Alex Brinkman

Team D: Project HARP (Human Assistive Robotic Picker)

Teammates: Abhishek Bhatia, Lekha Mohan, Feroze Naina, Abhishek Bhatia

ILR #11: Progress Report

Submitted 4/13/2015

1. Individual progress

Since the last progress review, my individual efforts have been focused on supporting image acquisition for training the perception convolutional neural networks (CNN), improving grasping and arm planning to enable multi-item picking, improved the runtime efficiency of the new CNN perception pipeline, and created an approach to grasp specular objects.

The image acquisition is driven by the need to train the identification CNNs for both the picking and stowage tasks. During the last review, I had constructed the software and hardware to perform the item segmentation using a green screen background. The problem is some items are green and cannot be segmented easily from the background and that specular items were reflecting some green hues that will not be present in the training set. I switched to using a brown cardboard colored background to omit the green hue artifacts but this required more advanced segmentation. Using OpenCV, I added a convex hull to the segmented item which surrounds the points in the smallest fitting polygon. This ensured that the segmentation did not omit pixels within the center of the body.

The turntable and filtering took quite a bit longer than I had planned. By the end of it, I had put approximately 25 hours into the construction of the turntable and generating the dataset. Had we not constructed the turntable and manually segmented the images we estimate the amount of work would have been approximately 60 hours of work so it was well worth the effort.

Next, I focused on grasping efforts to better organize the planner code. Now when we want to plan a path we have the following options to choose from that are easily modified in the executive controller:

- Trajectory Playback – replay stored trajectory if it exists
- Cartesian Plan – Fast solver to find straight distance points to goal, executes partial paths
- Cartesian Plan Strict – Fast solver to find straight distance points to goal, only executes complete paths
- Fast Single Query – Performs full path plan around collision objects using OMPL or SBPL planners with a short timeout
- Slow Single Query – Performs full path plan around collision objects using OMPL or SBPL planners with long timeout

All of these features were being used throughout our system and were being offered by several different servers. Bundling this functionality into one easy-to-use server allows us to keep our ever-expanding codebase simpler and executive scripts cleaner.

To get picking to work on the Kiva pod, we needed to extend the end effector by approximately 5 inches. The planners were having difficulty planning with the tight margins required by our original design. I first tried to extend the end effector by 2 inches with little improvement and later added the additional 3 inches. This complicates motions outside of the bin but we are handling that difficulty with precomputed trajectories.

During development, we were experiencing high computational loads from the perception pipeline. The pipeline requires 5 distinct operations to pre-filter, segment, identify, reproject, and post-filter the RGBD data stream. During development, most of these were implemented as nodes that ran continuously for visualization and debugging but continuously running processes could no longer be sustained. I refactored the nodes to run as servers that run only on request, reducing the computation loads from 150% across 2 cores to just 45% of one core. Initially, I tried to pass the pointclouds and images to the servers through the SMACH executive but the system did not respond well. After some online debugging, I learned that all SMACH userdata is shared with the introspection server used to visualize the data and state flow. Given I was passing around numerous pointclouds and images within SMACH, this made sense. I experienced approximately 7 seconds of delay between each of the 5 perception servers. To improve this, I created a python server to pass around the userdata outside SMACH to avoid costly recompile time and work around the SMACH slowdown. In total, the perception pipeline runs in approximately 10 seconds including optional data logging.

Finally, I implemented a work-around for grasping specular objects. The Kinect2 reports distances further than expected since the light reflect off the reflective surface and causes poor depth association. This results in the pointcloud for specular objects appearing behind or below the shelf which is filtered out by our post-filtering operation and can cause segementation faults within our syteme. Sensor fusion techniques can be used to improve the accuracy of specular object depth estimation but the prior art I found could increase accuracy by 30% but would still be unacceptable in our application. Instead, I implemented a ray tracing algorithm that projects the specular points onto the nearest shelf wall along the incident ray to the camera frame. This results in perfectly flat pointcloud of the object of interest along the bottom and sides of the shelf for specular items and works surprisingly well. Initial testing shows this approach will work well with our grasping approach… just in time for the Spring Validation Experiment.

2. Teamwork

Bhatia and Rick have been working through the implementation of the identification CNN. Rick has been doing excellent leading the research and development of the perception system and can demonstrate some amazing results. Feroze, Abhishek, and I have been working non-stop on improving grasping ahead of the demonstrations for National Robotics Week and the SVE. Lekha has been generating models for SBPL's Perch perception approach.

3. Plans

My upcoming plans are focused on increasing the system robustness ahead of the National Robotics Week demonstration and SVE. This list below highlights some of the more critical tasks:

- Add collision checking to the trajectory playback feature
- Autonomously detect and recover UR5 safety disabling events
- Conduct a robustness study to ensure we can handle shelf placement variability (up to 2 inches)
- Help develop sideways grasping primitive for tall objects
- Improve the end-effector design to minimize collisions
- Implement arm motions in parallel with perception to decrease runtime
- Continue to work with project sponsors and promote awareness of the project