Lekha Walajapet Mohan

Team D: HARP

Teammates: Alex Brinkman, Rick Shanor,Abhishek Bhatia,
Feroze Naina

ILR07

Feb. 10, 2016

I. Individual Progress:

For Progress Review 8, I was working on devising algorithms for estimation of grasping surfaces. Brain storming ideas with my team gave me a good insight of how to approach grasping. Myself and Alex, we decided to work on grasping together. Given the several approaches for estimating the grasping surface, it is vital to retain the global information on a point cloud. By global information , we mean edges, curvatures, smoothness of a surface etc., Initially, we tried to estimate the grasping surface by determining the center of mass of a point cloud. This is was a naïve method, yet, gave a head start on deciding the grasping surface and plane for rectangular objects. This would not work for cylindrical shapes and complex shapes. The next method, we tried was region growing.

Region growing is a point cloud clustering algorithm based on smoothness constraint. The algorithm is discussed as below:

- Every point in a point cloud is looped over to compare its smoothness constraint with its neighboring points.
- Start by choosing an arbitrary ***seed pixel*** and compare it with neighboring pixels.
- Similar points having similar smoothness constraints are clustered together or grown from the seed pixel(selected point) by adding the similar neighboring pixels
- When the growth of one region stops we simply choose another seed pixel which does not yet belong to any region and start again.
- This whole process is continued until all pixels belong to some region.
- A ***bottom up*** method.

The region growth has certain limitations. The region dominates the growth process which might cluster two different edges together. This would result in clustering of two different objects together as one object. Also, we users set the parameters for seed regions and there need not be a global set of parameters that would be applicable to entire dataset that we have. This will again reduce the accuracy of the region clustering. Different choices of seeds may give different segmentation results.

The next method I tried to implement was cylindrical segmentation. As most of the objects in the given dataset could be resolved into primitive shapes of cube and cylinder, I estimated the grasping surface of the point cloud data with cylindrical segmentation. Once the point cloud is read, the points are downsampled to estimate the normals from the downsampled point cloud. By modelling the planar co-efficients, the planar inliers can be obtained from the point cloud. This planar inliers are the cylindrical segment obtained from the point cloud[Fig 1a, 1b]. The cylindrical segmentation results displayed below. Show figure does not have a good estimate of grasping surface, especially around the edges. It would be risky for the vacuum suction gripper to grasp from the edges. The basic decomposition algorithm of point clouds into primitive shapes is the ideal algorithm that has been used by the general robotics community.

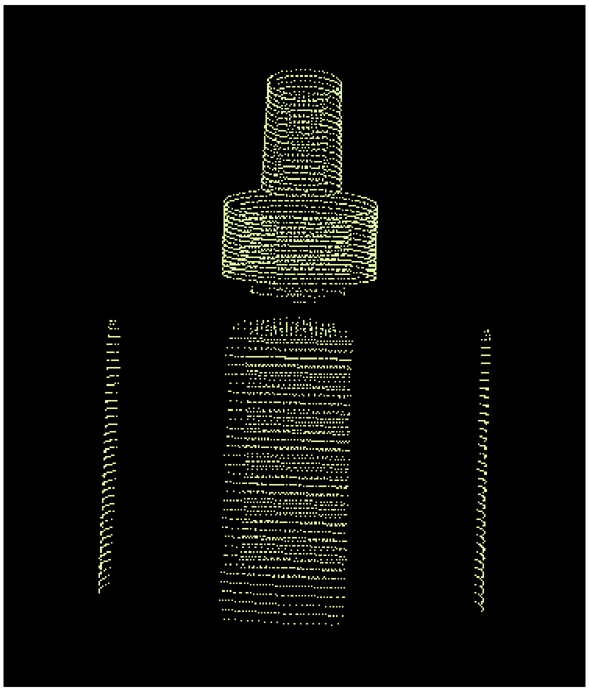
Fig 1a
Cylindrical segmentation

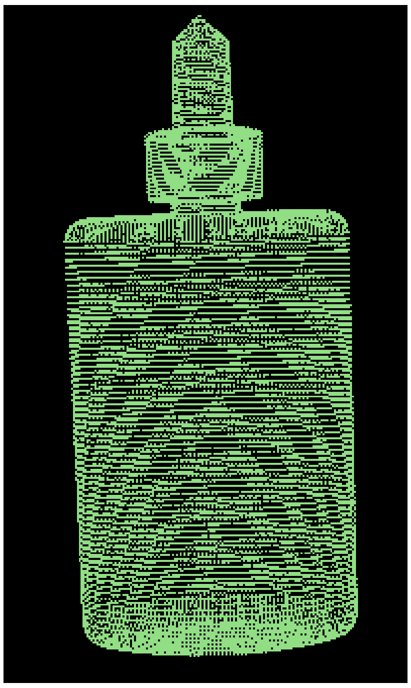
Fig 1b
Complete point cloud data

II **Challenges**

Biggest challenge that I faced for this week was to design a global solution for estimating the grasping surface. Our technical advisor suggested that we generate an offline database for grasping surfaces, where we manually mark the grasping surface, upload it in the cloud. While testing, we can generate online(real-time) cloud data points, compare with the generated data base and then estimate the resultant grasping surface over the point cloud data.

This would be a huge challenge as we don't have the arm yet and we are unable to guess the behavior of the suction cup coupled with the arm. Although simulation estimates a great grasping surface position, the real time robot might have some latency. Also, since the offline generation of database is subject to human judging, we might be wrong in estimating surface or might global surface estimation. The dataset is yet to be released by Amazon Picking Challenge officials, which is another big challenge. If the shapes or sizes are too complicated, we would have to design an algorithm in lesser duration of avialable time.

III **Team Work**

Alex Brinkman was working on online generation of grasping points from the normals of the point cloud data. He generated a simulation that demonstrates the online grasp surface estimation(Fig 2). Alex is also closely in touch with Universal Robots officials to ensure faster arrival of UR5 arm, so that we can start our testing. Rick Shanor was working on PERCH(Perception and seaRCH) algorithm provided by the Search Based Planning Laboratory. Rick Shanor along with Feroze Naina was closely involved in generating the video for Amazon Picking Challenge video. As the Search Based Planning Laboratory uses its own dependencies, Rick had problems running it on his machine. Our technical advisor is working closely with us to get it working. Abhishek Bhatia was working on Object

Recognition Kitchen, a package exclusively for modeling objects in mesh. Abhishek also helped Rick Shanor in setting up PERCH algorithm in the machine.

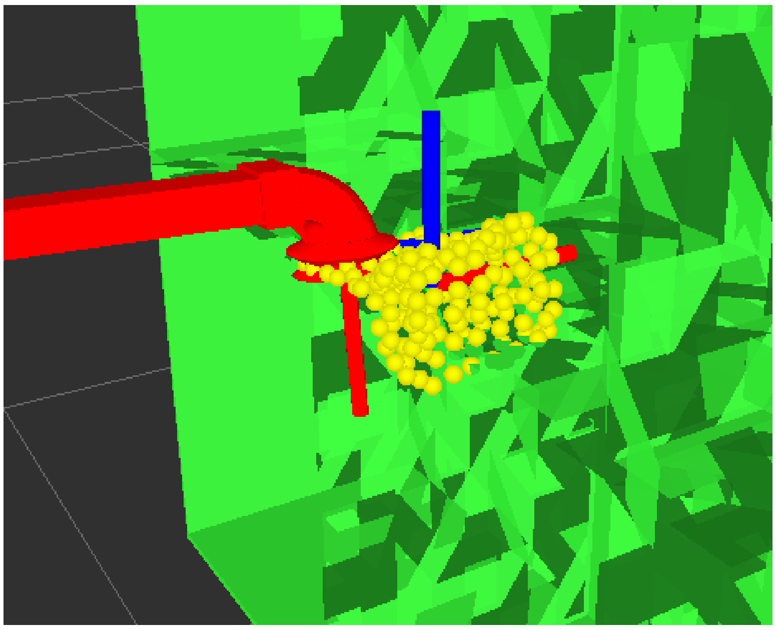
Figure 2 – Estimation of grasping surface in simulation
Picture Courtesy: Alex Brinkman

IV Future Plans

I will be working on generating offline database for the grasping surface. I will also be involved in generating the ground truth model once the items are released. In future, I will also be involved in updating the changes in website reflecting the recent changes along with Alex. Alex will be involved in refining the online grasping surface generation. Once the UR5 arm arrives, he will be prototyping the camera mount and suction gripper mount for the UR5. Rick will be involved in installing, testing the PERCH algorithm including cases for occlusions and cluttered environments. Abhisek will be involved in testing various point clouds, with their ground truth model using Object Recognition Kitchen for object detection. He will also be drawing comparison between the performance of the above mentioned package, our perception pipeline and PERCH algorithm by SBPL.