



Harikrishnan Suresh

Team C: Fly Sense

Teammates: Shivang Baveja, Nicholas Crispie, Joao Fonseca, Sai Nihar

Tadichetty

ILR10

April 5, 2018

1.

1. Individual Progress

1.1. Obstacle Coloring

The past week has been super productive for me, as I was able to complete all the tasks that were promised for this progress review. My major contributions were to the obstacle coloring and the sound warnings, and the final code integration.

In obstacle coloring, a few design options were proposed during the 2nd pilot workshop at NEA. The details of the same have been listed in Table 1. From the proposed designs for the quad icon, the blue arrow was preferred by the pilot as it gives a sense of the heading and the tail. It was scaled down based on the size of Bird’s Eye View.

	Options	Finalized
Icon (Color + Shape)	<ul style="list-style-type: none"> • Icon options - Triangle, Simple Arrow, Arrow with Tail, Quad sample image • Color options - Blue, Brown 	<ul style="list-style-type: none"> • Simple blue arrow (scaled correctly)
Single Color / Multiple Colors	<ul style="list-style-type: none"> • Single color obstacles <ul style="list-style-type: none"> ○ Kd-tree with KNN approach to extract clusters and segment obstacles ○ Metric - Euclidean distance • Multiple colors obstacles 	<ul style="list-style-type: none"> • Multiple colors (inflated to look nicer, factor of safety)
Ground Points	<ul style="list-style-type: none"> • With ground • Without ground <ul style="list-style-type: none"> ○ Pass-through filter when altitude <1m 	<ul style="list-style-type: none"> • With ground (pilot able to relate better to surrounding)

Table 1: Outcome of pilot workshop 2

The single-color option for coloring required the introduction of a new clustering and segmentation algorithm into the perception pipeline. The obstacles in the environment were grouped into clusters using a Kd-tree data structure to identify the K nearest neighbors to individual points. The neighbors were selected by using Euclidean distance as the metric to threshold. The parameters that worked for us was a distance threshold of 1m, and cluster points threshold of 50. The algorithm worked real time and was also capable of removing sparse outliers in the point cloud. However, the pilot preferred the multi-colored obstacles as it always indicated a sphere of danger (red) while flying.

Finally, it was decided to keep the ground points as it helped the pilot relate better to the surroundings and the flight scenario. The z coordinate of the dynamic window was changed to -1 so that the ground points appear only during takeoff and landing. Figure 1 clearly proves the need to have ground points, as it helps to distinguish between two possible redundant images on the AR interface.

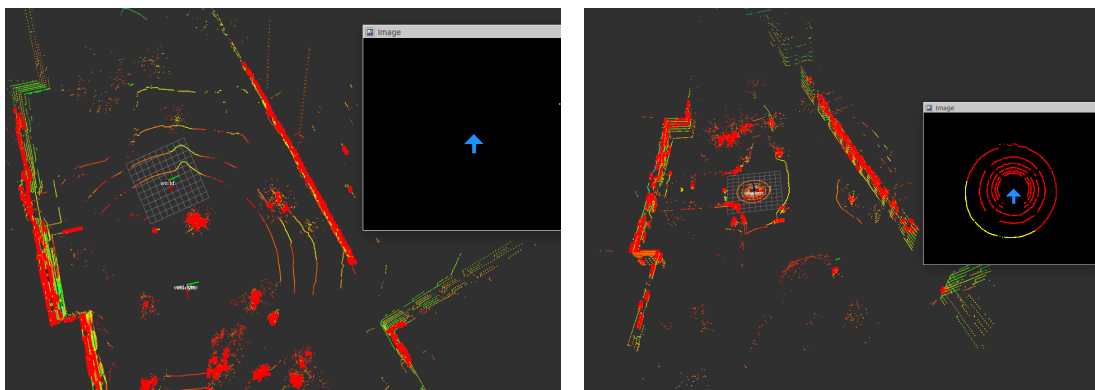


Figure 1: Quad in air and no points on BV (left), Quad before takeoff with ground points on BV

1.2. Sound Warnings

To implement sound warnings, it was necessary to filter out all the unnecessary points from the raw velodyne data. Since the time to impact calculation for sound warnings was done using the conventional Newton-Raphson method, the sound generation could have been really slow with the heavy processing. To select only the relevant points, the velodyne height map package backend was modified to give obstacle points in 3D. It involved a 2D grid-based approach where all the points in a particular cell are replaced by the centroid and the height is taken as the mean height of all points in the cell. A grid resolution of 0.2 m was chosen with a total grid size of 300 to give 30m coverage on all sides of the quadcopter. This helped to reduce the number of points from 4000 to around 1000 and is seen clearly in Figure 2.

A few important points to note in the sound warnings code are:

- It is processed parallel to the coloring code, and will provide an additional input to the perform the blinking in the coloring code
- The time to impact depends on the current pilot input and warnings appear only if the pilot tries to fly towards an obstacle at a fast speed
- The code functions only when the quad is above 1m altitude to prevent sound warnings from appearing during takeoff and landing.

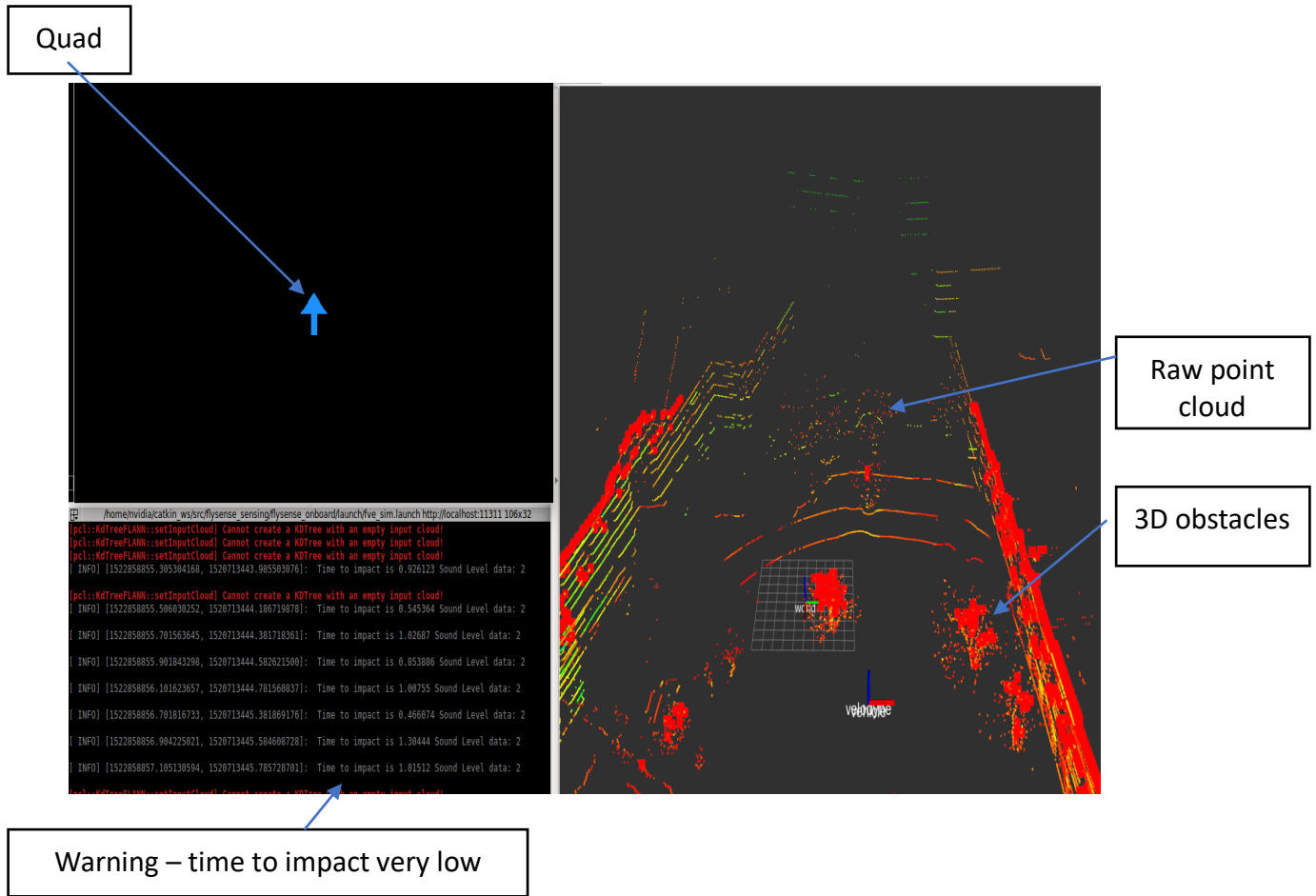


Figure 2: Sound warnings implementation

1.3. Fully integrated software stack

All the elements in the perception pipeline are now fully integrated and work well on the Epson AR interface. Figure 3 shows the output of the current version in Rviz and Figure 4 shows the output on the Epson.

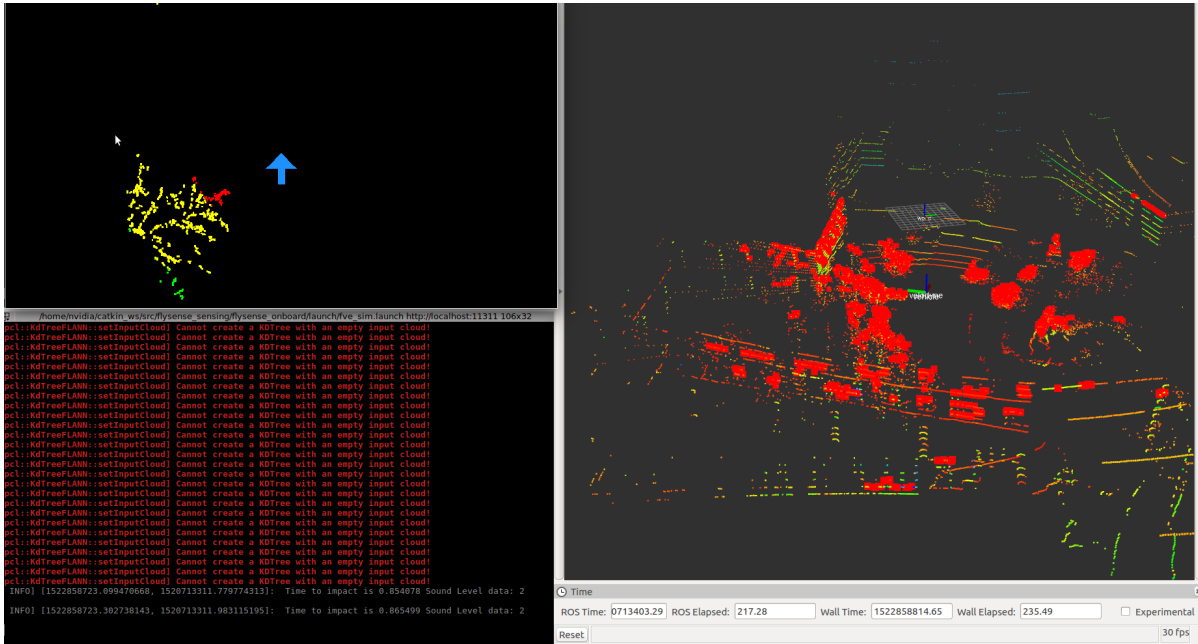


Figure 3: BV with 3 colors (top left), Sound warnings terminal output (bottom left), Obstacles in Rviz (right)



Figure 4: Final interface on Epson (FPV, HUD and BV combined)

2. Challenges faced

- Again, my laptop stopped functioning at a critical time of the project. This limited me to work directly on the Jetson, and I could no longer work remote. I had to also manage the schedule with Shivang and Nick, who wanted to test obstacle avoidance on the Jetson.
- The night before the progress review, the bag file which had our FPV video recorded got corrupted. As a result, we were not able to demonstrate our new interface on the AR

headset. However, we were able to recover it and an image captured from the headset is shown in Figure 4.

- Due to constant rain during the entire week, we were unable to perform the flight test at Flagstaff hill with our new interface.

3. Team work

Team Member	Contribution
Shivang Baveja	<ul style="list-style-type: none"> • Implemented pilot stick override code and tested in DJI flight simulation • Worked with Joao on emergency braking code for obstacle avoidance and is now testing and tuning • Worked with me on optimizing the coloring code and making it real time
Nick Crispie	<ul style="list-style-type: none"> • Worked on the Gazebo environment to simulate the whole flight experience – integrated the DJI Matrice 100 with velodyne
Joao Fonseca	<ul style="list-style-type: none"> • Collaborated with me in implementing the sound warnings code and testing it • Collaborated with Nihar on the front-end side of the sound warnings • Worked with Shivang on the obstacle avoidance code
Nihar Tadichetty	<ul style="list-style-type: none"> • Completed the front end of the new interface – refined the locations of the HUD and BV • Worked with Joao on the front-end side of the sound warnings

4. Tasks

Team goals

- Prepare the completed software stack for the first flight test at Nardo with NEA pilot
- Conduct more flight tests to improve the algorithms
- Complete enhancements to the Bird's Eye view like blinking of obstacles
- Complete the emergency braking for obstacle avoidance and test it with the actual quad
- Complete Gazebo simulator and integrate it with the DJI flight simulator

My goals

- Complete enhancements to the Bird's Eye view like blinking of obstacles
- Fine tune coloring and sound warnings code based on the feedback received from the flight tests
- Prepare the completed software stack for the first flight test at Nardo with NEA pilot
- Prepare the system for the National Robotics Week demo
- Start working on official project video for the National Robotics Week demo