3/31/2016

# Progress Review 11

Individual Lab Report #10

**Abhishek Bhatia**

**Team D:** Team HARP (Human Assistive Robotic Picker)
**Teammates:** Alex Brinkman, Feroze Naina, Lekha Mohan, Rick Shanor

## I. Individual Progress

For this week’s progress review, our major milestones were to get autonomous single bin-test up and running with great accuracy, develop turn-table setup for training dataset generation, integrate online grasping model, and develop offline test infrastructure for perception subsystem. Besides, our focus this week was also on fault testing for different subsystems and finding approaches to mitigate and successfully exit out of these fault scenario’s. The first thing that I started working on was to generate the testing infrastructure for the perception subsystem. We had developed a segmentation algorithm to segment out objects into different clusters using a region growing segmentation algorithm. Now, we had to modify the algorithm to determine the grasp points and grasp orientations to grasp objects one cluster at a time. But verifying various approaches of clustering and grasp point generation needed an offline model. For this, I wrote a simple ros client-server model, the client loads the saved point clouds (test scenarios), converts the point cloud into a ros message type and sends the request to the server. The server, receives the ros msg, concerts it back into the point cloud, does the segmentation to generate clusters and finally generates the grasp pointers for each cluster. The first version of the grasp pointer generation algorithm naively detected the maximum horizontal surface area corresponding to each cluster, selected the cluster with maximum surface area and returns the centroid of this selected horizontal surface as the grasp point. The orientation initially in this naïve approach was kept constant.

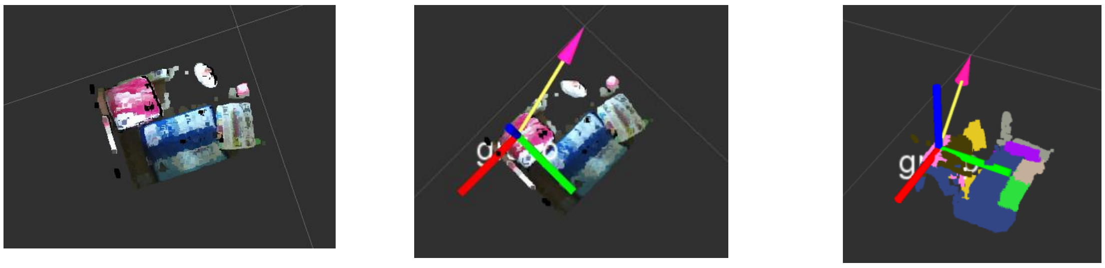
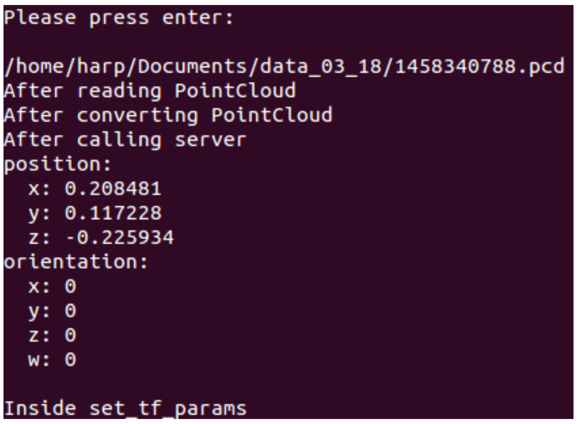


**Figure 1: Image on the left shows the scene of the bin (clutter), image on the right shows the segmentation of various objects into different clusters and grasp pointer generation using the horizontal surface area approach, and image in the middle demonstrates the final grasp pointer output on the original scene**

After this, I modified the client to load in point clouds from a particular directory based on command line input and pass it to server to generate the output. Once the client receives the grasp pointer it publishes the tf of this grasp pointer with respect to the world (as shown above) and waits for another command line prompt to load another point cloud from the dataset. This helped us debug our algorithm and tune the parameters in a much more efficient manner. Figure 2 below shows the command line prompt from the client.



```
Please press enter:

/home/harp/Documents/data_03_18/1458340788.pcd
After reading PointCloud
After converting PointCloud
After calling server
position:
  x: 0.208481
  y: 0.117228
  z: -0.225934
orientation:
  x: 0
  y: 0
  z: 0
  w: 0

Inside set_tf_params
```

**Figure 2: Image shows the command line prompt, user should press enter for the client to load point cloud, pass data to server and prints the received output**

But there were some glitches in this approach, like if the base of the bin was not segmented out properly from the scene, it used to show up as a cluster and generated a bad grasp pointer. Figure 3 below discusses this scenario.

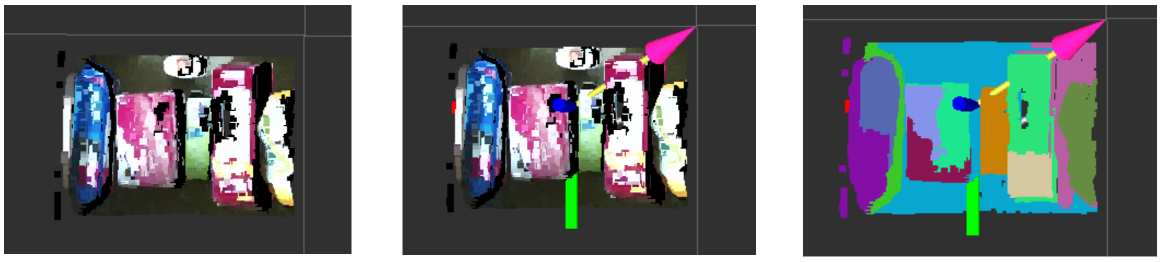


**Figure 3: Image on the left shows the scene of the bin (clutter), image on the right shows the segmentation of various objects into different clusters, the blue cluster corresponding to the base of the bin and bad grasp pointer generation using the horizontal surface area approach, and image in the middle demonstrates the final grasp pointer output on the original scene**

To fix this error scenario, I fine-tuned the Kinect parameters to filter out the points below a certain threshold depth, which seemed to work fine afterwards. Another scenario which was causing us trouble was if there were objects with more height than certain objects, it made sense to grasp them first, even if they had less horizontal surface area. Hence, I modified the algorithm to also take the maximum height of each cluster into account along with the horizontal surface area and come up with a score and generate the best grasp pointer for the object with maximum score. This approach gave us good accuracy, the only thing remaining was that this approach worked really well for objects with flat surfaces, as we were keeping the orientations of each cluster constant. To tackle clusters with non-flat surfaces, we decided to generate normal corresponding to the grasp pointers generated and change the orientation of the gripper to grasp the object perpendicular to the surface of interest. Rick mostly worked on modifying the algorithm to incorporate the normal and I worked on integrating everything back into our state machine, once our algorithm was tested offline.

Besides this, I also worked on many miscellaneous tasks like extensive fault mode testing of the single bin-test, understand various failure modes and deciding on risk mitigation for these [1]. Modifying the state machine to account for these failures and automatically recover. Assembling the Kiva-pod that we received recently and finally testing out the single-bin test with new objects (that also we received recently).

## II. Challenges

Surprisingly, this progress review didn't give me grave challenges that generally take huge time and effort to solve. I faced some minor trouble while writing the ros client-server routine, which I resolved with Rick's help. Later, I faced some issues with publishing the transforms to display the grasp pointer in rviz, but this was also fixed instantly with some help from the internet. I hope to have more such productive weeks till the Amazon Picking Challenge to really spend time and energy on issues that matter.

## III. Teamwork

For this week's progress review, we worked to improve the accuracy on our single bin test, developed a rotating table approach to generate huge image database required to train the convolutional neural network and transitioned from our temporary cardboard shelf to the Kiva-pod.

**Alex:** Alex primarily on generating and fixing the failure cases associated with the suction and planning subsystem. He also worked to fixi the issue with fluctuating pressure sensor values.

**Feroze:** Feroze worked on integrating the online grasp planner approach as part of our picking pipeline and tested various scenarios.

**Lekha:** Lekha worked with Rick to setup the CNN pipeline on Caffe.

**Rick:** Rick primarily worked on setting up the new CNN based vision pipeline. He also spent a lot of time trying to setup PERCH and also get venkat up to speed wit apc dataset so that he can also tune PERCH to work for apc dataset offline.

**Abhishek:** I primarily worked on setting up the tools for perception pipeline offline testing. Besides, I worked with Alex towards system integration and testing.

## IV. Future Plans

My major targets for the next Performance Review are to continue working on perception related tasks, provide Rick/Lekha with automation tools to streamline the process of training the CNN and generate the object identification framework as soon as possible. Besides, I also plan to help Rick in setting up PERCH and test it extensively on apc dataset (multi item in a bin scenarios).

## IV. References

1) Grasping Demo: https://www.youtube.com/watch?v=gy7EnvjXrrg&feature=youtu.be