Harikrishnan Suresh

Team C: Fly Sense
Teammates: Shivang Baveja, Nicholas Crispie, Joao Fonseca, Sai Nihar Tadichetty
ILR07
Feb 15, 2018

Individual Progress

My major contributions after the last progress review has mainly been in setting up the software stack on our new onboard computer Jetson TX-2. The Jetson TX-2 module is mounted on the Connect Tech's Orbitty carrier board, which helped us reduce the weight of our onboard computer to around 185 grams. We plan to reduce the weight further by shaving off some portions of the heat sink. Figure 1(b) shows the current status of the onboard computer.

ROS Kinetic along with all the necessary packages were installed on the TX-2. The FlySense fall semester software stack was ported to the TX-2 and successfully compiled. The bird's eye view of the simulation version of our stack is shown in Figure 1(a).

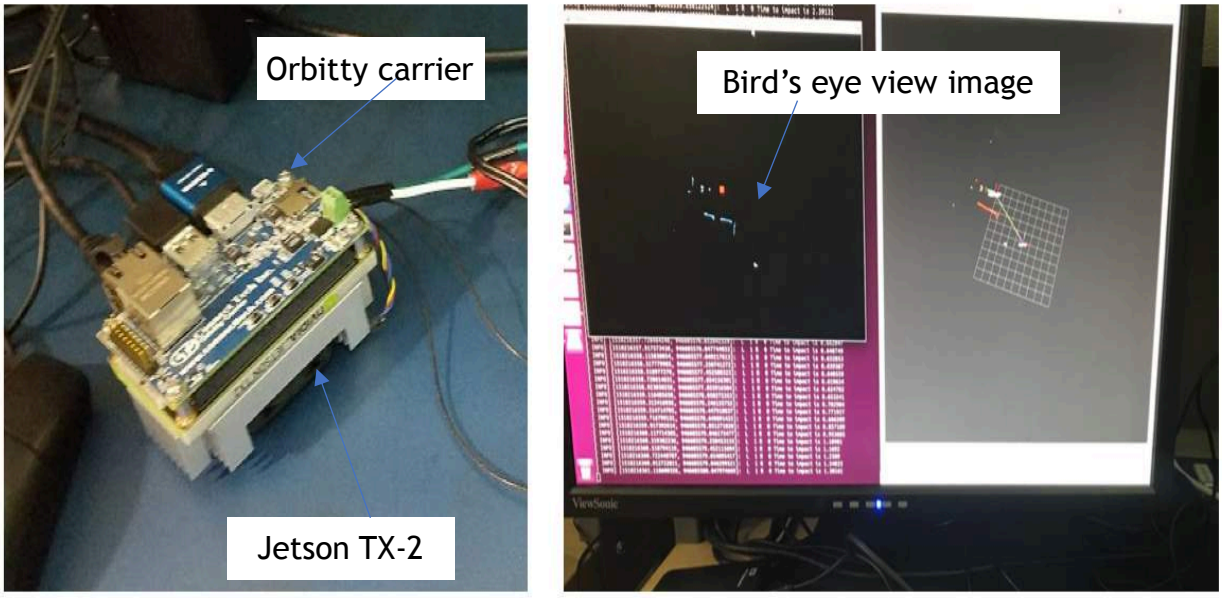

Figure 1: a) TX-2 on carrier board b) Output of FVE sim

I also worked on the code to get list of relevant obstacle points in 3D for sound warnings. Last semester, the sound warnings code was implemented in 2D where the obstacle points were all projected to the same plane as the cart and used for computing the most critical one based on the time to impact. The same package 'velodyne_height_map' was modified to list out the obstacles in 3D.

The algorithm used to generate list of obstacles in 3D is given below, and also in Figure 2:

1. The environment is first discretized into a square grid of user specified dimensions. The cell size is fixed at 5cm. In our case, the flight dynamics will decide the dimensions of the grid, just like the crop box filter that comes earlier in the pipeline.
2. Every input velodyne point is converted to a cell in the grid. We only consider those points that lie inside the grid boundary at this time step k.
3. At the end of the time step k, cells in the grid will be populated with multiple points. For every cell, the maximum height and minimum height among the points is stored, and their difference Δh is computed. Only those cells where $\Delta h \leq threshold$ will be considered. The threshold is the minimum height of obstacle we consider as relevant.
4. From the cells considered, all the velodyne points (x and y) will be replaced by the center of the cell (x and y).
5. The original algorithm is modified to give the height output as the maximum height of that cell, instead of just projecting to the plane of the velodyne sensor.

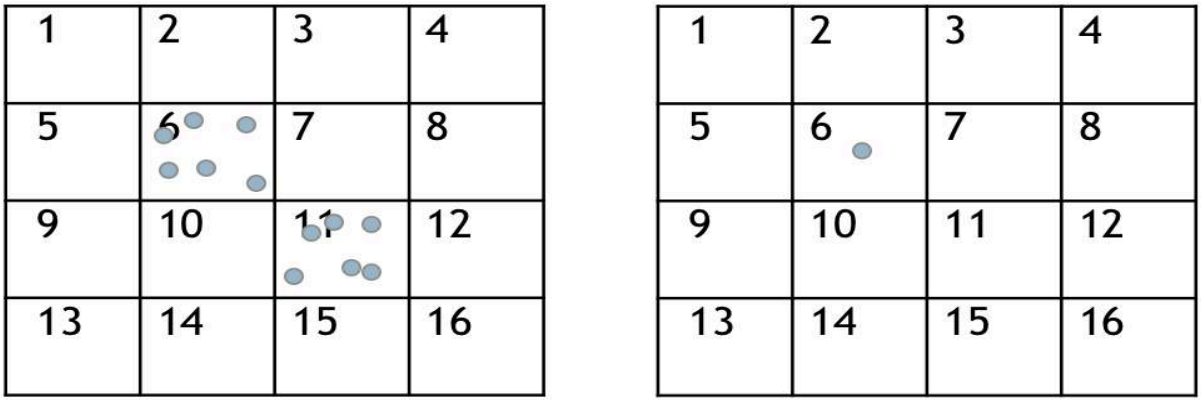

Figure 2: Velodyne height map algorithm

The output is velodyne height map is shown in Figure 3. The red dots (Figure 3a) show the relevant points of the obstacles, and Figure 3b shows the velodyne raw input.

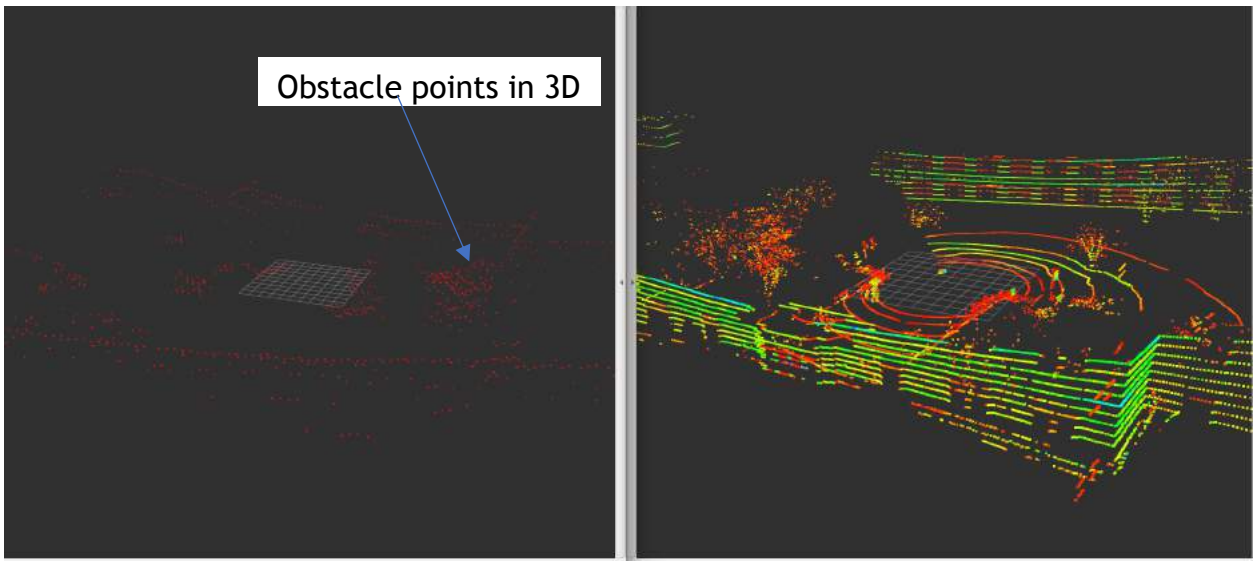

Figure 3: a) Output of velodyne height map b) Raw input

Challenges faced

- The interface of velodyne with the TX-2 caused problems as the velodyne raw data was not being displayed on Rviz. It seems to be some Ethernet or driver issue and will be addressed immediately.
- For the velodyne height map algorithm, the raw data height values are used to decide whether the obstacle is valid or not. The height values are with respect to the velodyne sensor's position, and not absolute with respect to Earth. So, either a correction term has to be applied based on the height of the quadcopter or the algorithm must include this relative height value.
- Due to weather and tight academic schedule, the team has been lagging behind on the flight testing.

Team work

Team Member	Contribution
Shivang Baveja	• Conducted static flight tests with dummy weights to check payload capacity • Conducted flight dynamics tests to calibrate the quadcopter model to design the control algorithms • Assisted in obstacle avoidance control algorithm development

Nick Crispie	• Assisted in both the static and flight dynamics tests • Procured the hardware to design velodyne power module
Joao Fonseca	• Developed the algorithm to color obstacle point clouds and find the critical points for sound warnings in 3D • Developed the algorithm for obstacle avoidance algorithm
Nihar Tadichetty	• Setting up ground station for image segmentation • Research on speech recognition algorithms • Faced a huge setback due to damage to his laptop

Tasks

Team goals

- Complete the onboard hardware package setup – cable replacement and power module for velodyne PUCK, mounts for the PUCK, TX-2 and FPV camera
- Demonstrate the Fall semester software stack working for the quadcopter in air, with live data from velodyne processed onboard
- Implement the obstacle avoidance algorithm with interface to the DJI controller and test it in simulation
- Test FPV video reception on the Epson standalone

My goals

- Get the Fall semester software stack working on onboard TX-2
- Assist in conducting flight tests and collecting data
- Implement sound warnings algorithm in ROS and test it on the quadcopter