Lekha Walajapet Mohan

Team D: HARP

Teammates: Alex Brinkman, Rick Shanor,Abhishek Bhatia,

Feroze Naina

ILR09

Mar. 16, 2016

I. Individual Progress:

For Progress Review 10, I was working on devising the grasping strategies for the APC list of item dictionary(2015) and implementing deep learning strategies for our project. For offline grasp generation, we manually define grasp points on a point cloud. The offline grasping model looks reasonable, yet it is not accurate enough. This is due to the manual error made by human while annotating the grasping points. Yet, this model is good enough to get our system working and testing for integration. Due to the inaccuracies, we are debating on using offline models for grasping any further. Although online model grasping gives us good estimate of the surface normals, relying on these models might increase our risk of failure in grasping the object successfully due to accumulation of errors.

Perception subsystem gives us an accuracy of not more than 60%. This again puts us far behind the competitive edge. RGB-D segmentation seems to work well only for the single items. As the Amazon Picking Challenge has included partial occlusion upto 10 items per bin, totaling to around 40 items in total, the above mentioned method doesn’t well for occlusion. As the APC 2016 item dictionary is out, we understood that objects that have good reflective surfaces have been included. Objects with high luminance are prone to fail when it comes to RGB-Segmentation.

Interestingly, we have decided to go with the machine learning approaches for the perception algorithm. This is an interesting turn for me in this project at this point of time. I will be involved in implementing deep learning algorithms training our net work to output the right detected item. I had discussed with my team about the recent deep learning concepts I had learned from my elective and its potential to answer the challenges at perception subsystem. The trends and papers discussed in the class seemed promising and insightful but we suffer setbacks which has been discussed further below.
Our Deep Learning pipeline goes like this:

- Capture around 2000 images (around 7 hours at 5 images per minute) of cluttered shelves
- Train a CNN using the SegNet structure
- Generate a script that applies mask to the depth cloud , applies RGB randomly, distorts the image on a minute scale
- Create train/test list of modified images
- Pass individual segments to CNN, which outputs probability vector of each item

We already have the turntable setup which has the Kinect camera to constantly capture images. Currently we are capturing images at 7.5 degrees of the entire object to collect the image dataset. Linear actuator actuates the turn table set up which we will rotate the rig for the camera to capture the images. The captured images contain background of the setup, which we will be filtered out(Fig 1a). Later, using 3D and 2D information we will cluster the image(Fig 1b), which will give us a probability output vector of item.

We are going to distort the images so that our architecture is robust to lighting variations at the competition. The network we will be implementing is the SegNet architecture. SegNet. As per definition, this core trainable segmentation engine consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network . The role of the decoder network is to map the low resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature map(s). Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps. We compare our proposed architecture with the fully convolutional network (FCN) architecture and its variants

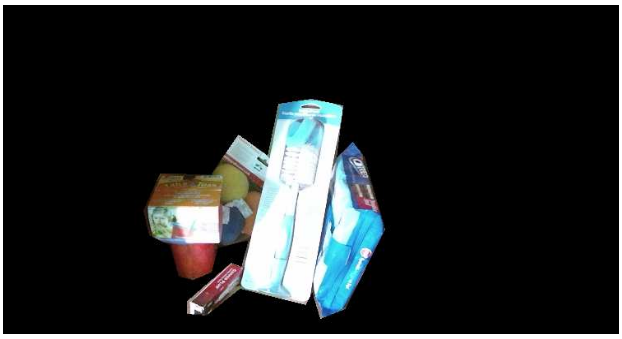

Fig 1a – Filtered out shelf items

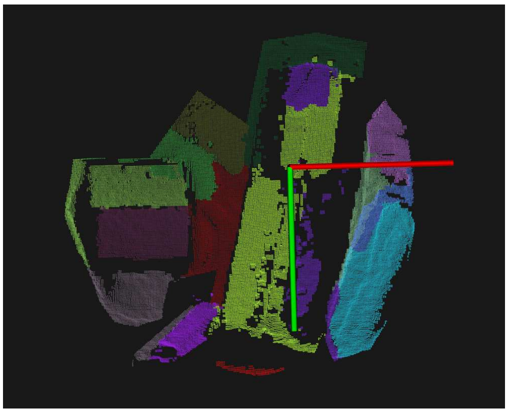

Fig 1b Segmented image cluster

II **Challenges**

Biggest challenge that I faced for this week was to develop a robust methodology for the perception. I am new to deep learning and do not have a formal training on machine learning. This hinders my chances of putting forward an optimal solution to perception. I am doing my literature survey, yet, research in this field is limited for cases that do not use huge data set. This will be my biggest challenge is cracking the challenges posed by the perception subsystem

III **Team Work**

As we have received our new UR5 arm, we have started testing for single item integration test. Alex Brinkman was working on extrinsic calibration for the UR5 arm. He was also involved in building the turn table rig for generating my training set. Rick and Abhishek were working on fine tuning detection of objects using RGB-D segmentation. Rick was also working on training a very basic network using deep learning to get an intuition of how to proceed further based on the achieved results

IV Future Plans

I will be working on training deep networks for object segmentation and pose estimation. As I am still doing my literature survey on robust algorithm, my plans might change in future. Rick will also work on perception, delving deeper into neural networks. Rick will be involved in improving the PERCH algorithm. Alex will be rigorously working on fault mode testing analysis, testing various possible failure cases. Abhisek will be involved in eye-in-hand coordination package implementation. Feroze will be involved in generating the grasping models and executing its pipeline.