MRSD Project 2

Individual Lab Report #09

Parv Parkhiya

October 24th, 2019

Team H:

PhoeniX

Teammates:

Shubham Garg
Akshit Gandhi
Zhihao (Steve) Zhu

Individual Progress

The first task for this progress review was to clean up the husky pipeline. The husky pipeline was manual and required launching 6 different nodes. There were some software conflicts with the other team with whom we are sharing the husky. Shubham and I worked on the pipeline cleanup of the husky. We created a new catkin workspace and forked the husky_core package so that we can keep our version for interfacing with husky without worrying about the other team modifying their husky core. We also created a new launch file and bash script which can run all the nodes and launch files required for getting Realsense cameras to move_base planner with a single command.

The next part was to figure out the DJI pose controller since we decided to move from the visual-servoing approach for entering through the opening. We had two options here. DJI provides pose controller through its SDK and the other was to use Airlab's core stack to use its pose controller which converts pose controller commands to velocity commands. Since we are running short on time and DJI doesn't have very good documentation on the APIs, we decided to get Airlab's pose controller. Akshit and I went through the code pose controller to see how the pose controller and what inputs does it require. We had to make some modifications to get the pose controller to work. Specifically, we had to make a pose controller accept the visual odometry data instead of traditional odometry which uses GPS and GPS is not available indoors. Once all those details were figured out, we performed flight tests to verify the functionality. While the drone was reacting to our commands, it was not going to the exact location. After careful observation, we figured out that the drone was taking goal locations with respect to some global origin. For our purpose, we would need to give goal location in the reference frame of the drone. So a transformation was required which would take goal point from the drone's reference frame to the global reference frame.

With pose controller somewhat figured out, I started creating a new package that can help align the drone parallel to the wall. I created the ROS interface with the Intel Realsense. With the experience of working with pointcloud on husky, the process was fairly straightforward. I used the PCL library to process the pointcloud and detect the dominant plane using the ICP method. The output of the same can be seen in figure 1. Sometimes the ground plan was getting detected as the dominant plane. I put various checks to eliminate nonsensical plans for consideration as a wall. Once the wall's plane was identified, I computed the desired yaw change for the drone that would align the drone with respect to the detected wall.

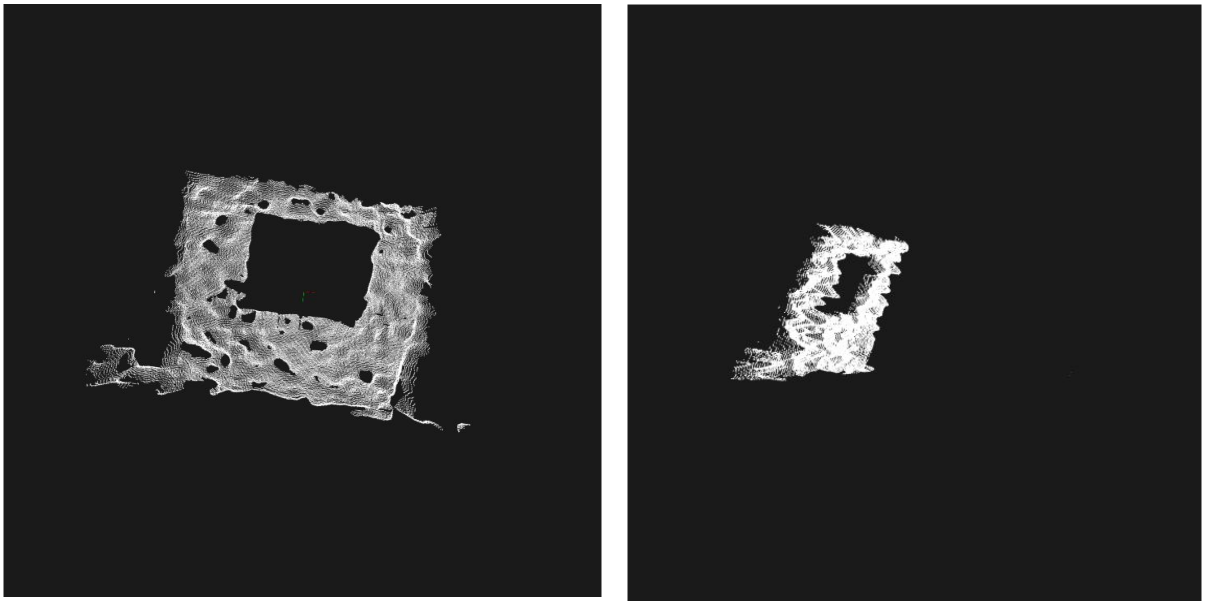

Figure 1: Pointcloud’s points on the detected wall plane

Shubham worked on opening detection using pointcloud. He was having a problem with getting the depth of the window correctly. We combined our packages to improve opening detection. Since my code was already finding the plane of the opening. Using our combined package, we can get the 3D location of the opening in the drone reference frame.

I was also responsible for the new extinguishing mechanism. I ordered a new set of water pumps, pipes, T-junctions, etc. We created a prototype as well as seen in figure 2. When we tested the new water pump, we realized that it was not powerful enough for our purpose. We compared the performance with our old water pump. The old water pump was working far better. We decided to keep the old water pump but replace the heavy container with lightweight plastic bottles.

I also worked on designing and creating mounts to attach the plastic bottles to the drone and modifying plastic bottles to have pipes to carry water. The final version of the attached extinguishing system can be seen in figure 3.

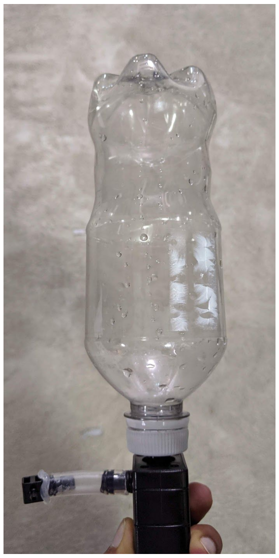

Figure 2: Extinguisher prototype with new water pump

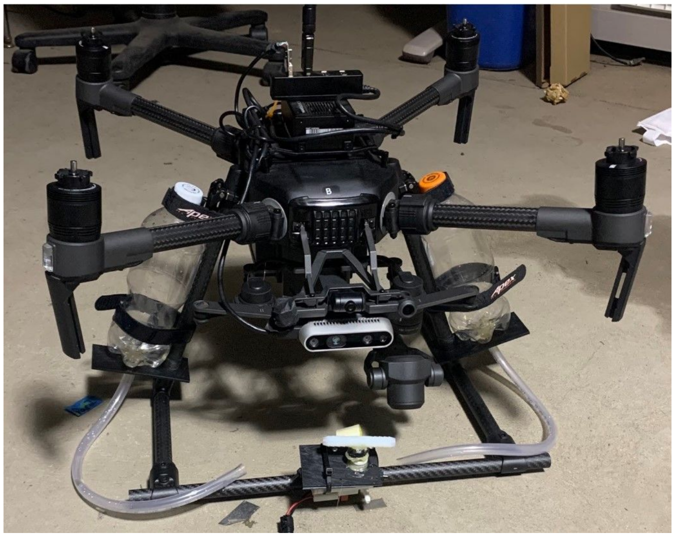

Figure 3: DJI drone with new extinguishing mechanism attached.

Challenges

The biggest challenge was to debug the problem with the DJI drone not entering through the opening. Even with our new approach of pose controller, we started facing problems where somehow drone was not responding to the command that we send. After many tests, we finally figured that DJI was ignoring our commands selectively based on its obstacle avoidance. We tried to disable it but when the drone is put on the offboard mode, it overrides the setting to turn off the obstacle avoidance. We covered one of the stereo cameras to stop the drone in detecting the obstacles which allowed us to successfully performed the flight through the opening.

Another major challenge was in designing mounts for the extinguisher. Since we don't have much time, we can not order multiple parts or 3d print them in time, we had to improvise based on what we have. We quickly designed and created mounts using scrap material in the machine shop.

Teamwork

We realized that if multiple people work together in debugging tasks, it’s more time-efficient in finding bugs. Shubham and I worked on the Husky Cleanup task together to start all the nodes with a single script and resolve conflicts. Akshit and I worked together to debug Shubham’s package that was detecting the window/opening using pointcloud. Akshit and I also collaborated in creating mounts for attaching the extinguishing mechanism to the DJI. Steve was looking into using form spray as an extinguisher. Together we performed multiple tests to see how far each mechanism throws materials under the high wind of the UAV. Steve and I performed a trade study to decide the extinguishing mechanism. Shubham and I also worked together to merge our packages together.

Future Plans

I would be working on getting the interface between the UR5 arm which is being controlled using Jetson and the husky which is being controller using Intel NUC. I would also work on the global planner which will provide an abstract trajectory that the UAV and the AGV can follow of maximum coverage with limited overlap. I would collaborate with other people in writing the pipeline for performing the full mission with all the parts integrated. As a team, we would like to perform short, independent missions on both systems and debug any problems we faced while integrating.