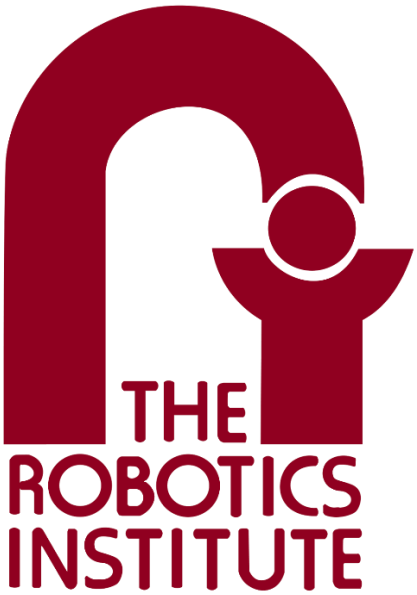
Individual Lab Report #3

VIVEK GOPAL RAMASWAMY

TEAM E- BEYOND SIGHT

Team Members: Chien Chih Ho, Pengsheng Guo, Oliver Krengel, Rohit Murthy, Vivek Gopal Ramaswamy

27th October 2017



Individual Progress:

As a part of the team, I was responsible for the following

1) Solving the problem of GPU compatibility with ZED.
2) Interfacing ZED stereo Camera with Ubuntu 16.04.
3) Getting the standalone application of the ZED called the Depth viewer to launch and show depth images of the environment.
4) To test the range of ZED Camera.
5) Installing ROS Kinetic and the ROS wrapper by ZED and using Rviz to visualize the Depth point cloud.

Progress in detail:

1) **ZED Explorer and Depth Viewer:** I followed the instruction [1] given by ZED, to install the necessary tools required for running the ZED camera. Some of the important packages which were required for the proper functioning of the device are stated below
   a) OpenCV 3.0
   b) CUDA 8.0
   c) ZED SDK v 2.0

After the successful installation of the packages, I could run both the ZED Explorer and the Depth Viewer.

ZED Explorer: This is a standalone app provided by ZED to get a live video with both the cameras. I was able to run and check it by using the below command. Refer the figure 1, for results.

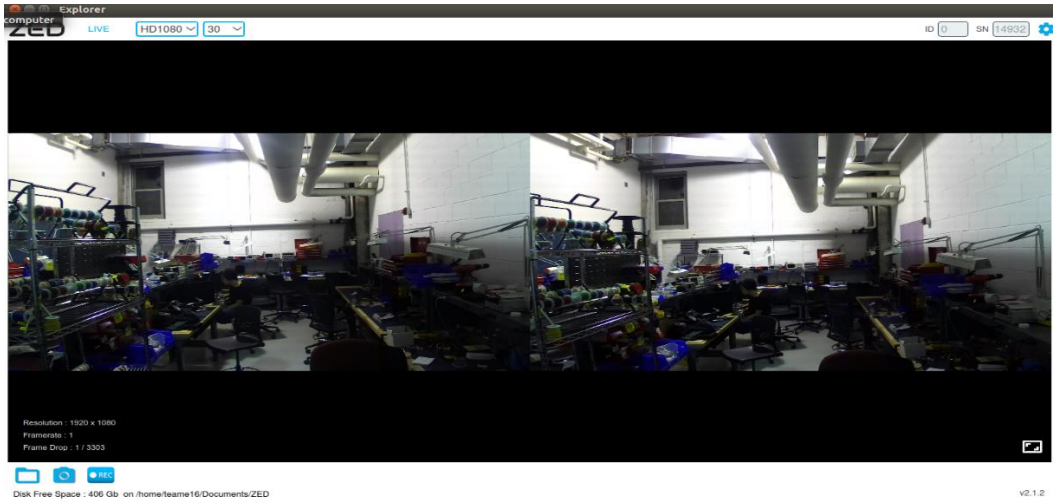


Figure 1. ZED Explorer

sudo ./use/local/zed/tools/ZED \Explorer

ZED Depth Viewer: This is also a standalone app provided by ZED, which gives PCL, depth map of the environment. The depth images were taken at 720P,@60 FPS. Refer the figure 2 for the results.

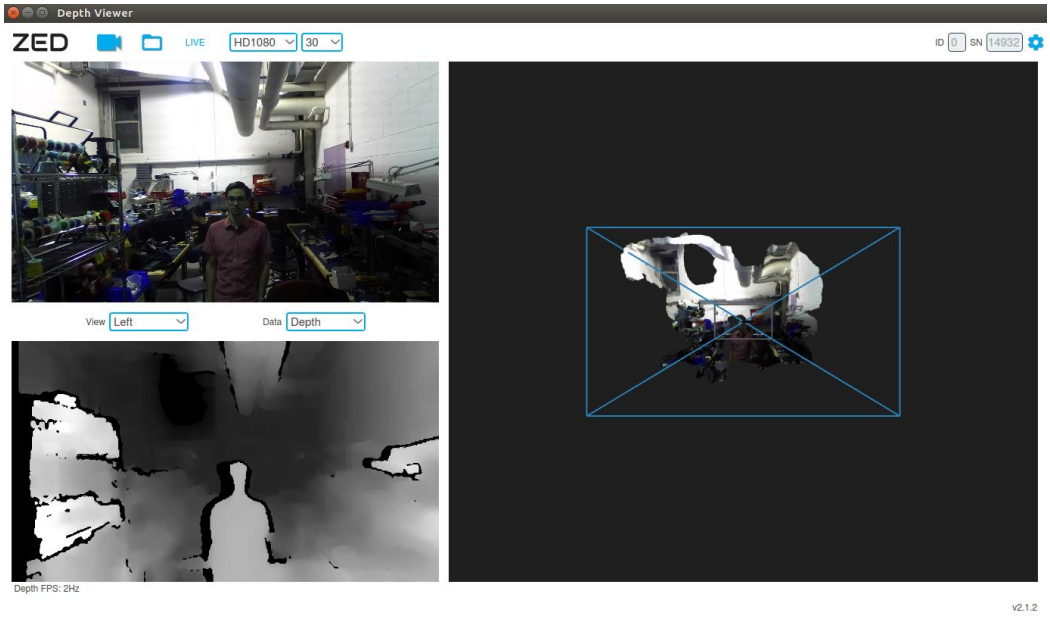


Figure 2. ZED Depth Viewer

2) **Range Testing:** Once the Depth Viewer was running, I had the ZED setup on the sensor mount and had asked one of our teammates to walk around, from the distance of 20, 15,10 and 5 meters. From the experiments, above, I concluded that the camera could perform only up to a range of 10 to 12 meters. This was important to the team, as for the performance requirements we had taken 20m as the maximum range.

3) ROS Installation and Rviz Visualization

a) For Ubuntu 16.04, I followed the steps from ROS wiki and had installed ROS Kinetic.
b) Followed directions to install the ZED wrapper for ROS.
The Zed depends upon the following ROS Package [2].

1) tf2_ros
2) tf2_geometry_msgs
3) nav_msgs
4) roscpp
5) rosconsole
6) OpenCV

Starting the node:

roslaunch zed_wrapper zed.launch

This publishes all the data on the various topics. I used two topics for generating depth images and point cloud.

a**)/camera/depth/depth_registerd:** Subscribing to this topic and visualizing, I got a depth image map in grayscale.

b**) /camera/point_cloud/cloud_registered**: Subscribing to this topic, gave me a 3D colored point cloud as shown in figure 3 below.

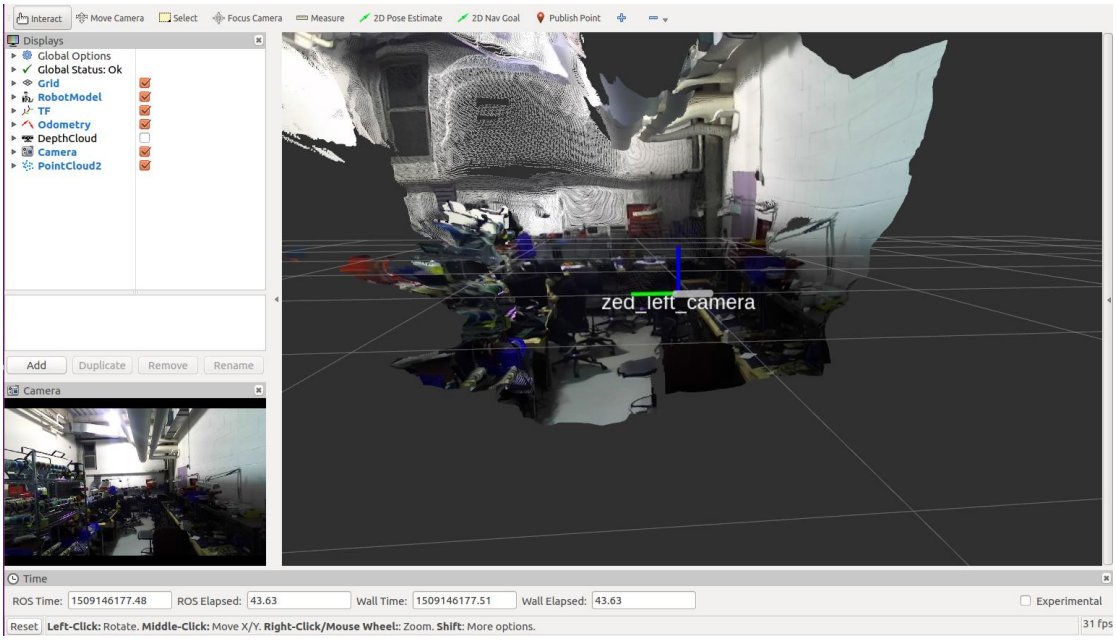


Figure 3. Point Cloud in Rviz

Challenges Faced:

1) **GPU compatibility:** After going through the ZED related blogs, I found that, the older SDK version 1.0 which I was using it with Ubuntu 14.04 did not have any support for the TITAN X with graphics driver version 385. But since only the version 1.0 SDK could be used with

Ubuntu 14.04, I upgraded to 16.04 with SDK version 2.0 which had support for our GPU and therefore solved the problem.

2) **CUDA Error[2]: GPU out of memory:** While launching the zed node, I was getting this error, frequently. To solve this, I had referred to the internet where I had found some useful commands for checking in the system usage of GPU. One of them was the nvidia-smi [3] .Executing this command gave me a list of the process which was being executed by GPU. I was shocked to see the result because I was already running 5 instances of the ZED depth viewer, which almost took the entire memory. To solve this, used pkill to kill all the instances of ZED and then launched my Zed node.

3) **Computer Usage Time:** I found it little difficult to use the common computer of the team, considering that, two of my teammates were already working on it. To solve this I gave my team a suggestion of allocating time for all of the team members on the common computer.

Team Work:

In my view, this week we as a team had really covered some ground in terms of achieving our milestones. The team had put in extra hours to make sure that, we are in sync with the schedule.

We also after having multiple discussions and feedbacks from various sources had changed our performance requirements, in a way that it becomes practical for us to measure.

One more important change we brought about, was that we decided to meet daily, to discuss the work to be done and this really helped us in gauging our performance at the end of the week.

Vivek Gopal Ramasamy: ZED camera installation and visualization of depth images and point cloud using Rviz.

Chien Chih Ho: He was responsible for the pedestrian detection using CNN, Fast and Faster RCNN, with the pre existing model.

Pengsheng Guo: He was responsible for implementing Euclidean clustering and also for finding the centroid of the pedestrian.

Oliver Krengel: He was mainly responsible for the teleoperation of the previous Team D's car. He also with the help of Rohit Murthy, designed the PCB for our project.

Rohit Murthy: Apart from helping Oliver in the teleoperation and PCB, he was mainly trying to understand the tensor flow structure for implementing the prediction model.

Current Project Progress and Future Plans

In terms of the project, we have

1) Perception pipeline ongoing with clustering and base level pedestrian detection ready.
2) Previous Team D's car running with teleoperation.
3) Prediction pipeline in progress using either PyTorch or TensorFlow.

Our Future Plans include:

- On the learning part, we will try to learn and understand on how to overlay RGB images on LIDAR data.
- Detection of the pedestrian from the live feed.
- Learning and implementing basic LSTM models.
- Implementing basic motions with waypoints on Team D' car.

References

[1] https://www.stereolabs.com/documentation/overview/getting-started/getting-started.html

[2] https://www.stereolabs.com/blog/index.php/2015/09/07/use-your-zed-camera-with-ros/

[3] http://nvidia.custhelp.com/app/answers/detail/a_id/3751/~/useful-nvidia-smi-queries