



11/25/2015

# Progress Review 4

Individual Lab Report #5



**Abhishek Bhatia**

**Team D:** Team HARP (Human  
Assistive Robotic Picker)

**Teammates:** Alex Brinkman, Feroze  
Naina, Lekha Mohan, Rick Shanor

## I. Individual Progress

For the Progress Review 4, my primary responsibility was to procure the laptop with USB3.0 support and set it up to make it work with Kinect 2. I shortlisted 2 laptops: 1) Dell Inspiron i7559-763BLK (This one had an I-5 processor, an on-board graphics card, costs \$799 and weighs around 5.6 pounds) and 2) Lenovo Flex 3 (This one does not have a graphics card but has an I-7 processor, costs \$749 and weighs around 4.6 pounds, Figure 1). The only concern we had was linux compatibility. I researched online and also talked to some of the seniors at CMU and discovered that there have been compatibility issues between Ubuntu 14.04 and Nvidia graphics cards. Based on this we decided to go ahead with our option 2), the Lenovo laptop with an Intel integrated graphics processing unit and a better central processing unit.



**Figure 1: Lenovo Flex 3 laptop**

After placing the order for this laptop, I started working with Rick on an issue related to the perception system. We had developed the complete vision pipeline and tested it on the Rutgers/Berkeley database to get a decent accuracy. Once that was achieved, we decided it was time to move on to testing the vision pipeline with real-time Kinect data. Rick started interfacing Kinect 2 on his laptop and started carrying out the initial tests. One problem he faced during this was regarding the minimum distance of the Kinect 2 from the shelf. Although the operating range of the sensor was supposed to be between 0.5m - 5m, it was not generating point clouds for that range. Rick had to keep the Kinect 2 far away from the shelf which was decreasing the overall resolution of the feed. This was causing the vision

pipeline and the existing algorithm to perform below the expected threshold and eventually giving us the reduced accuracy.

To mitigate this issue, we decided to test the vision pipeline with real-time input feed from Kinect 1 in parallel. We had earlier decided to go ahead with Kinect 2 only because of the higher resolution capability of the Kinect 2 camera. I started working on setting up the Kinect 1 on my laptop. I used the `freenect_stack` ROS interface to Kinect using the `libfreenect` library. Further I used the `openni_launch` ROS package [1] to open an OpenNI device (Microsoft Kinect 1) and load all nodelets to convert raw depth/RGB/IR streams to depth images, disparity images, and (registered) point clouds. I used the `rviz` package to visualize the data and make sure I installed everything correctly. After that, I used the `pcl_ros` interface stack [2] as a bridge for 3D processing and applications in ROS. I developed a node similar to `pointcloud_to_pcd` that subscribes to a ROS topic and saves point cloud messages to pcd files. Finally, we inputted these pcd files generated from the Kinect 1 input feed into our vision pipeline to generate bounding boxes around the objects and made a prediction for our desired object by comparing to the ground truths already available. But, as expected, we were not getting a great accuracy, mostly because of the low resolution output of Kinect 1. With this we decided to contact the person who had written the `libfreenect2` drivers for Kinect 2 and discuss the issue related to the operating range that we were facing, which later turned out to be fruitful for us as this person found out a small bug in his driver code and fixed it. This eventually solved our problem with Kinect 2 and we were successfully able to interface Kinect 2 with our vision pipeline and achieve expected accuracy.

After this, I received our new laptop and it was my responsibility to get it up to speed to be able to use it for our vision processing. Feroze helped me with installing Ubuntu 14.04 with an option of dual-boot on this laptop. But as soon as we installed ubuntu, we figured out an issue related to the wireless connectivity on this laptop. I was not seeing any wireless device on the list of peripherals and found out that the issue was related to wireless drivers not being present. After doing extensive research about this issue, I found out that this was a common issue and people had resolved similar issues by kernel updates to the latest kernel version. To fix this, I downloaded .deb kernel related files on another laptop and manually installed them on the new machine. But I still needed it to connect to internet somehow because I needed to do “upgrade” and “update”. This was a nuisance as I figured out there was no way to connect to internet on the campus using ethernet/LAN. I tried to establish an Ethernet connection on this laptop at my home, but was not successful in doing so because of some issues. Finally, I researched more about this issue and discovered a way to connect to the internet using external wireless LAN network interface controllers. I knew that some of the teams in MRSD were using ODRROID's for their projects and I borrowed an external wifi card (Figure 2) from Team E. This solution worked instantly for me, I was able

to connect to wireless LAN using this external wifi card and did the updates on my system. Finally, after a day I saw our new laptop getting connected to the wifi on a reboot.



**Figure 2: The Realtek RTL8188CUS-GR single-chip Wireless LAN (WLAN) USB2.0 network interface controller compatible with the 802.11n specification**

Finally, this week I also started working on developing a ROS wrapper for our vision pipeline. This wrapper should interface with Kinect 2, subscribe to its input pointcloud stream and publish pcd data to the vision pipeline on the input side and takes the vision pipeline output (xyz grasp location of the item of the interest) and inputs it to the state controller. Currently, I am stuck with an issue related to the interfacing of Kinect 2 with the new laptop. I am using a libfreenect2 based driver [3] and [4] for Kinect 2 and looks like there is an OpenGL based dependency that is giving me the error. I plan to work on resolving this issue this week.

Besides this, I also helped Rick, Alex and Feroze with the task of setting up the final gripper subsystem together for this week's progress review.

## II. Challenges

The major challenge I faced this week was enabling the internet connection on my new laptop. I am always amazed by how much support internet provides in resolving these technical issues faster. I understood that this issue was related to a kernel update, the new laptops have new chipsets and the drivers required to make these chipsets work are not available with older versions of open source software packages. I did not find reference to my Intel wireless card directly, but found a post that discussed a similar issue with an older version of wireless card from the same Intel family. I worked on the assumption that what worked for this person can also work for me and it did. Besides, I also felt like discussing the problems with batch mates helps a lot. Only after discussing this with Rushat, I discovered they already had an external wifi card that I could use, other option was to buy one such card which would have caused huge latencies in getting these things up.

## III. Teamwork

For this week, I wanted to work on defining the shelf transforms in the simulation for our progress review, but as we always do, we analyzed our deliverables and figured out Rick needed support in the perception block with respect to testing Kinect 1 in parallel. Feroze had already worked with defining transforms in simulation and it made sense for him to work on defining the shelf transforms and made more sense for me to help Rick with the perception related deliverables, owing to my prior experience with the vision pipeline. This was one form of load balancing we did to successfully finish all our deliverables for the progress review in time.

**Alex:** Alex continued with his work on the state controller and worked with Feroze to integrate PR2 base movement with arm planning and state controller. He also worked with Rick to complete the electronics box for the suction subsystem.

**Feroze:** Feroze worked on defining the shelf transforms in simulation for the demo and worked with Rick to solder our final PCB board.

**Lekha:** Lekha worked on publishing ROS messages encapsulating XYZ position for items of interest using Kinect 1.

**Rick:** Rick primarily focussed on setting up the interface between Kinect 2 and vision pipeline. He also worked on setting up the electronics for the final gripper subsystem.

**Abhishek:** I worked with Rick on perception deliverables, worked to procure and setup the new laptop. Besides, also worked on setting up the final gripper subsystem design.

## IV. Future Plans

Our plan for the coming two weeks is to prepare for the Fall Validation Experiment demonstration and will focus all our energy to refine the 2 experiments (gripper and perception experiment). I will be working on adding the neck and gripper initialization control in our state controller. Besides, we will be working together as a team to carry out extensive testing on both our subsystems up for demonstration during FVE.

## V. References

- 1) [http://wiki.ros.org/openni\\_launch](http://wiki.ros.org/openni_launch)
- 2) [http://wiki.ros.org/pcl\\_ros](http://wiki.ros.org/pcl_ros)
- 3) [https://github.com/code-iai/iai\\_kinect2](https://github.com/code-iai/iai_kinect2)
- 4) <https://github.com/OpenKinect/libfreenect2>