

Individual Lab Report #7

Akshit Gandhi

Team H (PhoeniX)

September 25, 2019

Team-mates:

Shubham Garg

Parv Parkhiya

Zhihao Zhu

Individual Progress

The tasks to be completed by the Progress Review 8 for me were: -

1. Work on a drone mission to fly through the window
2. Setup the AirLab Core SDK on the new DJI drone with the Manifold 2C

Drone Mission to fly through a window:

In the team, I was the primary point of contact for the AirLab software team hence all the responsibility of getting up in the air autonomously was on me. As a part of the task, I used image-based visual servoing package developed by team RAMS and with help of Karun, we modified it to suit our requirements of aligning in front of the window. To check if the algorithm was working properly, we tested in simulation with an Aruco marker where the 4 corners of the marker were used as a reference and the z-estimate of the marker center was used for the image-based servoing algorithm. The output of the simulation can be found in the video link attached [here](#).

Once the simulation results were as expected I started to modify the existing opening detection code to publish the 4 corners of the window and also using the depth image and the camera parameters, I started to publish the z-estimates for each of the 4 corners.

Since the DJI drone comes with an inbuilt front-facing stereo camera it was publishing the raw disparity images but they were very noisy and running any post-processing on it was not going work. Hence our sponsors told us to use the raw images from both the cameras and run our own stereo disparity node. As I was the only person on the team working with the DJI interface I took the responsibility to create a stereo disparity node. The images coming off from the cameras are 240p (very low res), the DJI SDK does not publish any camera info parameters, hence I had to calibrate the stereo pair and write a time-synced camera_info publishing node. Once the input for the pipeline was ready, I feed it through ROS stereo_image_proc package to compute the disparity. The initial approach was to use the package but I didn't know that the images were not time-synced and hence I was not getting output! With setting a queue_size and an approximate time sync parameter I got it to work.

Using a semi-global matching algorithm has a lot of hyperparameters which need to be tuned hence I spend a lot of time to fine-tune the knobs. The result in color (shown below) is the output from my node and the output in black and white is DJI's disparity map. We can see a vast difference in the output quality! The texture-less walls are not detected by the disparity node which was a challenge and hence we reverted back to the Realsense D435.

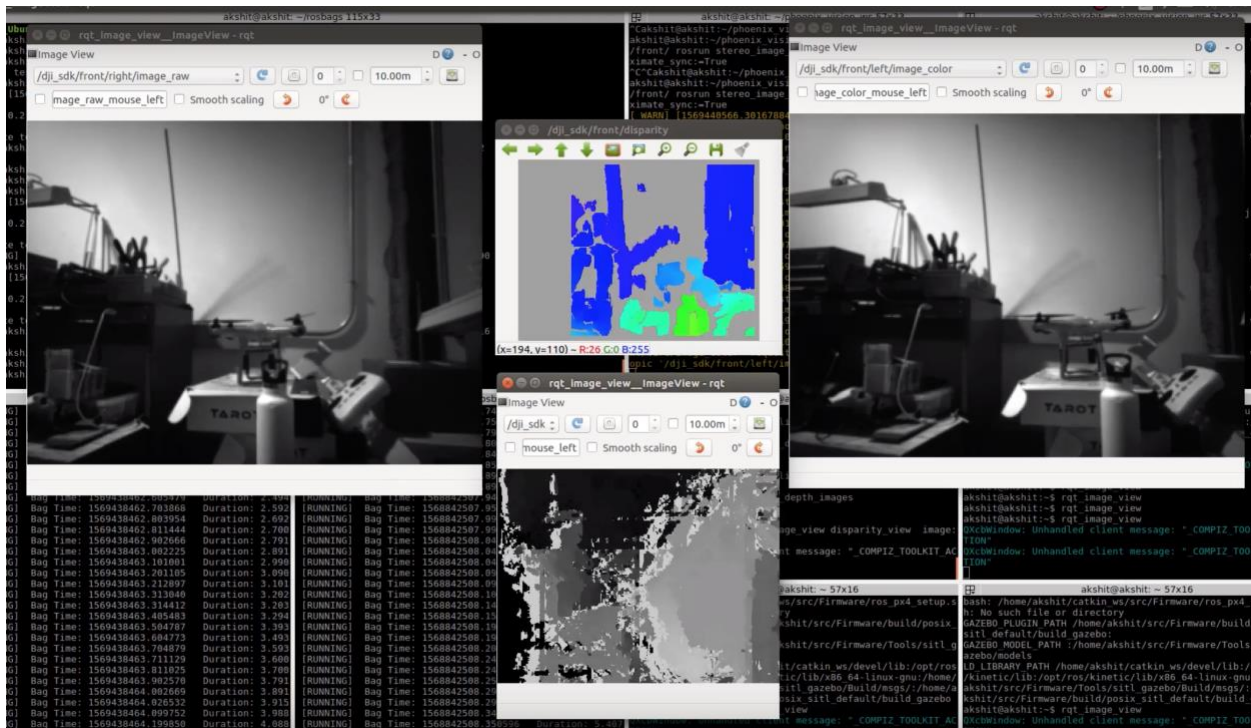


Figure 1: Stereo Disparity node output

Since we decided to revert back to the realsense, there was a question on where to mount the camera and how to mount it. Fortunately, Kevin Zhang (R.I Ph.D. student) and Oliver's student helped us out with a simple mount design and I 3D printed the mount + attached it on 2 drones along with the camera.

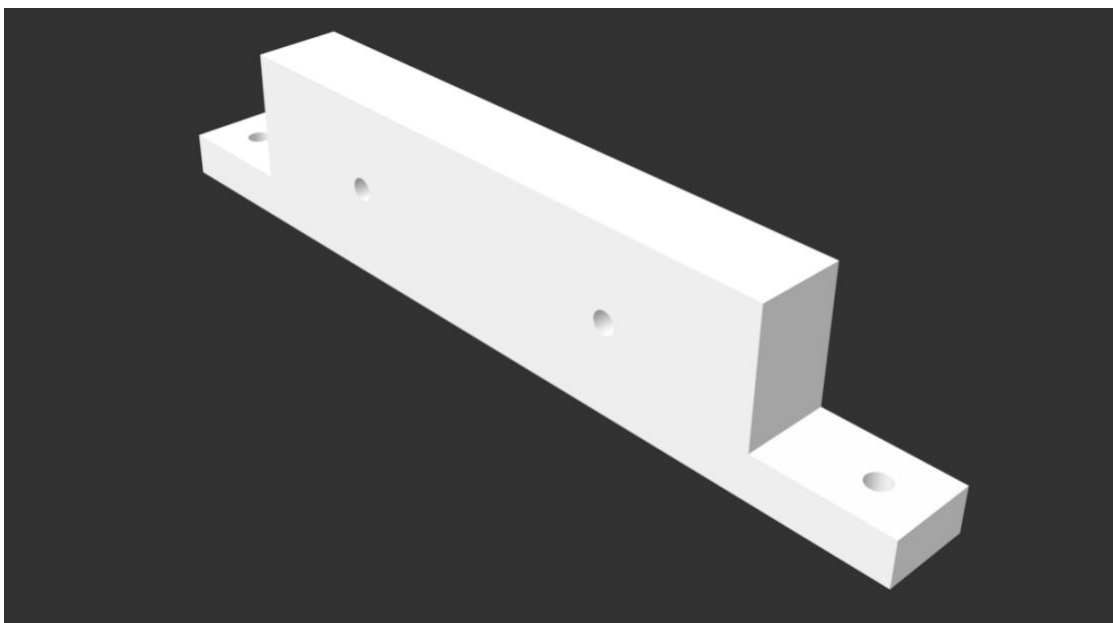


Figure 2: Realsense DJI Mount

Once the camera was in place on the drone, we were ready to test fly and move closer to the progress review goal for the drone, but the Manifold computer on the existing already setup drone had a jetson computer which had to go through a kernel patch in order to use the realsense. With prior experience on the tx2 in the spring semester, during the kernel patch we lost all the data on the OS (3 times! Fortunately, we had backup of the system image) and this time I didn't want to go that route because there was just one working drone for the whole MBZ team of 18+ members and my single mistake could've cost us the deadline. Hence, I ventured the route to setup a DJI Manifold with the intel NUC.

This setup took me 1 day as the core stack by AirLab as not compatible with ROS kinetic and hence some modifications had to be figured out and hence compiling the code and making sure it works took some time. Once I got done with the setup, we started to get low CMOS battery warning on bootup. An initial approach on replacing the battery showed up that the battery was hard glued on the circuit board, hence I had to take another NUC and set it up from scratch. The problems could've been severe if we used the low battery CMOS computer as the Date, time and BIOS settings could've caused a hardware error according to the documentation. Hence another 6 hours were spent on setting the new computer up.

The new DJI Matric M210 V2 was not getting setup properly by the students in AirLab and since we had to use it anyhow for the PR and September deadline, I spent a chunk of time on getting things to work as they should (like proper serial communication, testing arming, and takeoff commands) on the new drone and pc. At this point, it was already Tuesday 24 September 2019 8 PM (a day before PR) and we did not get a chance to test the visual servoing.

Miscellaneous contribution:

Given that the scope of the project is too large and a lot of deadlines were on the verge it was implicit that people wear multiple hats and help the other team-mates when they need help. Hence I did play on and off roles by helping with the camera and URDF setup mainly by helping the team with the axis aligned with the actual STL.

As some assignments were due, Parv was the only one who was working on the husky and in order to test his setup he needed extra personnel, hence I helped him out test the husky obstacle avoidance and entering a doorway.

Team also needed some of my help with time sync for the point cloud messages that were being published by the realsense cameras on the husky and hence I worked on the async time filter on ROS for the husky. Apart from this, my roles have been from debugging installation, build issues especially, with cpp and OpenCV.

Challenges:

1. Setup of disparity node with proper ROS time sync between messages for the DJI drone
2. Installing AirLab core stack and debugging issues with the kinetic build
3. Learning about the dead CMOS battery on the computer and reinstalling the whole thing back on a new computer
4. Setting up the new drone and the on-board computer with proper user permissions

Teamwork:

Shubham was working on the husky URDF and camera setup and he worked with Parv in order to fuse the point clouds for a total of 160-degree coverage. Parv also worked on converting the point cloud into a laser scan, feeding it to the husky gmapping package which generated a costmap for us which was then fed by his node to the husky move base package which did the local planning for us. Parv and I also worked on the testing of the entire pipeline on the actual robot. Zhihao was working on the door detection algorithm as the challenge over there was slightly different compared to the window where we see all the 4 sides of the object to be segmented compared to the door where at a time we can just see only the 3 sides (mainly as the camera is mounted way close to the ground).

Future Plans:

The plans for the next presentation are to get the communication subsystem ready and thus the responsibilities are divided as follows:

1. Shubham - Selection of a WIFI router with a suitable bandwidth range.
 2. Akshit – Setup of a central database and a schema to store ROS messages and fire locations as world coordinates. Work on the autonomous door entering test for the UAV
 3. Parv – Setting up a cross-machine ROS communication
 4. Zhihao – Continue his investigation with door detection
- As a stretch for the next PR we may also look into an improvised fire extinguishing system with a lighter payload and higher projectile range (with a possibly powerful pump)