# Amazon Picking Challenge
## Team HARP (Team D)

**Abhishek Bhatia, Alex Brinkman, Feroze Naina, Lekha Mohan, Rick Shanor**

## Problem Statement

Rapid growth in the worldwide market for warehouse automation and control systems is being driven by the global boom in e-commerce. Amazon is able to quickly package and ship millions of items to customers from a network of fulfillment centers all over the globe. Amazon sells 306 items per second and has 96 fulfillment centers across the United States. This would not be possible without leveraging cutting-edge advances in technology.

We developed the Human Assistive Robotic Picker (HARP) as an entry to the competition 2016 Amazon Picking Challenge. The goal of HARP is to enhance warehouse automation. HARP is equipped with highly-sophisticated features: **item identification**, **suction-based manipulation, and 6DOF motion planning**. This allows the system to operate in dynamic environments and perform the core functions of item retrieval and item stowage. Team HARP developed the algorithms around the UR5 robotic arm to achieve the pick-and-place warehouse task.

## Use Case


User places his order on amazon.com


Kiva shelves autonomously drive from storage and arrive in front of HARP system.


HARP retrieves the items from the shelves and places them in a box


In less than thirty minutes, the order is out for delivery.

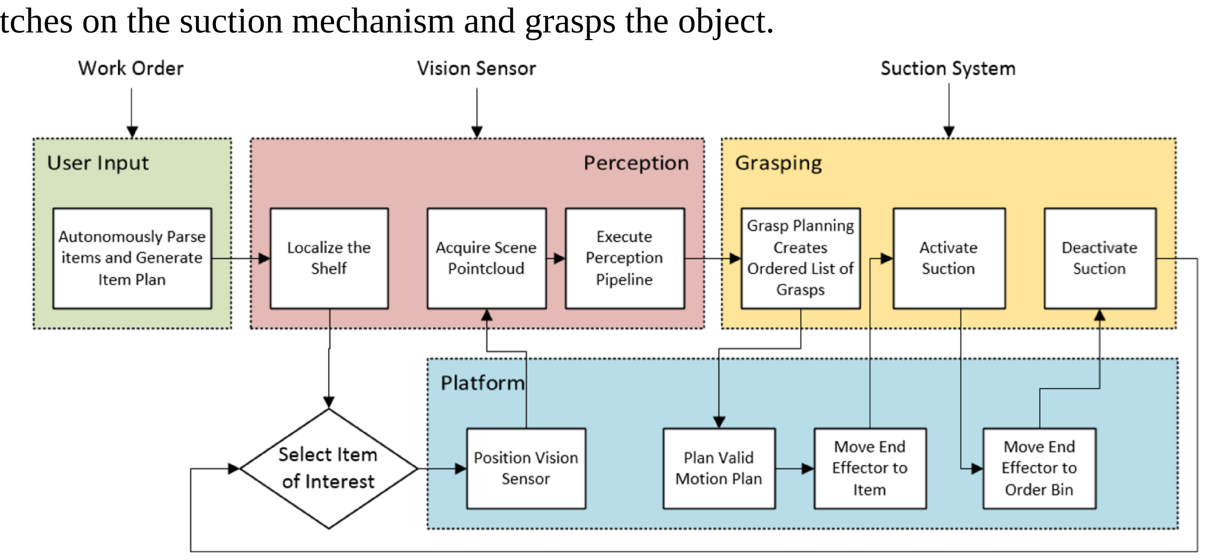
## Functional Architecture

**Input Handling:** The robot autonomously parses the items in the list to generate an item plan.

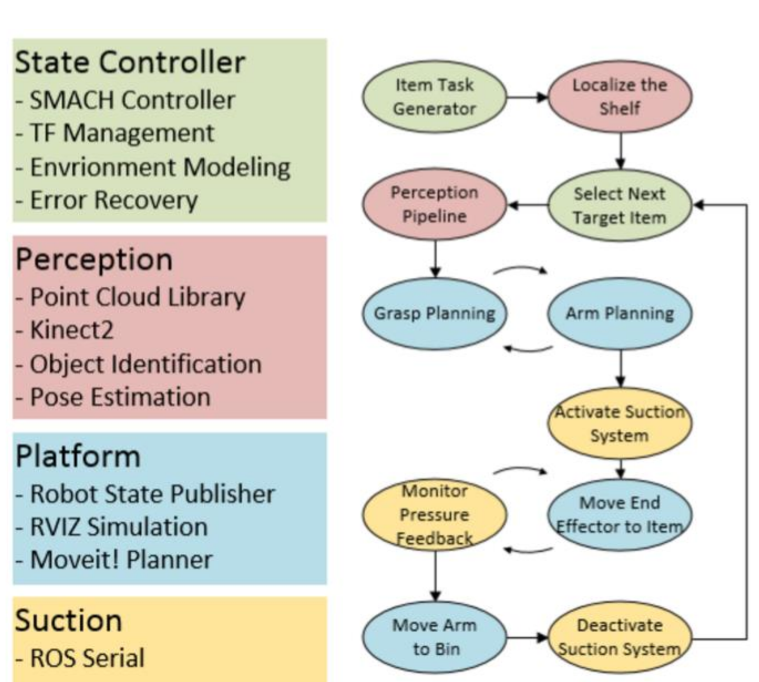
**Perception:** The perception function is responsible for scanning the shelf, scanning individual bins, determining item pose and providing the system with sufficient data to plan the manipulator trajectory to grasp the item from the shelf bin and place it in order bin.

**Platform:** The platform planner takes the item pose data as input and generates a valid collision-free motion plan to move the arm to a valid grasp position.

**Grasping:** The grasping function decides on the best grasp strategy and orients the end effector with respect to the object pose consisting of a suction system. Once the suction arm is close to the object, the grasping function switches on the suction mechanism and grasps the object.




## Software Architecture



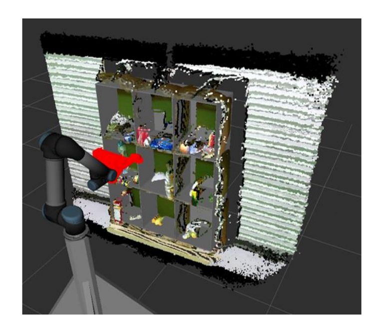

Software Architecture highlights state flow and robot decision making

## System Design


**Amazon Picking Challenge 2016 setup**
**Level – B Newell Simon Hall, Carnegie Mellon University**

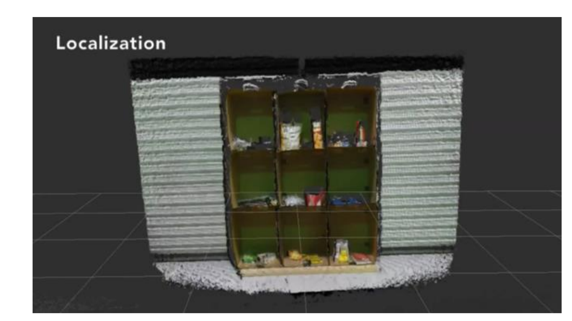
### Simulation and State Control

- SMACH state controller manages robot actions
- Actions simulated and visualized in RVIZ
- RVIZ visualizations help validate robot actions



### Localization

- Images captured from multiple perspectives merged using robot kinematic chain
- ICP Algorithm minimizes error between point cloud and shelf CAD model

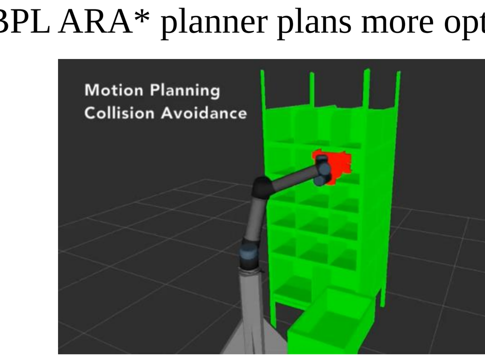



### Grasping Subsystem

- High flow low pressure vacuum system
- Shop-Vac impeller provides 200 CFM /40 kPA
- Custom suction cup mounted to UR5 wrist


### Motion Planning

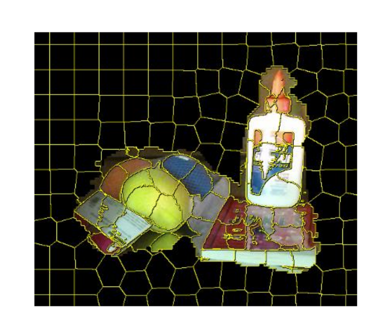
- MoveIt! software package manages arm kinematics and path planning
- OMPL motion planner implements RRT* algorithm to quickly find paths
- SBPL ARA* planner plans more optimal paths

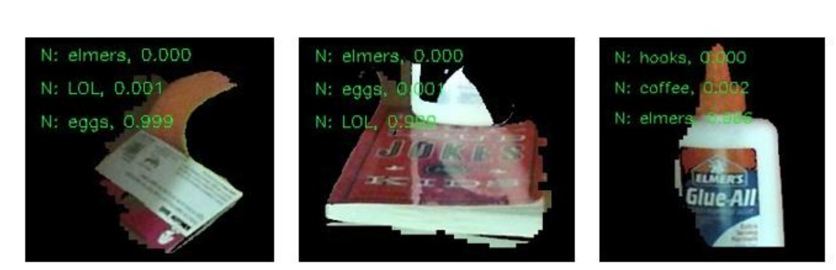


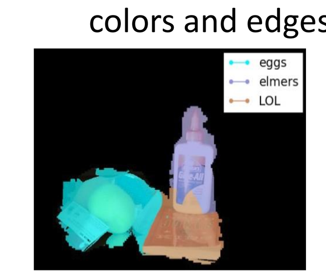

### Vision Processing


**Filtering**: Shelf contents isolated from shelf based on localization results


**Segmentation**: Label each pixel as 'shelf' or 'item' using SegNet CNN

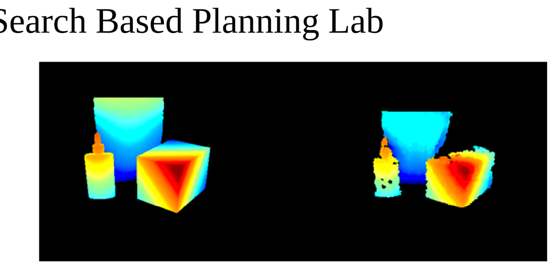

**SLIC**: Divide image into small "superpixel" segments based on colors and edges


**Segment Identification:** Individual segments are classified using a CNN


**Item Identification:** Segment identification outputs are merged to solve for the globally optimal scene

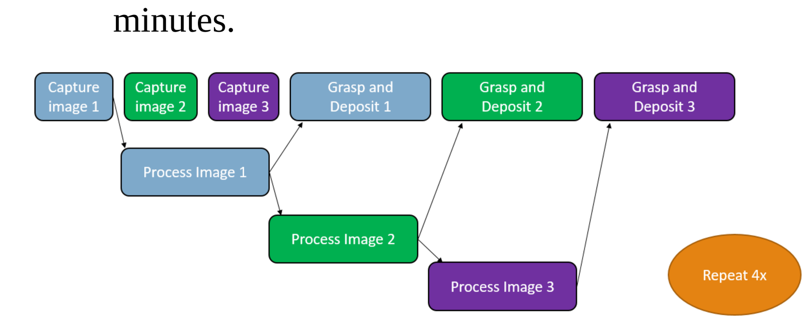
### PERCH

- Geometry based search recognition and location solver developed by Venkat Narayanan and the Search Based Planning Lab



### Parallel Perception

- Improved overall runtime by approximately 2.5 minutes.

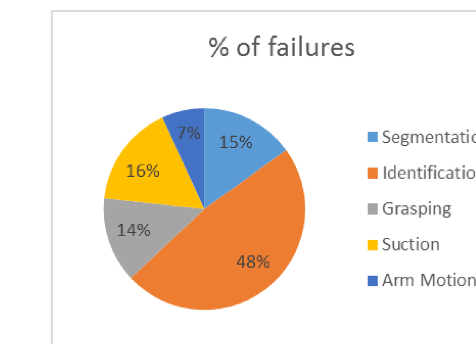



## Test Results

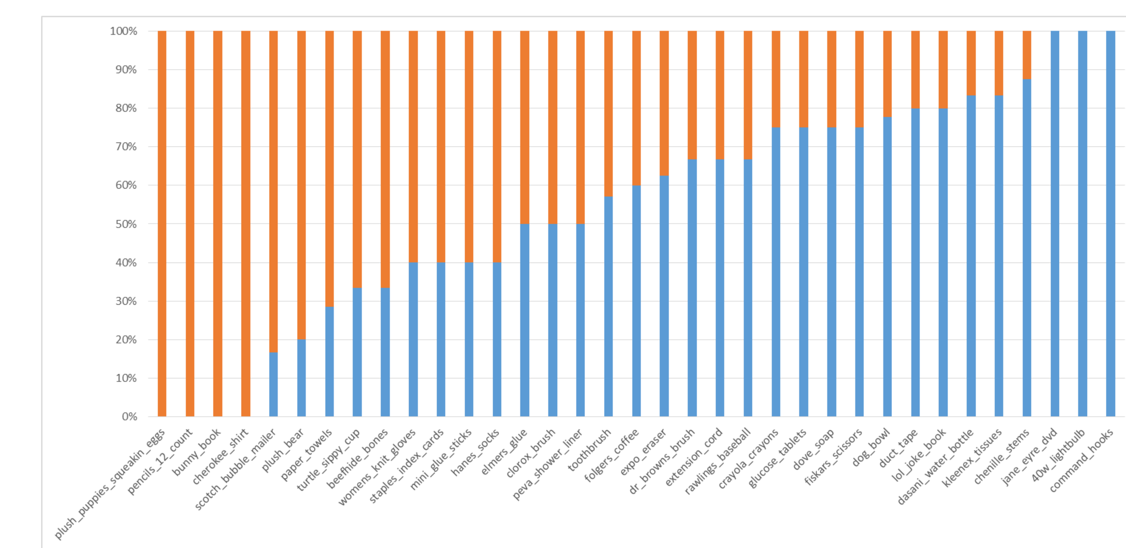
Random input lists (JSON files) were generated that spanned over all the items in the item dictionary. End-to-end test runs were carried out to generate the following statistics. During each test, the Kiva shelf was stocked with 30 items. Each shelf bin contained between 1 and 4 items.

| | |
|---|---|
| **7** | Random Shelf Configuration |
| **1-4** | Items per shelf |
| **244** | Grasp Attempts |
| **147** | Successful Picks |
| **97** | Failed Pick |
| **61%** | Success Rate |

**System Level Test Results**



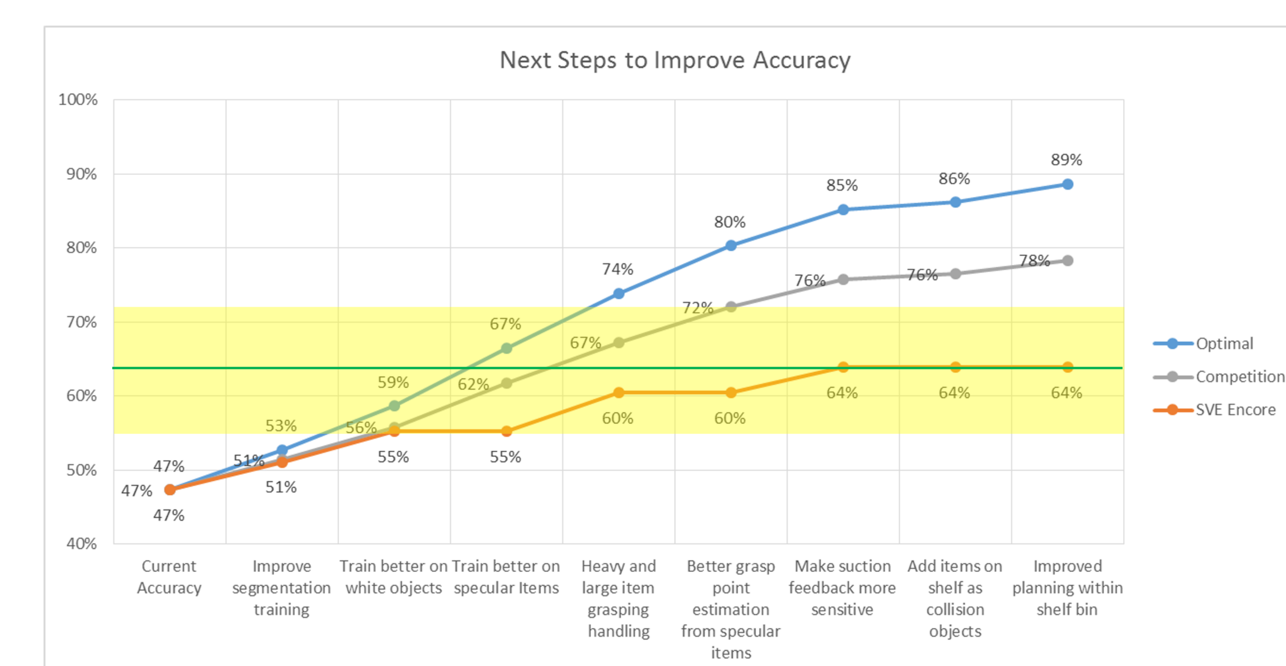

**Failure Mode Analysis**



**Success/Failure distribution corresponding to each item in the dictionary**

## Future Work

There is still a long way to go before Team HARP flies to Germany to compete at the Amazon Picking Challenge in June 2016. The chart below describes the relationship between the current accuracy and the desired accuracy with respect to the functionality yet to be added.




**Desired and actual accuracy improvement with respect to functionality added**

## Acknowledgements

We would like to thank our professors and mentors for their enormous support and contributions towards our success, including but not limited to: John Dolan, Dimi Apostolopoulos, Maxim Likhachev, Hagen Schempf, Venkat Narayanan, and Andrew Dornbush. We are also thankful to Universal Robotics, Nvidia and Search Based Planning Lab for their generous donations that helped us achieve this ambitious task.

## References


[1] Radhakrishna Achanta, et al. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, May 2012.
[2] Alex Krizhevsky, et al. ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
[3] Vijay Badrinarayanan, et al. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, CVPR, 2015
[4] Venkatraman Narayanan and Maxim Likhachev, PERCH: Perception via Search for Multi-Object Recognition and Localization, ICRA, 2016
[5] Maxim Likhachev, et al. ARA*: Anytime A* with Provable Bounds on Sub-Optimality, NIPS 2004
[6] PointCloud Library: http://pointclouds.org/
[7] Open Source Computer Vision: http://opencv.org/
[8] Robot Operating System: http://www.ros.org/
[9] MoveIt!: http://moveit.ros.org/
[10] Open Motion Planning Library: http://ompl.kavrakilab.org/