dot{X}, \dot{Y}, \dot{Z}, \dot{\theta}, \dot{\phi}, \dot{\psi}, \ddot{X}, \ddot{Y}, \ddot{Z})$ for fusion. I tried changing this configuration for IMU but for almost all the configuration the pose output drifting tremendously (like 10m in a minute) even when the robot was stationary. After checking online, I found that we need to fine-tune the covariance values from the IMU. Currently, the covariance value is pretty low i.e. 0.1 and because of it, even the noise is creating a lot of issues. So, for now, we are just fusing the T265 tracking camera and wheel odometry. In our testing, we found that tracking camera and wheel odometry fusion improves the localization and for longer runs. There was 2.5 improvement in the localization for the 20m trip.

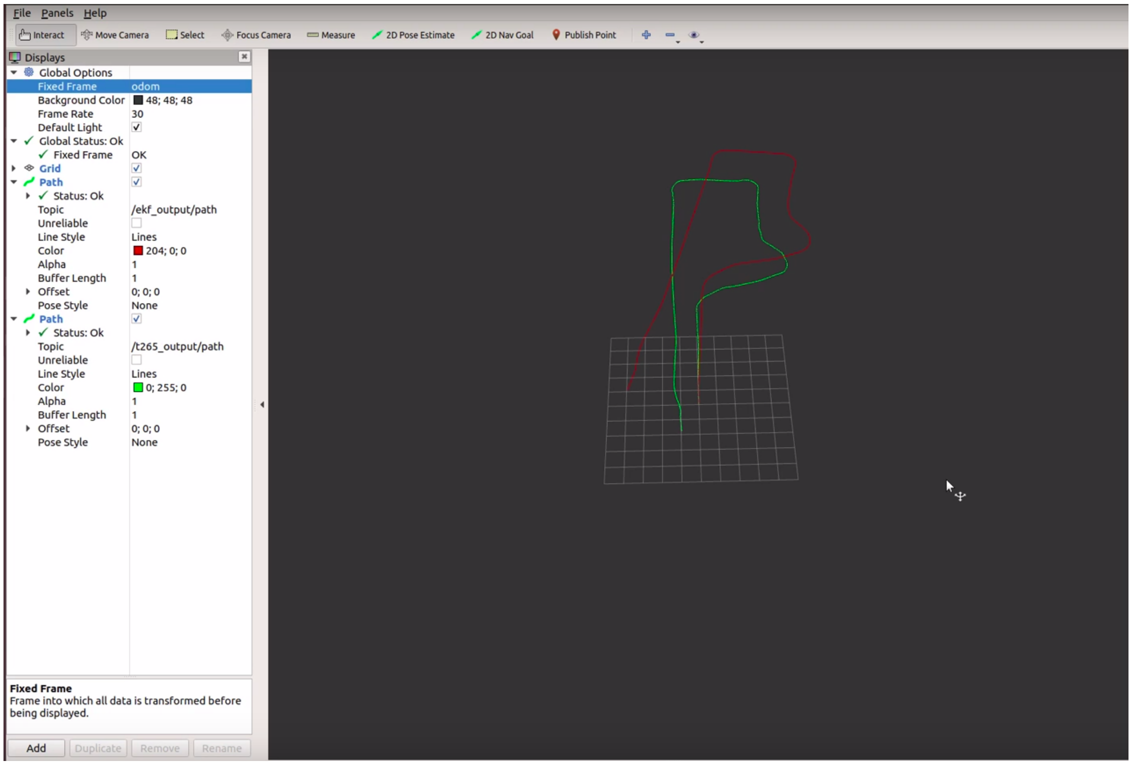


Figure 2: Red line shows the trajectory with wheel odometry, green line show the trajectory with fused output

Challenges:

This week I faced multiple challenges and most challenging so far in the project as I spent almost one continuous week in implementing the opening detection algorithm and debugging sensor fusion issue. Some of the challenges faced in the last two weeks are discussed below:

1. Personally tried a couple of opening detection algorithm. So, I had to go multiple research paper to understand the very time-consuming approaches.

2. Current opening detection algorithm requires a lot of hyperparameters fine-tuning like the sliding window size, size of x, y-axis slice, etc.

3. When we were testing the opening detection algorithm, it was crashing intermittently somewhere and since we wrote code in C++, it was very difficult to debug. Then Parv suggested me to use cLion which has an inbuilt debugger. I am pretty sure it would not have been possible to debug the issue without it as it was a typecasting issue where vector size was getting typecasted to unsigned int and it was supposed to int.

4. Once the algorithm was ready, we faced multiple challenges in testing the code on the DJI as it was not following the position commands. Later, we realized that in offboard mode, DJI turns on the obstacle avoidance and it was considering our window as an ob-

stacle.
5. Testing actual flight testing through the opening was another challenge as the current opening detection algorithm is not robust and sometimes it's not able to detect the window.
6. Testing different approaches for extinguisher and quickly designing and creating lightweight mounts for the UAV.

Teamwork

I felt this was the more collaborative team effort till now where we all helped each other in debugging various issues and brainstorming different approaches and algorithms.
Shubham worked on writing the code for opening detection and **Steve** helped him in designing the algorithm. **Akshit** & **Parv** helped him in debugging various parts of the code.
Akshit & **Parv** were mainly responsible for testing the position controller thoroughly and **Akshit** wrote a script to test some pre-computed trajectories on the UAV. Since the opening detector finds the coordinates in the camera frame. **Akshit** wrote the script to transform it in the UAV frame, he also integrated the opening detection code, yaw computation code.
Parv & **Shubham** worked on fine-tuning, debugging & improving the opening detection algorithm.
Steve worked on brainstorming the extinguisher mechanism for UAV. He compared the results with foam and water.
Parv, **Akshit** & **Shubham** together worked on testing the opening detection algorithm on the UAV. During testing **Akshit** realized that the UAV is not following commands as it was considering the window as the obstacle. We tried turning off the obstacle detection from code but we couldn't turn it off. So, we covered the front stereo camera with black tape temporarily.
Shubham wrote the script to test sensor fusion algorithm for longer mission and was also responsible for debugging the drift issue.
Parv & **Shubham** worked on software cleanup on Husky and wrote one script to run the entire pipeline with just one command.

Future plans

1. **Shubham** and **Parv** will be working on integrating various software packages and create a pipeline for Husky and **Akshit** will working on the same for UAV.
2. **Akshit** & **Steve** will be responsible for setting up the microcontroller and relay switch-based circuitry for testing water pump, also they will on integrating thermal camera
3. **Akshit** will work on writing scripts and testing UAV mission with deployment capabilities
4. **Steve** and **Shubham** will be responsible for making the opening detection algorithm more robust.
5. **Steve** and **Parv** will work on writing scripts and testing AGV mission with deployment capabilities.
6. **Parv** will be working on global path planning in abstract simulation which we developed last semester.