2/9/2016

Progress Review 8

Individual Lab Report #7

Abhishek Bhatia

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

I. Individual Progress

For this progress review, our team's major task was to officially register for Amazon picking challenge. This required us to write a proposal explaining our complete system in details along with a video submission that show's our progress till date. Besides this, we were hoping to acquire our robotic platform UR5 before this progress review and were highly optimistic to hack a quick demonstration on the actual robot to be included as part of the video submission, but we couldn't get the platform.

For the video submission, my major responsibilities were to generate high quality images and videos of our working grasping subsystem with Kinect2 mounted on top of it. Since we couldn't get the platform, we decided to individually demonstrate the grasping subsystem by holding the end-effector in our hand and displaying its functionality to grasp different objects [Figure 1].

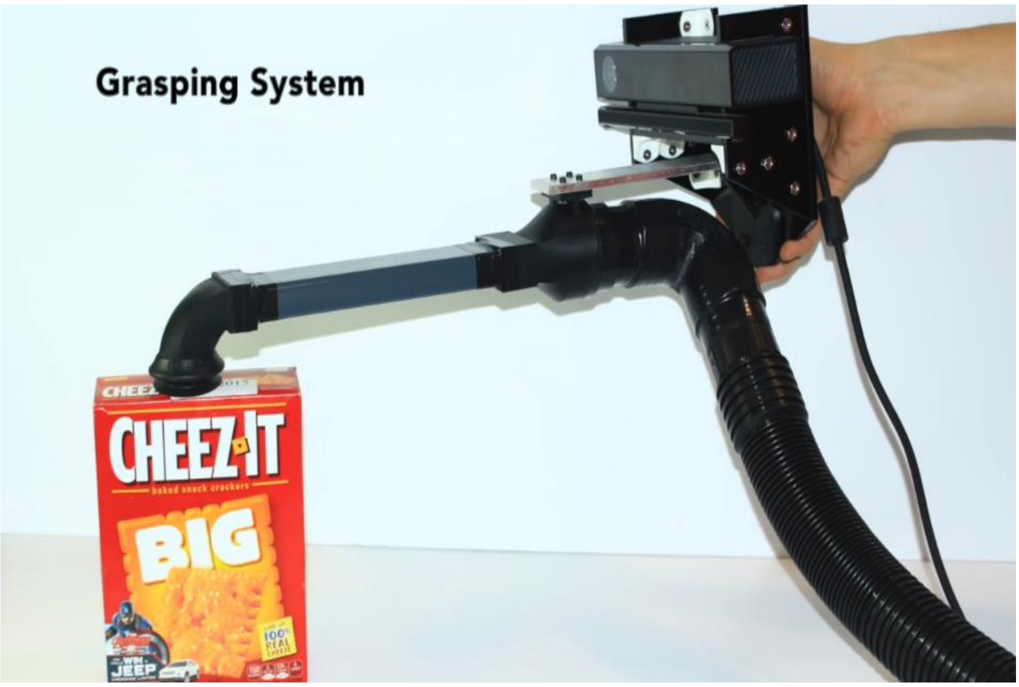

Figure 1: Image displaying the grasping subsystem with custom end-effector and Kinect2 mounted which will be attached to the UR5

Besides this, I worked with Rick to generate a better localization video snippet that displayed our localization around the 9-bin shelf [Figure 2]. Our intent was to demonstrate that by moving the Kinect2 our localized points also move and realign themselves around the shelf. We had several such shorts experiments displayed as part our video submission to demonstrate capabilities of our individual subsystems.

The official APC 2016 submission video by team HARP could be accessed here: https://www.youtube.com/watch?v=oGq05wN7mmg&feature=youtu.be

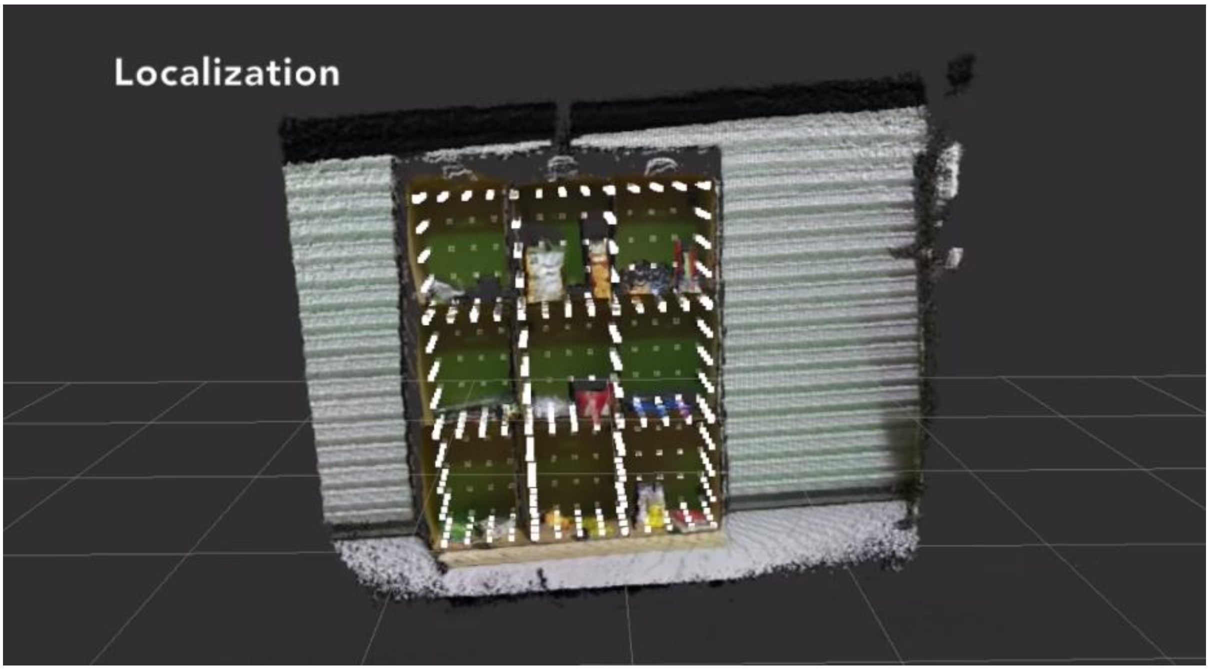

Figure 2: Localization displayed across the 9-bin shelf

After completing the work with setting up the video for APC, I went back to work on my experiment with Object Recognition Kitchen. I followed the tutorial "Object Recognition Using Tabletop [1]" to test the object recognition pipeline to find planes, and identify specific objects. I created a database of 3 objects from APC-2015 item dictionary (pack of oreo, pack of crayola and first year cups). The first part of this experiment was to detect plane surfaces using ORK_tabletop package, which worked well. The pipeline was able to detect various planes including tabletop, wall, and ground beneath using live feed from Kinect1. After this, the second experiment was to find objects. The first thing I tried was with a single object on a plane surface. The ORK_tabletop identified the object correctly but could not identify the pose and orientation of the object correctly. It basically tried to match the model (mesh) of the object from the database to the test object, but could not do so accurately. Further, as soon as I increased the number of objects on the plane surface from one to two and tried to recognize either object, I couldn't get any result. Same was the case with 3 objects. I also tried the same experiment with a different package "Object Recognition using Linemod [2]", but could not gather any recognition. Before trying to debug further into this issue, I thought I would do some research on how many people had benefitted from this off the shelf package. I researched through last year's APC team "plocka_packa" code base on github [3] to figure out what exact functionalities they used from the ORK package, but failed to make much sense of the code base. Finally, some further research made me stumble upon "Johnny Wang's" Individual Project Report from last year [4], where I figured out that Team-I last year also tried to use ORK for object recognition in some part of their development cycle, but couldn't get results because of some issue which they couldn't debug. This is the point where I decided to not spend any further time trying to play with this package. I discussed this with Rick and we concluded that even if we had got some results from these experiments, we had to modify the ORK source code to make it work with Kinect-2. But with these results, it was probably a good point to stop and explore other options.

After this, me and Rick met Venkat and discussed about starting with PERCH [PERCH description in references section] setup and experiment it with our database. I started with setting up PERCH with Venkat's instructions [5]. I faced a lot of issues to get the package build as it had several dependencies that needed to be installed. After installing several packages including 'gsl-bin, libgsl0ldbl, libgsl0-dbg, libgsl0-dev ,libhdf5-serial-dev, and ros-indigo-keyboard' and making several hacks like commenting some dependencies which were not required for PRECH specifically, I got the package to build successfully. Later while trying to run the first example to test PERCH using Venkat's database, I got the below mentioned error:

```
Xserver found. dri2 connection failed!
/dev/dri/card0 not authenticated
/dev/dri/card1 not authenticated
```

After some research I figured out that this error could be related to some issue with video card support for rendering on the lab machine [6]. Digging deep into the issue, I figured out that this could further be caused due to last systems updates happened on the lab machine. I can see that system updates included updated on openCL and beignet packages and I guess these updates caused this issue. Another reason for my belief that these system updates could be causing this error is because, after these updates we have not being able to run 'Kinect_bridge' package on 'gpu'. The same package was running on 'gpu' before this update and now the 'Kinect_bridge' runs only on 'cpu'. I am still trying to fix this issue on the lab machine. Meanwhile, we made the same package updates (required for PERCH) on Rick's laptop and were successfully able to build and run Venkat's example on Rick's laptop. Rick is currently trying to modify that example to use APC database and make PERCH run with it.

II. Challenges

Working towards this progress review was full of challenges, a lot of which are still unresolved. Firstly, I faced some difficulty in setting up Object Recognition Framework for our dataset. But I figured out ORK was not exactly a framework that could help us based on the research that I conducted. Later, I struggled with building PERCH package, a lot of issues I figured out based on support on the internet and Venkat's help.

Finally, my biggest struggle was figuring out and trying to fix the OpenCL and beignet driver issue. It took me a while to figure out that the updates might be the reason behind the 2 issues I am facing with PERCH and 'Kinect_bridge'. Even after I had figured out the issue, I tried a lot of workarounds to fix this, like reinstalling the complete libfreenect2 package (Kinect drivers), but I have been unable to fix this. The only approach in mind currently is a complete system reset including reinstalling Ubuntu 14.04 and all the necessary packages and restricting any updates.

III. Teamwork

For this week's progress review, we worked together to generate material for our official APC registration. Besides this, we worked towards trying out different object identification techniques to improve our perception algorithm.

Alex: Alex worked with Andrew to integrate the SBPL planner to work with UR5, as part of our system. Besides, Alex also worked to generate the complete configuration space of UR5 within and around the shelf. Finally, he worked with Lekha to integrate the grasp planner within our state controller.

Feroze: Feroze worked on generating the video of the localization subsystem. Later, he helped me with trying to resolve the Ubuntu driver issue.

Lekha: Lekha worked on developing the grasp planner and integrated it within our state controller. The grasp planner currently supports cylindrical segmentation, normal estimation, hull generation, clustering, and GUI for grasp planning.

Rick: Rick mostly focussed on compiling the proposal and demo video for official APC submission. He compiled all the individual videos into a final video and also took care of the narration to explain various subsystems.

Abhishek: I worked on generating demo video for the grasping subsystem. Later I worked on getting Object Recognition Kitchen framework work with our database. Finally, I worked on setting up PERCH and try it with our database.

IV. Future Plans

My major targets for the next Performance Review are to fix the Ubuntu driver issue, setup PERCH on the lab machine and start working on Kinect Camera Calibration. Once PERCH is setup on the lab machine and we are able to make it work for one example from our dataset, there are many things that we need to figure out after that. These things include modifications required to make PERCH work for our complete use case, which further include reducing PERCH's complexity and make it specific to our requirement and also exploring some machine learning tricks to do some pre-processing, before final classification is made using PERCH. Besides, we are all eager to setup, integrate and test some initial configurations of our whole system, as soon as we get the UR5 arm.

IV. References

1) http://wg-perception.github.io/ork_tutorials/tutorial02/tutorial.html
2) http://wg-perception.github.io/ork_tutorials/tutorial03/tutorial.html
3) https://github.com/amazon-picking-challenge/plocka_packa

4) https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxtcnNkcHJvamVjdDIwMTQxNXRlYW1pfGd4OjI3ZjYxZTk1NTM3NGU5NjU

5) https://github.com/venkatrn/perception/blob/master/README.md

6) http://stackoverflow.com/questions/31901160/installed-beignet-to-use-opencl-on-intel-but-opencl-programs-only-work-when-run

7) **“PERCH:** PErception via SeaRCH, an algorithm that seeks to find the best explanation of the observed sensor data by hypothesizing possible scenes in a generative fashion. Our contributions are: i) formulating the multi-object recognition and localization task as an optimization problem over the space of hypothesized scenes, ii) exploiting structure in the optimization to cast it as a combinatorial search problem on what we call the Monotone Scene Generation Tree, and iii) leveraging parallelization and recent advances in multi-heuristic search in making combinatorial search tractable.”
Abstract from: **“PERCH: Perception via Search for Multi-Object Recognition and Localization - Venkatraman Narayanan and Maxim Likhachev”**.