



3/17/2016

Progress Review 10

Individual Lab Report #9



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 integrate everything with the actual UR5 hardware and setup the alternate machine learning based vision pipeline. The first thing that I started working on was to generate a masking script in python to read a png image (Figure 1) and a Json/XML file that contains the information about the pixel locations bounding the objects and the particular shelf bin. The idea behind this was to read the png image and generate two png images, one containing the shelf bin and the objects (Figure 2) and one just the objects (Figure 3). We will use these masked/segmented images to generate our training dataset with which we plan to train our convolutional neural network. The Convolutional Neural Network will further train on this dataset and segment out foreground/background on test images.

To generate the mask, I wrote the script with 2 functions, the first one being the parser. The parser was a simple function to read a JSON file which had one data object for the shelf and multiple data objects for the items in the scene and store them as 'numpy' arrays. Once I was able to read the file and positions of bounding boxes around the shelf and the objects, I wrote another function to convert every pixel outside of the bounding box which contained interesting pixels as black. I used functions from opencv to develop this masking function. Below are the results from the masking script.



Figure 1: Image displaying the unsegmented shelf scene



Figure 2: Image displaying the shelf scene after segmenting out the shelf of interest

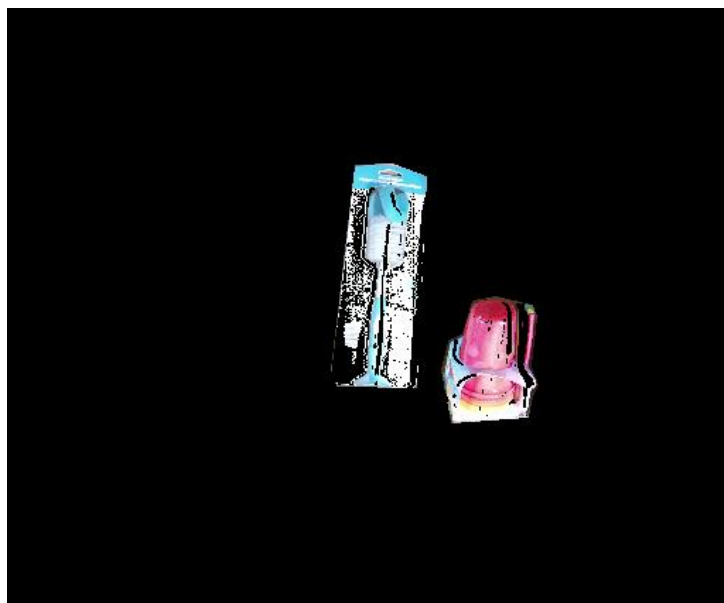


Figure 3: Image displaying the shelf scene after segmenting out just the objects within the shelf of interest

Once I had developed the masking script, I spent some time researching about the existing CNN based object segmentation and identification techniques. Rick forwarded me some of the interesting research papers in this domain like, Object Detection by Labelling Superpixels [1] and ImageNet Classification with Deep Convolutional Neural Networks [2]. Besides, we also discovered an interesting paper on Probabilistic Multi-Class Segmentation for the Amazon Picking Challenge [3]. This was the approach that the APC-2015 winning team, Team RBO-Berlin had followed. It was a simple approach to generate a dataset of training images

by generating 6-DOF feature vector corresponding to each pixel in the image. This feature vector consisted of the 'rgb' color, depth, edges and other interesting information about each pixel, that could be used for segmentation and identification. Once the training dataset is generated, a likelihood for each pixel within the test image to be amongst either the shelf or one of the objects is generated. Later these pixels are clustered together to segment out objects and shelf from the scene. This approach was interesting as it carried out segmentation and identification together and also gave approximately 80% accuracy, which eventually made Team RBO-Berlin the strongest. Another interesting approach that we figured out from this paper was that they used the missing depth data from transparent/translucent objects as a segmentation/identification feature. We are currently debating on how good a similar approach might work on 10-item bins with partial occlusion and also, if we should spend time to develop a similar approach as a backup.

Besides this, I also worked with Alex to integrate all the individual subsystems with the UR5 hardware and testing the single item in a bin configuration's. We had to do a lot of tweaking to get this up. This involved re-calibrating the Kinect camera using the Kinect_calibration ros package, manually calculating the error between the goal position in simulation and reality by observation, and modifying the executive script to cancel out this error. Once we were done with this, we achieved a ~50% accuracy in autonomously grasping objects off the shelf bin and dropping them back in the box. We also discovered a lot of issue's within the individual subsystems, mostly perception and grasp pointer generation, which I have discussed in the 'Challenges' section in some detail. We are currently working to fix these issues as we prepare for our next Progress Review to extensively carry out fault mode testing for individual subsystems.

II. Challenges

The biggest challenge we faced this week was after integrating and during testing the autonomous system to carry out single item in a bin test. As soon as we ran our first test, we discovered there was a huge error between the goal position in simulation and on real hardware. This was challenging to resolve as this error could have been because of multiple reasons including error in localization, error in Kinect camera calibration, error in grasp planner output and error in perception output. To fix this, we started tackling each of these issues one by one. We started with verifying the localization and discovered it was working fine. There was still minor error but we thought that was primarily because of not perfectly rigid card box shelf, hence we decided to let it be for now, till we acquire the actual Kiva shelf. To fix the Kinect camera calibration error, we manually calibrated the Kinect using the ros package and saw considerable change. As for the grasp pointer generation and perception, we are currently working new techniques separately such as, online grasp pointer generation and CNN based perception pipeline. We were able to resolve the issue with error in goal position by recalibrating the Kinect to achieve around 50% accuracy for single item bin tests.

III. Teamwork

For this week's progress review, we worked to integrate all the individual subsystems together to have our autonomous pipeline ready. Besides this, we worked towards improving the individual subsystems, mostly perception and grasp planning.

Alex: Alex primarily owned the system level integration to get the autonomous single item in a bin test working. Besides, he also worked on tray-table setup for perception database generation.

Feroze: Feroze worked on online grasp planner algorithm and setup.

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 and was able to train a network to generate object segmentation, the results looked promising. He also spent some time doing literature study to figure out better perception methods.

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

IV. Future Plans

My major targets for the next Performance Review are to get the basic stowage task working. The idea is to setup a pipeline on actual hardware to do basic segmentation of the tote and pick up items from the tote and place them on a table. Besides, I will also focus some time on perception related tasks to help Rick get the CNN based pipeline up as soon as possible.

IV. References

- 1) http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Yan_Object_Detection_by_2015_CVPR_paper.pdf
- 2) <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- 3) http://www.robotics.tu-berlin.de/fileadmin/fg170/Publikationen_pdf/Jonschkowski-16-Technical-Report.pdf