Individual Lab Report 9
Progress Review 10

Shivam Gautam
sgautam@andrew.cmu.edu

Team Daedalus
Mohak Bhardwaj, Pranav Maheshwari, Shivam Gautam, Richa Varma, Dorothy Kirlew

March 18, 2016

1 Individual Progress

I primarily worked on coding the Global Motion Planner. This involved coding the following sections-

1. Creating a ROS package for 'world-time' in the simulation
2. Creating the Environment config file
3. Writing the Global-Planner ROS package
4. Mapping and Navigation of the Oculus Prime

1.1 Creating the Environment config file

I worked with Pranav and Mohak to establish common ground for the Global Planner, the local planner and the visualization packages. The first step was to decide on a common environment for the parking lot as each of the three systems would receive a copy of the environment. The environment file was generated by creating an image of the parking lot and then converting it into a 2D array using MATLAB and is depicted in figure 1.

Apart from the 2D array of the environment, the file also include the environment parameters. Details for the environment parameters are as follows-

1. Robot Dimensions: 3m x 1.5m
2. Spot Dimensions: 3.5m x 2.5m
3. Number Of Spots: 104
4. Lane width: 5m
5. Environment Dimensions: 48m x 42.5m
6. Cell Resolution: 0.025m x 0.025m
7. Start Location(x,y): 2.5m, 2m
8. Exit Location(meters,x,y) 40m, 40m

After creating the 2D array, I created a ".cfg" file which could be used for planning in SBPL (Search Based Planning Laboratory)

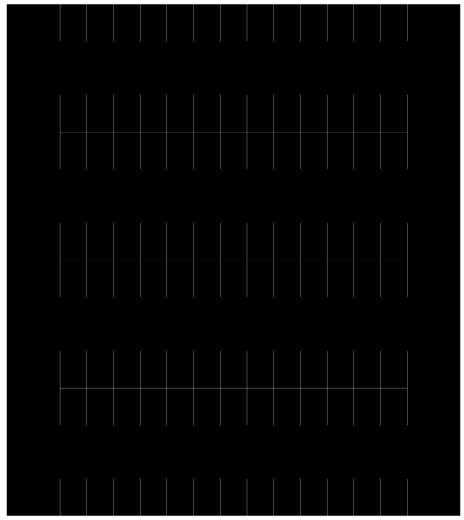

Figure 1: Simulated Parking lot

1.2 Writing the Global-Planner ROS package

The Global Planner decides on which spot to assign to a car currently at the entry. The spot selection is currently based on the following heuristics-

1. Distance to Exit
2. Proximity to Claimed/Occupied Spots
3. Cars in Queue
4. Time of Day

Since the last progress review, I have been working on writing the 'Distance to Exit' metric for the planner. I decided to work with the SBPL library for implementing this as it allows us to choose from a variety of path planners (ARA*, MHA*, AD* etc.).

SBPL essentially breaks the problem of searching in an environment into two aspects- break the 2D environment into a X-Y-Theta graph and then search this graph for a path to the goal. For the purposes of the planner, I needed to modify the existing planner as well as the lattice environment. I modified the former part for the MHA* (Multiple Heuristic A*) to include information about an additional heuristic- inflation of obstacles for objects like cars that have a different footprints in 3D and 2D. This would be useful in inflating the cost values of the spots that have been already occupied.

I also worked on an additional program that calls the SBPL planner on not only a single goal on the environment but on multiple starts and goals. This is great for a search that starts at all the parking spots of the lot and searches a path to the exit. I did this by selecting a goal in the 'willow-garage' environment file and selecting multiple spots by around it. The spots were selected based on a gaussian distribution around the goal. An instance of this search for the ARA* planner is depicted in figure 2. I tweaked the visualization to depict actual costs of cells from 0-255. The visualization shows a robot starting from the bottom left corner of the map and moving to the upper right corner.

To get a better understanding of the library, I different searches on the same environment and compared them with respect to the number of expansions it took to reach the goal state, the length of the path to the goal state and the time it took to return the first solution. The ARA* performed much better than the MHA* in these aspects apart from a few cases where MHA* took less expansions to reach the goal.

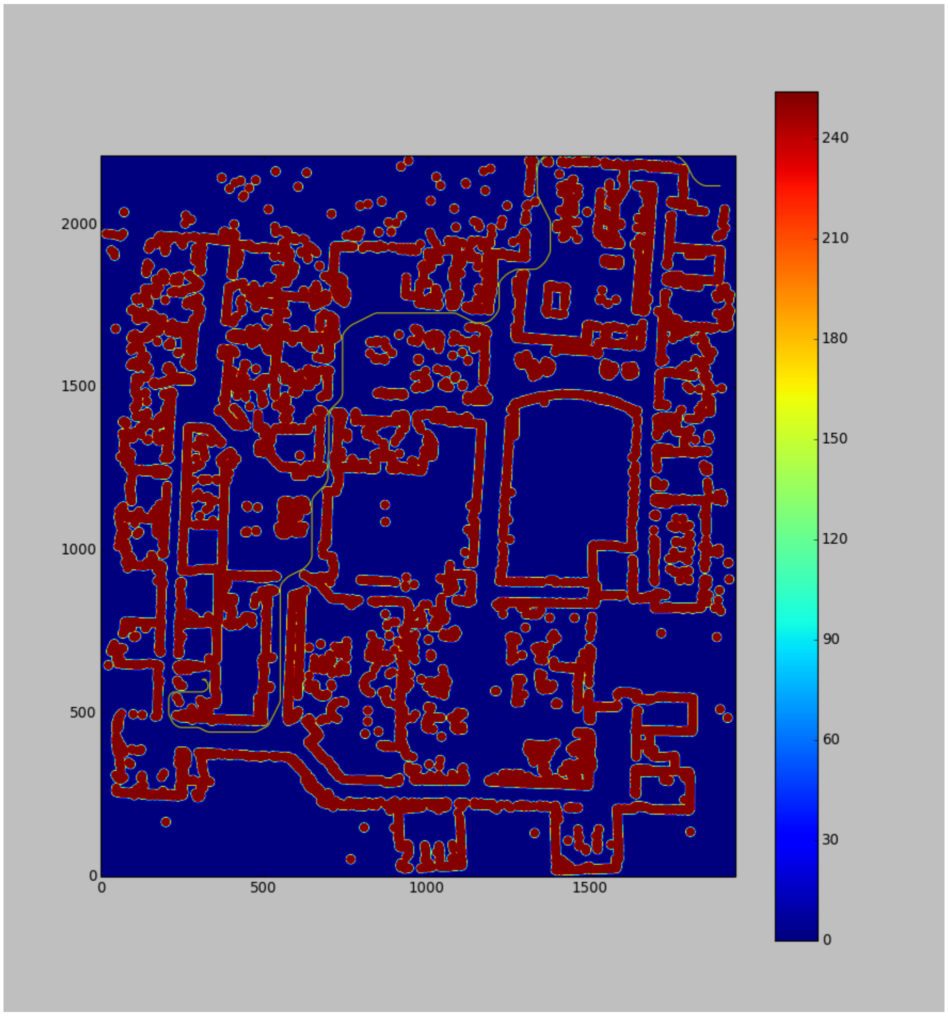

Figure 2: Anytime A* search on the Willow garage environment

1.3 Creating a ROS Package for 'World-Time; in the simulation

I created a ROS package that publishes the current time in the parking lot. The package observes the current ROS time and scales it to reflect the current time in the lot. This time is used by the global planner as a heuristic- the planner assigns a different spot to the car based on the time. The spot assigned to a car would be different if the car enters at the beginning of peak-hours as compared to when it enters during non-peak hours. Since we use the ROS time, it is synchronised accross all packages that might want to use it. The world time would also be used by the script that generates the cars for the entry queue.

2 Mapping and Navigation of the Oculus Prime

Unlike the previous progress review, we decided to test the mapping on a bigger parking lot. The parking lot is depicted in 3. The lot was made of cardboard and was roughly square because of the bends in the cardboard. The map created for the lot is depicted in 4.

The map is made by running the GMapping package and taking 3D depth data from the Asus Xtion. The depth data is converted to a laserscan ROS message which is used by the laserscanmatcher ROS package to match adjacent laser scans. This data, combined with the odometry data from the platform, is used by

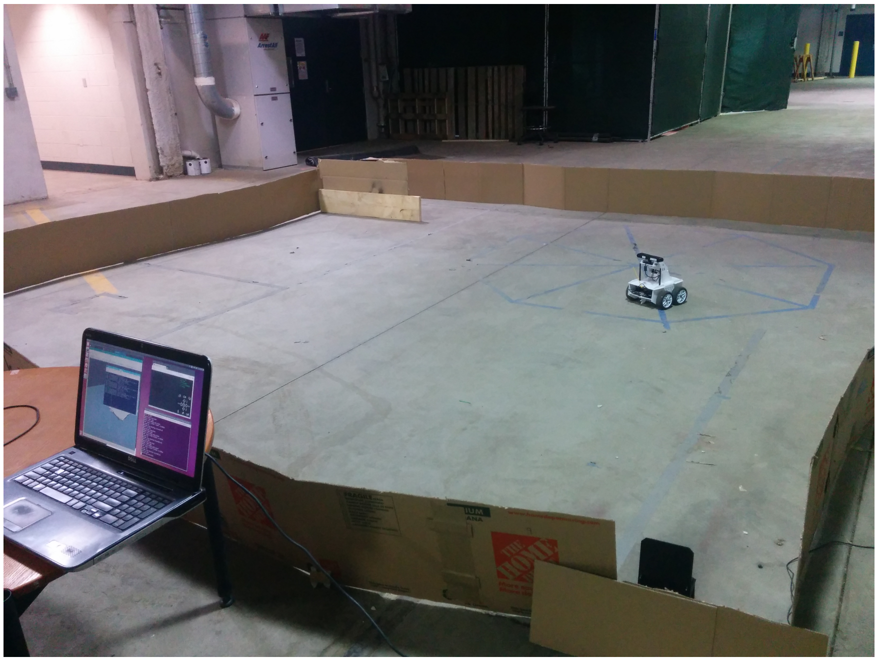

Figure 3: Parking Lot

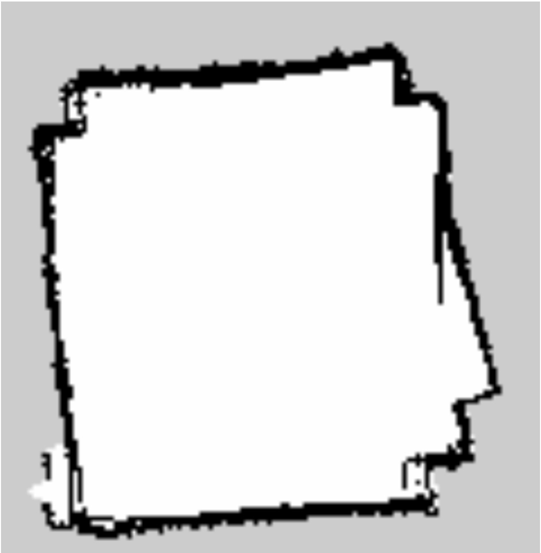

Figure 4: Map created of the parking lot

GMapping to map the environment.

For the navigation of the platform, Pranav's script for accepting waypoints and relaying it to the platform via Telnet was tested. I was able to work on the association of points on the map with actual waypoints and how to add waypoints without using the Oculus Prime web interface. 5 depicts the waypoint plan. Once this was complete, we could test the vehicle motion to a waypoint directed by the communication system. We also tested the obstacle detection for the platform as can be seen in 6.

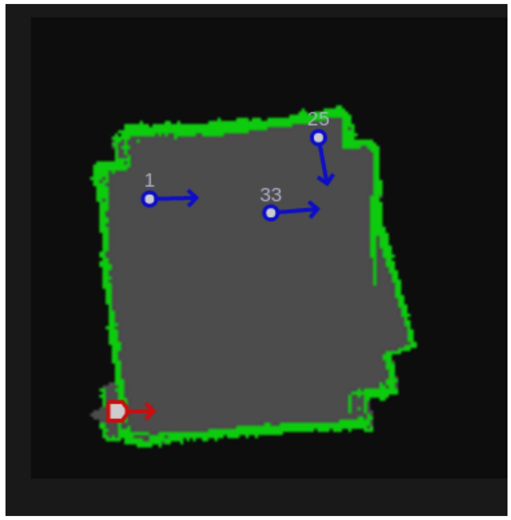

Figure 5: Waypoint plan for different spots

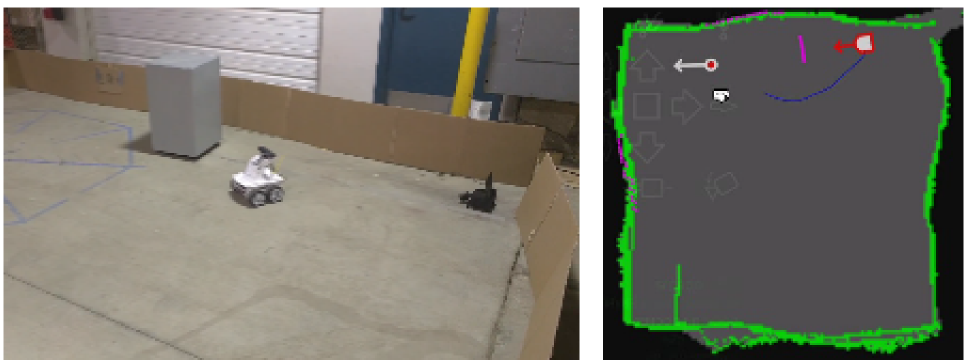

Figure 6: Oculus Prime Avoiding an obstacle (Photo Credits:Pranav)

3 Challenges

The most notable challenge I faced this week was working with the platform to integrate all the subsystems and run the code from end to end. This required fixing a few minor issues and troubleshooting the system. Also, mapping proved to be a challenge as we had to re-map the environment each day before starting testing due to the parking lot not being permanent. The issue I faced with mapping was that it was time consuming to generate maps that were sufficiently refined to be used by AMCL.

Another issue I faced was deciding on a common environment with Pranav and Mohak as we had to figure out the best possible configuration parameters that worked for everyone. During working with SBPL, I faced a lot seg-faults while compiling before I could figure out that overwriting the the 2D-grid search data structure for the SBPL was not a good idea. A straightforward fix to this was to reinitialize an environment variable for each spot. However, this is not efficient and I would be working on modifying the environment file to better handle multiple goal locations so as to eliminate this problem all together.

4 Teamwork

Dorothy and Richa worked together to refine their code for the communication system and the integration with the planner. Richa looked into integrating the motherboard into the old Oculus Prime platform. Pranav worked with Richa to integrate the code written for the communication system with the platform. I worked with Pranav to create maps of the parking lots we created during the past few weeks. I also collaborated with Pranav while we were trying to figure out the navigation system of the Oculus and test the full pipeline on the platform. For the simulator, I collaborated with Mohak and Pranav to discuss the outline of our codes, the environment of the simulation and the ROS services that would be needed. Besides this, Pranav and Mohak worked on creating wooden reinforcements for the parking lot and I chipped in wherever I could to help them.

5 Plans

We plan to complete a beta version of the simulator by next week so that we can test the three aspects of the simulation working together, even without all heuristics implemented. For this, Pranav would be working on the visualization, Mohak on the local planner and I would be working on the global planner. Specifically, I would be working on adding ROS functionality to my code to create the service routines to communicate with the local planner and the visualization. Dorothy and Richa would be working on getting the old platform ready from the hardware and software end. Dorothy would also be working to create a wooden parking lot so that we can eliminate the problem of having a non-permanent parking lot. Richa would also be coordinating with Dorothy to work on another platform that has been ordered but is yet to arrive. We expect to integrate the second platform before the next progress review and show two platforms parking collaboratively in a parking lot.

6 References

1. Search Based Planning Laboratory- www.sbpl.net