MRSD Project Course Team D
Human Assistive Robotic Picker
Entry for the 2016 Amazon Picking Challenge

December 17th 2015

Abhishek Bhatia
Alex Brinkman
Feroze Naina
Lekha Mohan
Rick Shanor

Abstract

Amazon has automated their warehouses by using robots to move storage shelves. However, they still require human intervention to pick each object from the shelf bin and place it into the shipping box. Our primary goal is to solve this problem by developing a robot system that can automatically parse a list of items, identify desired items on a shelf, and pick and place them into the order bin. We have partnered with Professor Maxim Likhachev and the Search Based Planning Lab to compete in the 2016 Amazon Picking Challenge.

Our system, the Human Assistive Robotic Picker (HARP), consists of perception, gripping and platform sub-systems. The perception system identifies items of interest based on their known geometric models. The UR5 robot platform, outfitted with a suction gripper, picks up small household objects from the twelve shelf bins. This semester we have validated the individual subsystems by achieving desired perception accuracies and demonstrating UR5 pick-and-place task planning in simulation. Next semester's primary focuses will be on integration and testing.

During the Spring Validation Experiment, the team demonstrated the warehouse pick-and-place task with a 60% success rate, compared to the 33% depicted in the requirements. Failure modes were analyzed; primary failure modes were grasping and perception. This report gives the details of current status and technical analysis of Team Harp's progress toward competing in the 2016 Amazon Picking Challenge. The conclusion highlights remaining tasks and deliverables to complete before the competition.

Contents

Abstract 2
Project Description 4
Use Case 4
System Level Requirements 5
    Functional and Performance Requirements 5
    Nonfunctional Requirements 7
Functional Architecture 8
Cyber Physical Architecture 8
    Physical Architecture 8
    Software Architecture 9
System Level Trade Study 10
System Description and Evaluation 11
        Simulation and State Control 11
        Localization 11
        Motion Planning and Collision Avoidance 12
        Perception: CNN Item Identification 12
        Perception: Dataset Generation 13
        Perception: PERCH Item Identification 14
        Grasping 14
    Modeling, Analysis, and Testing 15
        Simulation Environment 15
        Mechanical Design 16
        Reachability 16
        Suction System Feedback 16
        Perception Testing 17
    SVE Performance Evaluation 17
        Elements 17
        Verification Criterion 17
    Strong and Weak Points 18
        Strong Points 18
        Weak Points 19
Project Management 19
    Schedule 19
    Budget 19
    Risk Management 20
Conclusions 20
    Lessons Learned 21
    Future Work 21
Acknowledgements 22
References 23
Appendix A: 2016 Item Dictionary 24
Appendix B: Item-by-item Failure Analysis 25
Appendix C: Detailed Schedule 26

Project Description

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

John's final project demonstration for the MRSD spring validation experiment is due tomorrow. While running tests, the primary drive motor of his robot burnt out. With limited amount of time and no spares left, John thought he was out of luck. As a last resort, he logged onto amazon.com to check how fast he could receive the spare parts. Fortunately for John, Amazon recently implemented the Human Assistive Robotic Picker (HARP) in its fully autonomous warehouses. The Human Assistive Robotic Picker works alongside the existing Kiva shelving system to fulfill orders round the clock without human supervision.

John places his order on Amazon.com. The order is dispatched to a collection of robots in the warehouse. First, the Kiva shelves autonomously drive from storage to their place in the order queue. This is where HARP comes into play.

HARP performs the task of grabbing items off shelves and putting it in the order bin. HARP can easily handle shelf bins with multiple partially occluded items. First, HARP parses John's order and determines the items of interest – the motor. Next, the vision subsystem computes the position of the requested product on the shelf. Then a robotic arm strategically grabs the item off the shelf using a suction gripper. Finally, HARP places the motor into the order bin. This is then packaged into a box for delivery.

In less than thirty minutes, Johns motor is out for delivery. Hours later, the box arrives on John's doorstep, just in time to impress the MRSD professors before the final demo. The use case is graphically depicted in figure 1.

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.

Figure 1: Use Case Depiction

System Level Requirements

The functional, nonfunctional, and performance requirements are driven by the primary objective of creating a pick-and-place robot to compete in the 2016 Amazon Picking Challenge. In reading through the requirements, it is useful to understand the types of items we hope to pick-and-place. The item list for the 2016 Amazon Picking Challenge is shown in Appendix A. All requirements are identified as mandatory in order to compete in the challenge in accordance with the competition rules and specifications. The objects are of different shapes, sizes and transparency.

Functional and Performance Requirements

The functional requirements were written from analyzing the pick and place task. The performance requirements were produced by analyzing the operation of the top three teams during the competition last year. Our goal is to be competitive with these teams by successfully picking three items off the shelf in twenty minutes. Throughout the design process, the performance requirement metrics have shifted as we have learned more about the technical aspects of this problem. Specifically, accuracy requirements for the perception system have decreased. However, requirements for the grasping subsystem have proportionally increased such that our major functional goals are still met.

FR1	Accept order list from user
PR1	Interpret work order with 100% accuracy
Description	Amazon provides a list of target items which the robot must pick from the shelf. The item is input in JSON format and must be interpreted by the robot.

FR2	Autonomously determine positions and orientations of target items on shelf
PR2	Autonomously identify target object with 75% accuracy
Description	The position and orientation are calculated by the perception module using state-of-the art algorithms. The pose must be determined in order to acquire the objects. Shelf contains up to three items from the item list, non-occluding.

FR3	Accurately determine item grasp position
PR3	Autonomously determine suction grasping surface on 90% of attempts

Description	The perception module outputs position of end-effector for optimal grasping. After correct identification, a grasp surface is determined.
FR4	Autonomously picks item from shelf bin
PR4	Autonomously picks item of known pose from shelf bin on 75% of attempts
Description	After a grasp surface is determined, the kinematics planning is done to pick up the items from the shelf.

FR5	Autonomously places item in order bin.
PR5	Autonomously places 90% of picked item in order bin from a height of no more than .3 meters
Description	Once the item is picked, the robot drops it off it in order bin.

FR6	Must follow the dimensional constraints set by Amazon Picking Challenge
PR6	Acquire items from bins located at a max height of 1.86m and minimum height of .78m Acquire items from a .27m x .27m shelf bin Be able to lift items up to .5kg mass
Description	The items and shelf units specified by the Amazon Picking Challenge rules add constraints to our design.

FR7	Does not drop or damage items during grasping from shelf bin or during transportation to order bin.
Description	During robot operation, the robot should not allow items to fall down. The robot should not deform the items in any way. This ensures we are only adding value.

PR7	Acquire at least 3 items in under 15 minutes
Description	A time constraint of 15 minutes is set in place by Amazon. We must maximize the number of items successfully picked and placed in the given time.

Nonfunctional Requirements

Nonfunctional requirements are driven by both the MRSD course and requirements set forth to compete in the Amazon Picking Challenge rules.

NF1	Cost no more than $4000
NF2	Be completed by May 1st, 2016
Description	MRSD project requirement.

NF3	Transportable or available at ICRA 2016
Description	The robot should be capable of being disassembled and reassembled easily. Alternatively, the robot platform must be available for use at the ICRA competition in Sweden, Stockholm in May 2016.

NF4	Perception robust to lighting between 320-500 lux
Description	The robot's perception system should operate reliably under different lighting conditions and changes in physical geometry. This is because of the possible variations in test environment and competition environment.

NF5	Be available for testing at least 1 day per week
Description	Algorithms must be tested on the real platform every week to ensure consistency with simulation model. This is a desired requirement which aided in the selection of a suitable robotic platform to develop our system around.

NF6	Start and stay within a 2m by 2m boundary (except the end effector)
Description	The competition rules state that the robot should stay within the 2m x 2m work cell and only the end effector can reach into the shelf. The shelf is at least 10cm away from the work cell area.

NF7	Have an emergency stop
Description	The Amazon Picking Challenge requires a stop button to halt the manipulator platform in case of accidents. This is a safety requirement.

Functional Architecture

The functional architecture, shown in figure 2, is broken down into 4 main subsystems.

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.

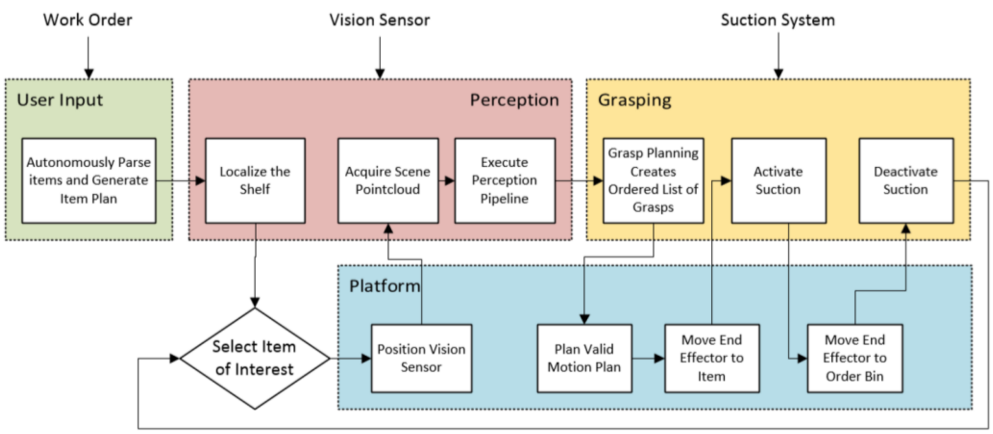


Figure 2: Functional Architecture

Cyber Physical Architecture

The cyber-physical architecture breaks down the system into physical and software architectures. The physical architecture describes the interaction of the hardware components and the software architecture describes the actual flow of data and synergy between different subsystems.

Physical Architecture

The physical architecture diagram (figure 3) shows the interaction of Universal Robots UR5 robot arm with perception and suction subsystems and various components. For planning and perception, we have a Quad-Core i7 Processor based processing unit that runs on Ubuntu 14.04 and ROS Indigo and takes care of the perception subsystem and state controller including UR5's arm planning. The Gripper subsystem consists of the suction mechanism which is controlled by the ROS state controller through an Arduino microcontroller and relays. The pressure sensor interacts with ROS state controller through the Arduino, using the ROSserial protocol and provides the grasp status for the current item.

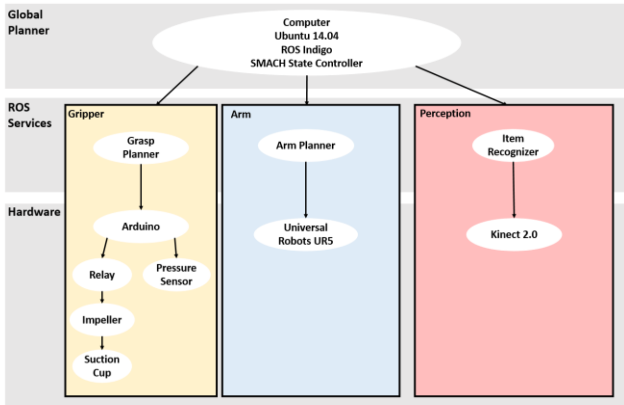


Figure 3: Physical Architecture

Software Architecture

The software architecture diagram below explains the control and feedback mechanisms necessary in order to achieve the functional architecture. The user input, in the form of a text file, is given to the master ROS controller, which begins the SMACH based state machine. In the 'input-handling' state, the master ROS controller passes in an item of interest, which also includes the specific bin number. The arm path planning is done using the MoveIt package. MoveIt uses the OMPL family of algorithms for planning. Using the arm planner, the robot aligns the Kinect2 to the specified bin and captures data. This raw data is passed back (over USB) to the workstation where vision processing happens. The item recognition algorithms then determine the item's position on the shelf.

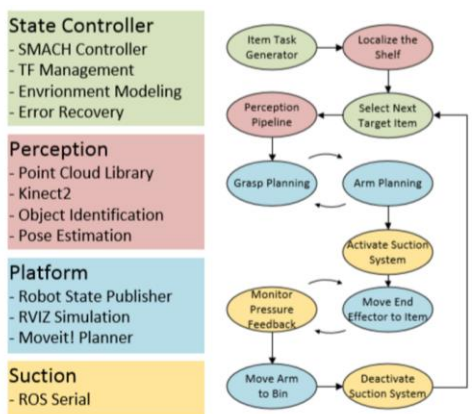


Figure 4: Software Architecture

Using this image data, we determine the grasp surface of the items using the grasp planner. The grasp planner outputs best grasp poses. This results in a position and orientation of the end-effector with respect to the base. These coordinates are passed back into the ROS state controller.

This desired position is passed into the MoveIt arm planner. The arm planner creates a series of actuator commands that are required to position the arm relative to the item. Collision detection checks are performed to ensure that we will not intersect with the shelf. Position feedback, supplied by encoders and other sensors, verifies that the trajectories are executed properly. Once this occurs, the final position of the arm is sent to the ROS state controller.

Finally, arm and grasp positions are sent to the planner. Using trained methods of item acquisition, unique to each item, a grasping plan is generated using IKFast Cartesian planner. A microcontroller is responsible for low level commands for controlling the suction. Tactile feedback (from a pressure sensor), indicates successful grasp. Once the item is acquired, the ROS controller receives a grasp success signal from the grasp controller. The motion and arm planner repeats, moving the item from the shelf bin to the order bin. This cycle repeats until all items from the input text file have been acquired.

System Level Trade Study

The primary trade study performed over the last semester was comparing the UR5 to the PR2. Initially, we planned to use the PR2 for the pick-and-place task. However, this system has several drawbacks, outlined below. Primarily, the PR2 was overly complicated, slowing development. In addition, one major programmatic risk was availability of the PR2 at the Amazon Picking Challenge. By switching to the UR5, we decreased the scope of the necessary technology without cutting the scope of requirements.

UR5

\+ 850mm radius workspace
\+ Easy to use teleop interface
\+ Consistent viewing angle with Kinect
\+ More accurate localization / collision object generation
\+ Fast
\+ 5 kg payload capacity
\+ Transportable (could use same robot for testing and competition)
\+ Good ROS support
\- Inexperience using UR5
\- Must return after competition

PR2

\+ Possibility of 2 arm manipulation
\+ SBPL knowledgeable user base
\+ Already developed baseline performance / set up simulation
\- Can’t transport to competition
\- More complex planning (base, spine, neck, arms)
\- Requires base movement
\- Relatively slow
\- Requires external mounted Kinect or 2 Kinects (sternum and head mount)
\- Modifications may not be desired on SBPL PR2 or possible on the provided PR2
\- Requires booster platform
\- ROS groovy and 2 computer requirements
\- Risk of Hardware or Software Version compatibility issues with the PR2 provided @ Robocup

System Description and Evaluation

The following section describes the technology developed to perform the pick-and-place task. A stationary robot arm with an eye-in-hand Kinect and suction end-effector was the hardware platform selected to fulfil the functionality and requirements outlined above. Main subsystems include high level planning and state control, shelf localization, motion planning and collision avoidance, and perception. Figure 5 shows the final robot configuration and test setup.

Figure 5: Team HARP Test Setup

Simulation and State Control

All software developed for HARP runs in the Robot Operating System (ROS). A SMACH state controller manages the high level states of the robot. The state controller has a similar state flow as the functional architecture. The state controller progresses through states by calling various ROS services. These ROS services perform actions like moving the arm, enabling suction, or capturing a Kinect frame.

Localization

After startup, a localization algorithm (figure 6) runs to determine the location of the shelf with respect to the world. The robot moves to a predefined position approximately 1 meter away from the shelf. A single Kinect depth cloud is captured. Then, the shelf CAD model, in the form of an STL, is loaded into the scene, approximately aligned with where the shelf is expected to be. From there, an iterative closest point algorithm runs. This algorithm iteratively minimizes the least squares error between the CAD model and the input point cloud. This algorithm solves for the optimal transformation of the shelf in world frame. Localization is critical for later avoiding collisions with the shelf.

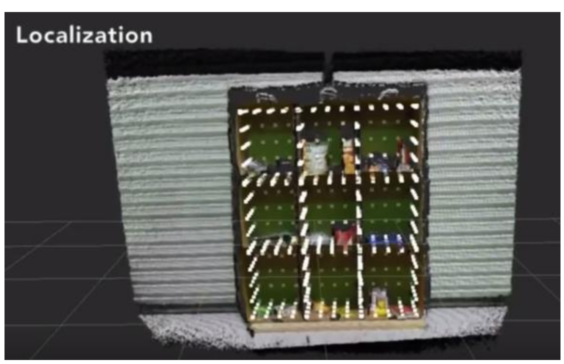


Figure 6: Localizing the Shelf (White Point Cloud) to the Kinect Image

Motion Planning and Collision Avoidance

Arm planning is implemented through ROS and Moveit! using Open Motion Planning Library planners (figure 7). A URDF of the robotic arm is provided by the manufacturer and modified to include our custom end effector. The default planning group starts at the base mounting point and extends through all 6 joints to the tool attachment mount. Our planning group is extended to include the end effector to enable target poses defined in the world coordinates for suction cup locations. Collision models for the mounting frame, order bin, and kiva pod shelf are loaded in using the MoveIt! planning scene interface. Plan details are specified using Moveit! like maximum planning time and which OMPL planner to use.

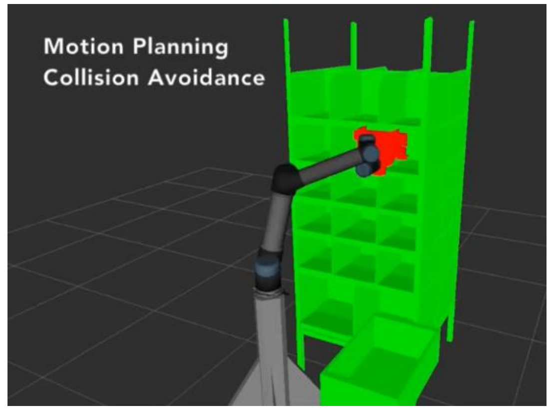


Figure 7: Environment Model and Motion Planning

Each single query plan took approximately 5 seconds to find acceptable paths. Given the competition time constraint, planning between setpoints would take too long and inhibit the effectiveness of the robot. A feature to precompute the path plans was developed and implemented in our system. The trajectory replay feature learns paths when learning mode is enabled. Whenever the trajectory-playback execution order is specified, the feature looks through the learned database for similar start configurations and goal poses and execute the stored trajectory. Otherwise, a Cartesian or single-query plan must be computed.

Our implementation of arm control implements a ROS blocking server that can accept planning details and execution orders. The motivation was to abstract planning as another step in our state controller so that infinite planning loops are not possible. Multiple plans and planning details can be tuned for the needs of each motion plan request. The available execution orders are trajectory-playback, Cartesian plan, strict Cartesian plan, fast single-query, and slow single-query. Trajectory-playback is a feature that and replay precomputed trajectories. The Cartesian plan is a fast planner that uses FastIK to quickly solve the inverse kinematics of the arm at short intervals along the path. Straight-line paths are computed from start and goal poses. Additional waypoints can be specified and the Cartesian planner will attempt to find collision-free, straight-line plans between each waypoint. The Strict version of the Cartesian execution order forces the arm to be able to travel to the goal state whereas the normal Cartesian execution order will result and execute a partial path. Finally, the single query execution order uses the OMPL planner to perform a traditional motion planning query. The only difference between fast and slow planners is the allowable planning time. The OMPL planner could be specified for each request. RRT* was our default planner for its ability to find valid plans and improve them as time allows. RRT plans were fast but caused large swinging motions that are not desired for our picking application.

Perception: CNN Item Identification

Our primary perception pipeline identifies items using a CNN. First, using the kinematic chain of the robot as well as the location of the shelf, the image is masked using the four corner points of the shelf. Next, a pixel-by-pixel labeling CNN, SegNet, is used to label each pixel in the image as shelf or item. After segmentation, the image is divided into superpixels using the SLIC algorithm. Each SLIC superpixel is then classified using an identification CNN based on Alexnet. Once each superpixel is classified, the outputs are merged to solve for the globally optimal solution. The entire vision process is outlined in in figure 8.

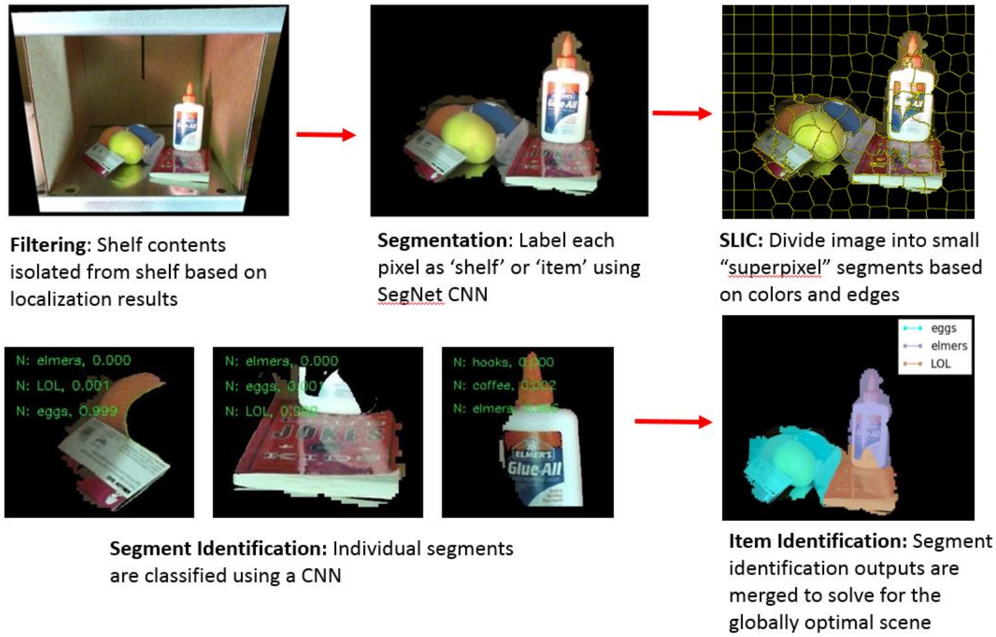


Figure 8: Perception Pipeline

Perception: Dataset Generation

Using a CNN approach for item identification requires a large amount of training data to create a classifier. An automated data collection method was created to generate a database of the 39 possible items. A turntable rotates in 10 degree increments over 360 degrees of rotation and an actuator varies the view angle of the Kinect v2 RGB-D sensor. HSV thresholding and convex hull filters were applied to the images to automatically remove the background from the training images. In total, approximately 100 masked images were captured for each of the 40 items in the item dictionary. These images were rotated, distorted, mirrored, lightened, and darkened to create approximately 400,000 images for future classifier training. The turntable is shown in Figure 9. An example output training image is shown in figure 10.

Figure 9: Data Collection Setup

Figure 10: Example Turntable Data

Perception: PERCH Item Identification

A second perception algorithm runs in parallel to CNN identification. PERCH, Perception via Search for Multi-Object Recognition and Localization, globally searches to match known geometric item models to the Kinect point cloud. PERCH specializes in identifying items with known geometry models under heavy occlusions. PERCH was developed by Vankatraman Narayanan and Maxim Likhachev in CMU's Search Based Planning Lab. An example of PERCH identification can be seen in Figure 11.

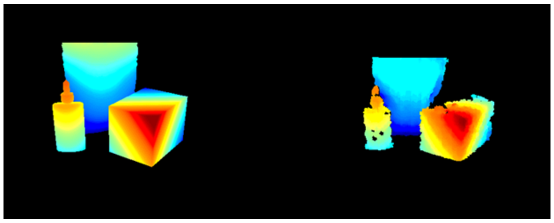


Figure 11: PERCH Output (left) and Noisy Depth Data (right)

Grasping

Initial trade studies showed suction systems would be far superior to traditional grippers for the pick-and-place task. After prototyping several solutions, we determined a high flow system was required to deal with imperfect seals of porous items. A custom suction cup gripper was designed which can be mounted to the end of the UR5. The gripper shown is capable of acquiring 36 out of 38 objects from the 2016 amazon picking challenge list. The grasping system is shown in figure 12 below.

Figure 12: Custom Suction End Effector

All the electronics for the suction system are packed into the enclosure shown in Figure 13. A custom PCB holds an Arduino, reads up to four analog pressure sensors, and controls two AC relays. The box contains inlet power connector, two output plugs to connect two vacuums, two serial connectors to attach up to four sensors, and status LED’s. Pressure sensors installed on the vacuum hose detect when a drop in pressure has occurred, indicating that an item has been acquired. This subsystem communicates with the main computer over ROS serial and is controlled by the main state controller.

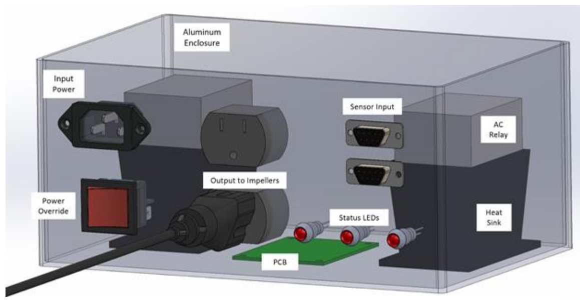


Figure 13: Custom Suction PCB Enclosure

Modeling, Analysis, and Testing

Simulation Environment

All elements of our system visualized in RVIZ. Verifying motion plans and visualizing point clouds aids in debugging. Our simulation setup can be seen in figure 14 below. This shows the robot model, the robot base, the order bin, and the shelf as collision objects, the Kinect point cloud, and the results of the perception pipeline. First, in simulation, the trajectories generated by OMPL are visualized to verify that the arm will not collide with the shelf. In addition, we verify the perception results in real time. The images on the right of figure 14 are the intermediate perception outputs. The final output point clouds are then rendered in the shelf frame.

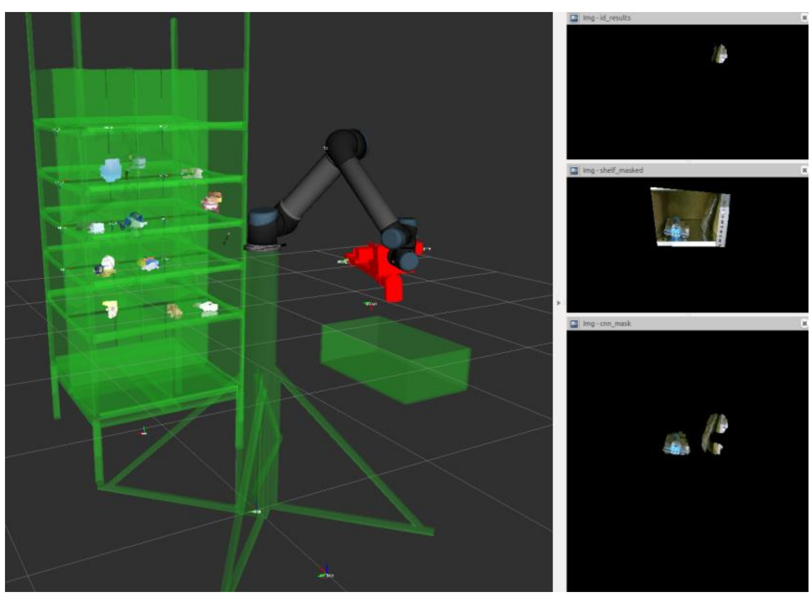


Figure 14: Simulation Environment

Mechanical Design

Before building any hardware, Solidworks was used to understand the workspace of the UR5. Our Solidworks model can be seen in figure 15. In the modeling environment, we were able to move the end effector around inside the shelf to verify the end effector could move into each of the bins. In addition, we verified the mating between the UR5 and our custom end effector.

The base was designed out of 80-20 aluminum. Although 80-20 is expensive, it is very modular and easy to manufacture. This will enable easy transportation to the 2016 Amazon Picking Competition.

Figure 15: System CAD

Reachability

One of the first tests run in ROS was verifying the configuration space of the UR5. To do this, we sampled approximately N points from in the workspace surrounding the robot. From there, we used inverse kinematic calculations to verify that the robot could reach a desired point in the workspace. Figure 16 shows points the robot can reach in the workspace.

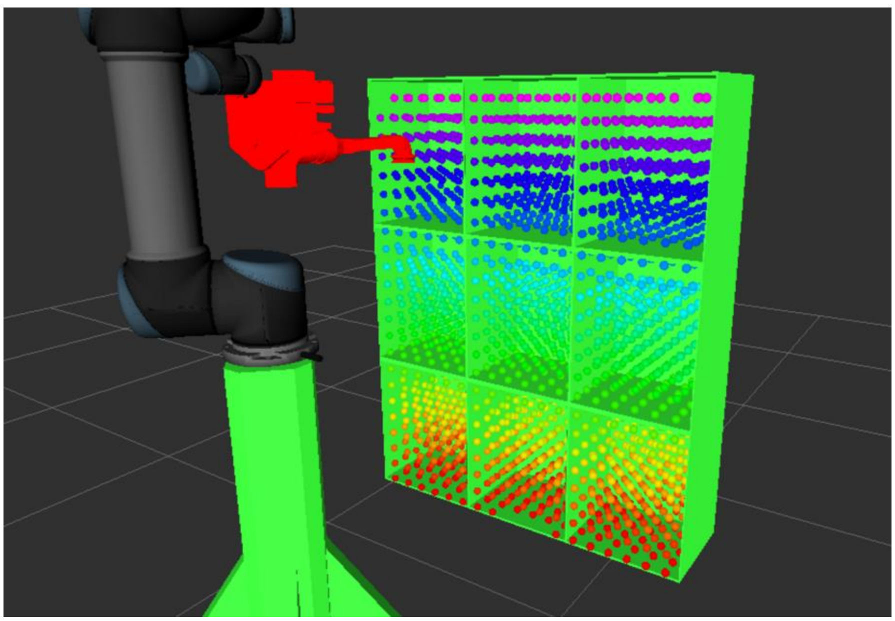


Figure 14: Reachability Study

Suction System Feedback

A pressure sensor, installed inside the shopvac hose, detects when items have been acquitted. A custom ROS node was written to monitor the state of the system. Preliminary results of the suction filter are shown in Figure 17. The raw pressure data (blue) is very noisy. Even after applying a rolling average filter, the signal to noise ratio is fairly small. In the test below, four items are picked up by the gripper. However, the filter only accurately detects three of four pressure drops in the example below.

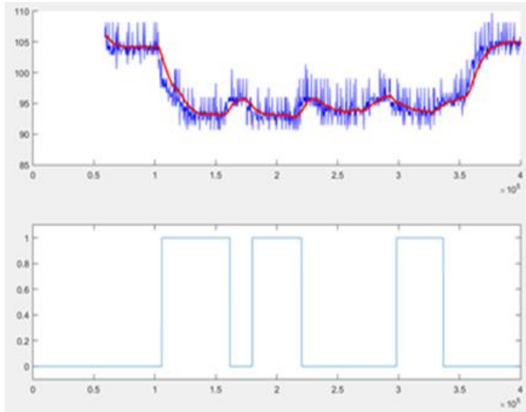


Figure 15: Pressure Sensor Feedback

Perception Testing

A dataset of 120 images was created to analyze the perception system. The images were captured in the same format expected in competition: a python script generates random shelf configurations. This allows for us to quickly verify any changes made to the perception algorithms. To label the dataset, we used a tool called LabelMe. Labeling the dataset allowed us to automate perception system tests to validate algorithm changes. This dataset will be published online after the competition.

SVE Performance Evaluation

The objective is to demonstrate the working of our integrated system. 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 distributed over different shelf bins (between 1 - 4 items per shelf).

Elements

- Complete system demonstration
- Shelf localization
- Collision avoidance
- Item identification
- Item Post estimation
- Path planning
- Grasp planning
- Suction based grasping
- Grasp feedback
- Error handling

Verification Criterion

The SVE performance evaluated verified all system level requirements, including:

- Automatically recognize items in the bin and report results to a GUI on the computer
- Automatically detect object and recognize its pose to find a valid suction surface
- Automatically move the arm to the desired grasping location
- Grasp the item without damaging or dropping it
- Withdraw the item from the shelf bin and place it into the order bin

While testing, we had a 61% success rate with 147 successful picks and 97 failed picks. Items such as the plush puppies squeaky toy, pencils, bunny book, scotch bubble mailer had really good success rates as they had distinct textures and easily graspable surfaces. The table below shows the high level overview of our performance

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

Figure 16: SVE System Test Results

System failures are broken down in figure 19. A majority of our failures were due to identification errors. It is difficult to correctly identify items if they are occluded. Identification failure occurred between items with similar textures such as the 40w light bulb and dove soap. Suction and grasping failure occurs for objects such as the DVD, joke book and duct tape. Also, some items such as the water bottle, glue sticks and command hooks were specular – creating sparse point clouds. This made it difficult to compute correct grasp surfaces. The figures below show our failure analysis statistics. Item by item failure shown in Appendix B.

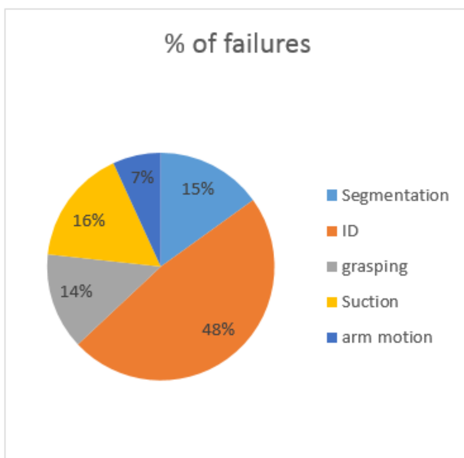


Figure 17: Failure Analysis

Strong and Weak Points

During the course of testing our system, we have been able to collect and log data of the successes and failures of various subsystems. Using this data, we have made observations on the strengths and weaknesses of our project and identified methods to improve overall system accuracy.

Strong Points

- Perception system can correctly identify non-occluded items with an accuracy of 80%. This is higher than our original requirement of 60%.
- We are able to handle partially occluded items which was not in the original scope
- We are able to segment and correctly identify items with 70 %.

- The system is capable of grasping 36 of the 38 items for Amazon Picking Challenge 2016.
- The improved end effector can reach the corners of all the shelf bins and has a thinner profile, increasing our workspace
- Trajectory caching and playback has greatly improved speed and quality of placing items from bin to order bin
- Parallelized perception system has reduced our run-time from 14:30 minutes to 11:30 minutes. The remaining time can be used for picking items missed in the first iteration

Weak Points

- Grasping is the biggest weak point – it is hard to correctly determine grasp surfaces for specular items or noisy point cloud data
- Occasionally, items may brush against the shelf when being moved to the order bin. This is because we are not planning paths with collision checking for picked up items. If suction grasping is not strong enough, the item may fall down
- Certain items such as the shirt and bear toy have a tendency to get stuck inside the end effector
- While using Cartesian IKFast for planning inside the shelf bin, the arm may occasionally collide with the wall
- Items which have similar textures or colors are easily misidentified – we have observed this with items such as the Folgers coffee and joke book, dove soap and the 40w light bulb
- If due to any accidental collision, if the robot's force safety stop gets enabled, there is no autonomous way to reset it

Project Management

Schedule

In order to complete all project goals, a schedule comprising deliverables for each progress review, shown in figure 20. In addition, we kept a more detailed schedule which assigned deliverables to individuals. This full schedule can be found in appendix C. As the Amazon Picking Challenge approaches, we have a day-by-day schedule to ensure all required tasks are completed by competition.

Test	Progress Review
Run FVE using UR5 simulator	PR07 (1/27/2016)
Demonstrate UR5 accessing the APC configuration space	PR08 (2/10/2016)
UR5 autonomy integration demo	PR09 (2/24/2016)
Single item system integration demo	PR10 (3/9/2016)
Fault handling and failure mode testing	PR11 (3/23/2016)
Aggregate statistics for 12-bin run	PR12 (4/6/2016)
Pick at least 3 correct items from the shelf in 15 minutes	SVE (4/20/2016)
Pick at least 3 correct items from the shelf in 15 minutes	SVE Encore (4/27/2016)

Figure 18: Test Schedule

Budget

Overall, the team managed our budget well and stayed under the MRSD requirement of $4000. Luckily, Universal Robots was generous enough to loan us the UR5 for the semester, a robot that sell for around $38,000. Our biggest purchase was of computer parts, which were required to run the robot and train our various CNNs. Detailed budget is shown in figure 21.

Item Description	Cost
Suction Prototype	$ (200.00)
Shop Vac (Quantity 2)	$ (245.58)
Electronics	$ (224.52)
Kinect V2	$ (340.00)
Item Dictionary	$ (100.00)
UR5	$ (0.00)
UR5 Kinect Mount	$ (59.22)
UR5 Stand	$ (125.00)
Computer HDD	$ (245.00)
New Computer	$ (2000.00)
Total	**$ (2838.74)**
Balance	**$ (461.26)**

Figure 19: Project Budget

Risk Management

The risk management chart (figure 22, below) highlights the major technical risks team HARP mitigated during the spring semester. Our biggest technical risks, #1 and #3, related to the perception system. Through lots of hard work and software development, these were mitigated. Gripper design was another major risk, since we did not know the item dictionary until about two months ago. Luckily, no major changes occurred between this year and last.

Risk Management

	Risk	Type	Cause	Mitigation
1	Perception system cannot segment or identify occluded items	Technical	Algorithm fails segment properly Algorithm fails to identify	Developing multi-tiered approach to use all available data
2	Gripper design may be insufficient for new 2016 rules	Technical	Requirements change	Have several designs in mind anticipating rule change
3	Perception algorithm does not scale on crowded shelf bins	Technical	ICP algorithm requires large amount of computation	Parallelize algorithm Benchmark algorithms Get better computer
4	System Integration will expose problems we cannot fix fast enough	Technical / Mgmt	Poor estimation of required work Significant last minute issues	Apply project management best practices
5	Robot uncertainty is to high to localize or pick items	Technical	Kinect-robot uncertainty, point cloud stitching, localization error	Characterize offsets Evaluate Deviation

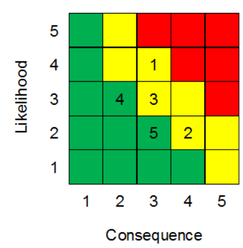


Figure 20: Risk Management

Conclusions

As the MRSD project course comes to an end, we have completed our project successfully while increasing the original scope to align with newly released Amazon Picking Challenge 2016 rules. We would continue making improvements to our system during summer to give us a competitive edge.

Lessons Learned

Perception was the hardest subsystem to develop. We had to try many different approaches and algorithms before we found one suitable for our item set and lighting. Using software from research papers of previous APC teams did not help us as they were designed for items placed without occlusions. Also, for training a CNN, dataset generation is an essential step. We had to build our own turntable and this gave us good insights on generating data with different lighting, background and item orientations.

We chose to use suction end-effectors over tradition robot manipulators. Using a conventional manipulator would have provided us the means to grasp all the 38 objects and given a greater freedom and control over item manipulation. However, the effort required to design the planning algorithms was outweighed by the simplicity of the suction end-effector.

We also made a good decision to first build the simplest version of the entire system before focusing on improving any one sub system. This allowed us to iterate improvements for various sub-systems, helping us to refine them independently.

It was a good lesson to break up trajectories into several smaller parts and cache common trajectories instead of replanning every time. This helped us speed up the entire system.

Our biggest take-away was learning how to design the complete software architecture and keeping various components like grasping modular. Since this was the first time we developed a complete system, we made a lot of mistakes and had to refactor our code numerous times. Using ros param server to store configurations and settings helped us greatly speed up development and tested as we did not have to recompile the program every time. In order to ensure compatibility between various software modules, we created a software specification document to capture all the interfaces and flowchart.

Another important lesson was to choose the right robot for the task. We took a risk to switch to UR5 from PR2 and develop software for it even though we did not have access to UR5 robot at that time. The PR2 had several drawbacks for use in our particular application. It would have been much slower and would have added another layer of complexity for base and spine planning.

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.

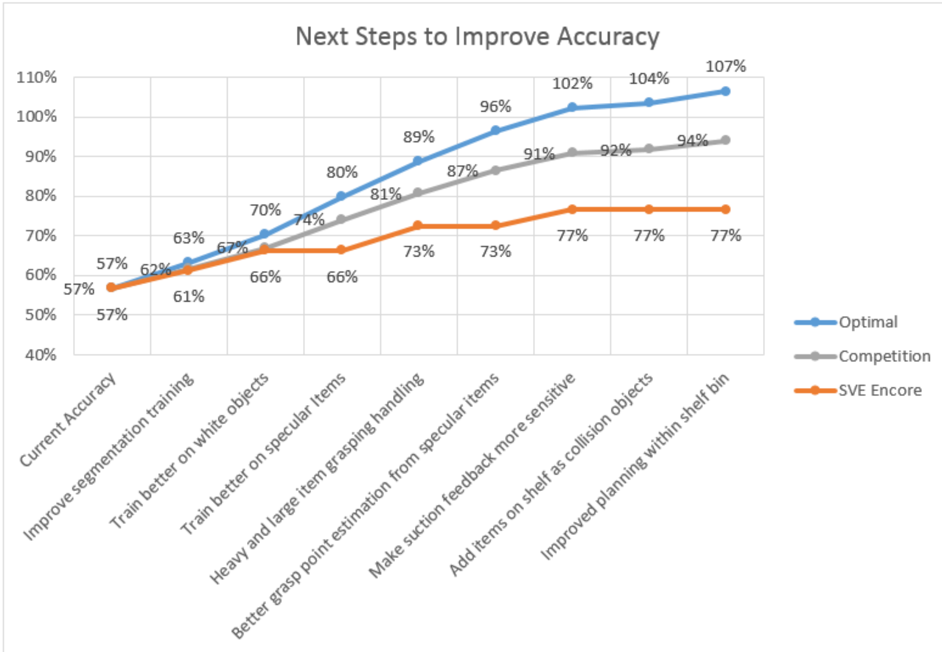


Figure 21: Future Work Before the 2016 Amazon Picking Challenge

Perception system would be modified to improve performance for occluded item segmentation and white and specular object identification. For planning subsystem, we need to the functionality to check trajectories for collision when an item is grasped by the end effector. This would help prevent arm motion failure when the item collides against the shelf while moving to the order bin. Sideways grasping will be implemented to pick up books and other items placed close to the shelf walls.

For the SVE, our overall accuracy was around 61%. With these improvements, we would be able to increase accuracy to at least 80%

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.

NVIDIA. sbpl UNIVERSAL ROBOTS

Figure 22: Project Sponsors

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/

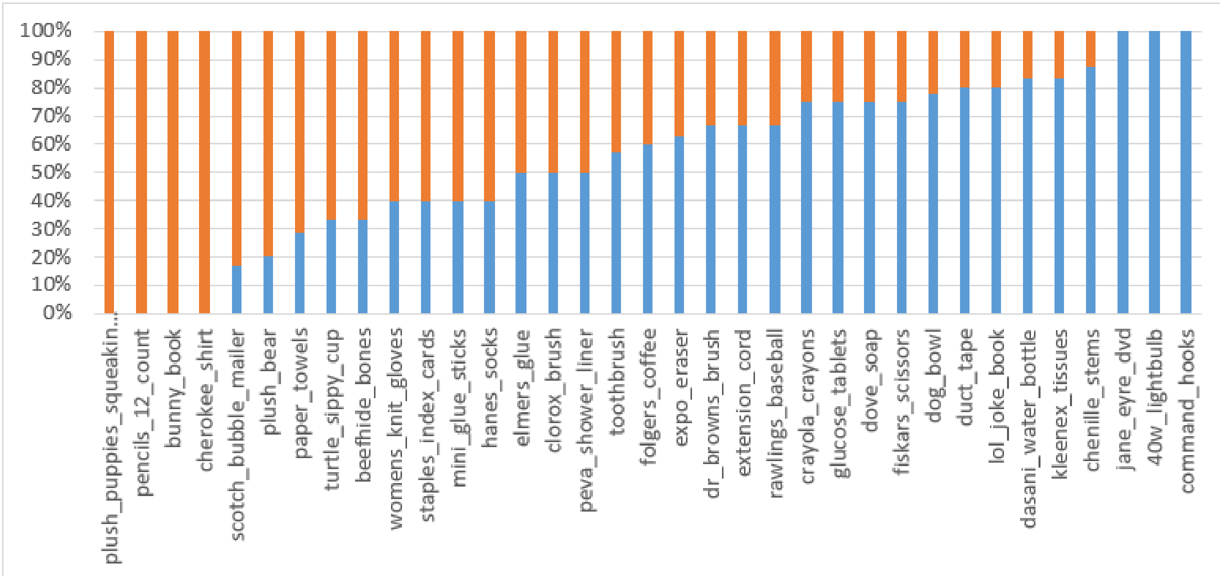
[8] Robot Operating System: http://www.ros.org/

[9] MoveIt!: http://moveit.ros.org/

[10] Open Motion Planning Library: http://ompl.kavrakilab.org/

Appendix A: 2016 Item Dictionary

Appendix B: Item-by-item Failure Analysis



Appendix C: Detailed Schedule

	% Co	Task Mode	Task Name	Duration	Start	Finish	Pre	Resource Names	Add New Colu
1	**0%**		**Perception**	**12 days**	**Thu 3/31/16**	**Fri 4/15/16**			
2	0%		Create Models of New Dictionary Items	7 days	Fri 4/1/16	Mon 4/11/16		Lekha	
3	0%		Integrate Perch	2 days	Thu 3/31/16	Fri 4/1/16		Rick	
4	0%		Test Perch	7 days	Sat 4/2/16	Sun 4/10/16		Venkat	
5	0%		Train Filtering CNN	4 days	Thu 3/31/16	Tue 4/5/16		All	
6	0%		Super Pixel Segmentation for ID CNN	2 days	Fri 4/1/16	Sat 4/2/16		Rick	
7	0%		Post Process ID CNN Results	2 days	Sun 4/3/16	Mon 4/4/16	6	Rick	
8	0%		Evaluate ID CNN on 5 Items	3 days	Tue 4/5/16	Thu 4/7/16	7	Rick	
9	0%		Train ID CNN	5 days	Thu 3/31/16	Wed 4/6/16		All	
10	0%		Refine ID CNN	7 days	Thu 4/7/16	Fri 4/15/16	9	Rick	
11	**0%**		**Grasping**	**8 days**	**Thu 3/31/16**	**Mon 4/11/16**			
12	0%		Handle specular objects	5 days	Thu 3/31/16	Wed 4/6/16		Feroze	
13	0%		Get grasping demo on Kiva Pod	5 days	Thu 3/31/16	Wed 4/6/16		Feroze	
14	0%		Demonstrate Multi Item Picking	8 days	Thu 3/31/16	Mon 4/11/16		Feroze	
15	**0%**		**Hardware**	**23 days**	**Thu 3/31/16**	**Mon 5/2/16**			
16	0%		Design Base and Kinect Mount	6 days	Thu 3/31/16	Thu 4/7/16		Rick	
17	0%		Send base to Paint		Sun 5/1/16		16	Rick	
18	**0%**		**Stowage Task**	**12 days**	**Thu 3/31/16**	**Fri 4/15/16**			
19	0%		Item Identification Pipeline	11 days	Fri 4/1/16	Fri 4/15/16		Abhishek,	
20	0%		Shelf Stowage Planning	11 days	Fri 4/1/16	Fri 4/15/16		Ihsane,Ale	
21	0%		Test grasping with Amazon order bin	6 days	Thu 3/31/16	Thu 4/7/16		Abhishek	
22	**0%**		**Deliverables**	**55 days**	**Wed 4/13/1**	**Wed 6/29/16**			
23	0%		Aggregate Statistics for 12-bin Run	0 days	Wed 4/13/1	Wed 4/13/16			
24	0%		National Robotics Week Demonstration	0 days	Thu 4/14/16	Thu 4/14/16			
25	0%		Spring Validation Experiment	0 days	Wed 4/20/1	Wed 4/20/16			
26	0%		Spring Validation Experiment - Encore	0 days	Wed 4/27/1	Wed 4/27/16			
27	0%		CMU Semester Ends	0 days	Tue 5/17/16	Tue 5/17/16			
28	0%		Amazon Picking Competition	0 days	Wed 6/29/1	Wed 6/29/16			
29	**0%**		**Failure Mode Testing**	**0 days**	**Sun 5/1/16**	**Sun 5/1/16**			
30	0%		Revist after Demonstrations		Sun 5/1/16			TBD	
31	**0%**		**Executive Functions**	**6 days**	**Sun 5/1/16**	**Mon 5/9/16**			
32	0%		Develop Feature to Move Non-Target Ite		Sun 5/1/16			TBD	
33	0%		Improve System Runtime		Sun 5/1/16			TBD