

Progress Review 12

Individual lab report – 11 || April 11, 2016

Team Daedalus
Pranav Maheshwari

Team Members:
Mohak Bhardwaj
Dorothy Kirlew
Richa Varma
Shivam Gautam

1. Individual Progress

- ❖ Simulation
- ❖ Physical Setup
- ❖ SVE Demos

Simulation

- **Rendering Script**

To enhance the visual appeal of the simulation environment and make it more intuitive, I added certain 3D visual elements. I replaced the native cube marker in RViz with 3D models of cars, as seen in Figure 1, which give a better sense of the direction in which the car is headed. I also added Exit and Entrance signs to denote the flow of traffic in the parking lot. The final layout is presented in Figure 2.

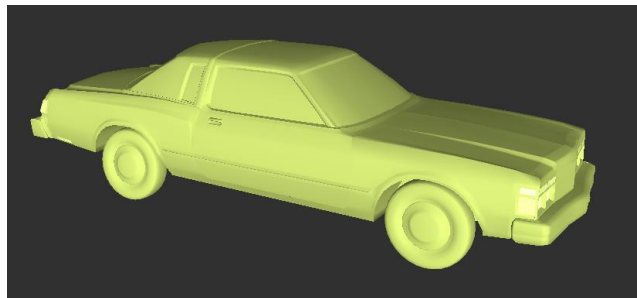


Figure 1 3D model of a coupe

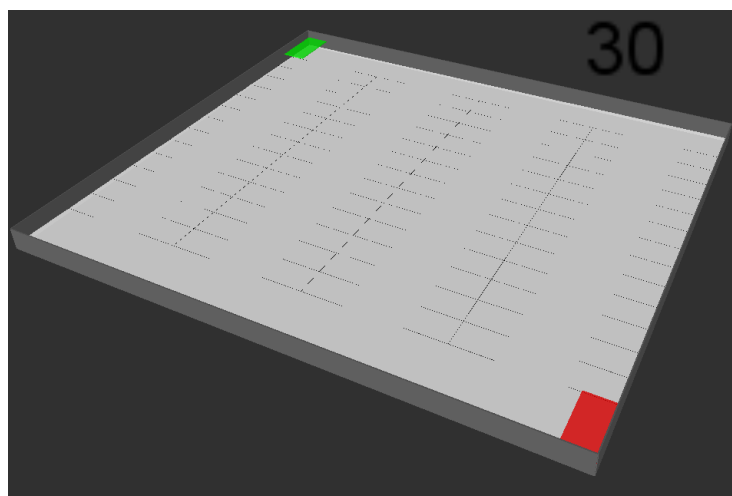


Figure 2 Final simulation environment with markers (Green: Entrance & Red: Exit) and a timer

Previously the rendering script didn't consider collision between different vehicles which led to overlap between multiple markers. To fix this, I created a "Pause" functionality in my script which stops a vehicle as soon as it reaches in a certain proximity region of another vehicle. Priority is assigned to vehicles based on their time of arrival in the parking lot and time of recall. As seen in Figure 3, the vehicles have a buffer region around them, which must be obstruction free before they can move.

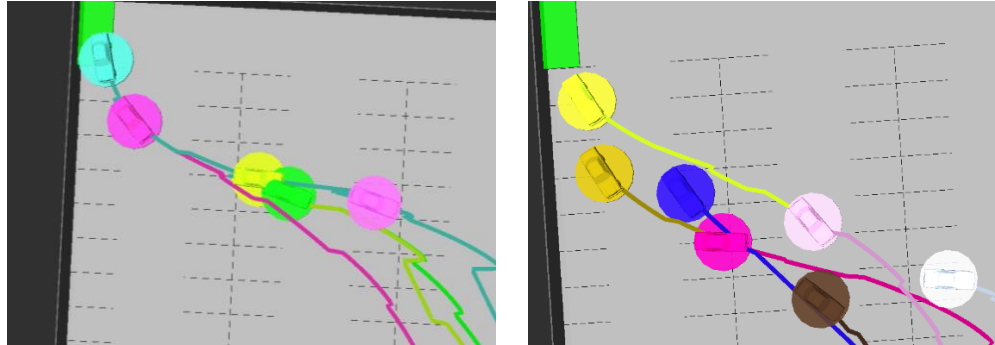


Figure 3 Cars tightly moving in coordination with each other with and without overlap (before and after)

- **Performance Metrics**

To get a tangible way of assessing our system, it is imperative to quantify the performance via certain metrics. To achieve this, I added a function to create a log of cumulative time taken by vehicles to park, return and total time spent by vehicles in "Pause" state. We will use these metrics to guide our future testing and tuning of weights given to various heuristics and cost factors.

- **Local Planner**

Mohak after having created the entire local planner, a continuous space A^* , was facing some issues in having it work in our environment. The planner was exhausting out of options while searching and wasn't able to successfully detect obstacles, causing it to plan improper paths. To fix these issues, we made three important changes; edit the environment file, correct the motion model, and add additional motion primitives. We processed the map to increase the size of obstacles and further added an inflation layer around them, as seen in Figure 4, to increase the cost of paths being planned in that vicinity and make the car stay clear. The motion model had certain numerical inconsistencies, causing it to compute incorrect trajectories.

Lastly, based on our observations during testing, we created additional motion primitives to help the planner converge faster towards the goal. All these measures helped in fixing the planner and greatly improved its performance.

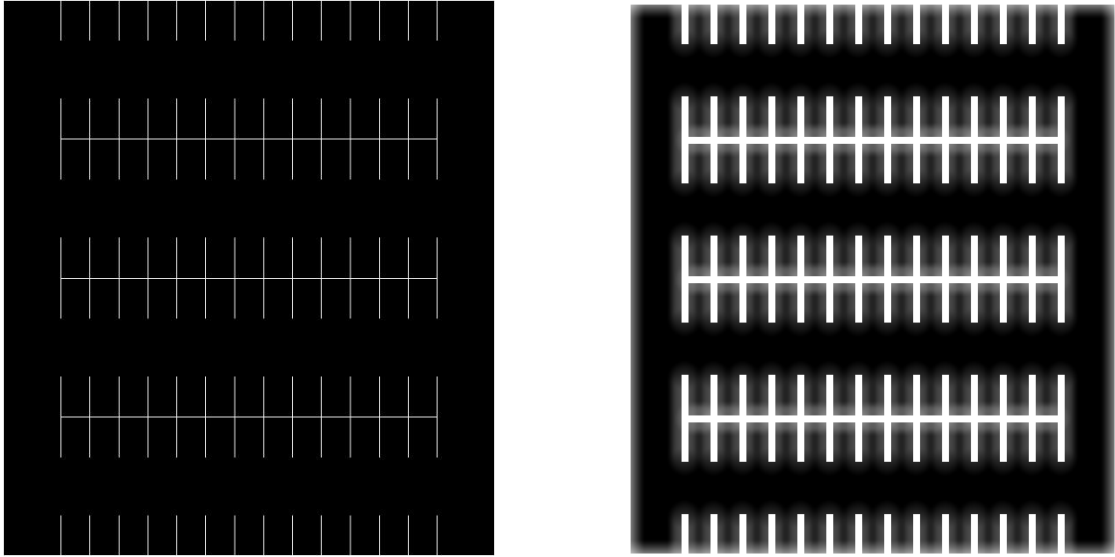


Figure 4 Environment file for local planner (before and after enhancements) (Collaborators: Mohak and Pranav)

- **Integration**

Shivam and Mohak had already created services to integrate their planners with the backend script. To transmit the data regarding state of the parking lot, motion segments, etc., I created three services, one for local and two for global planner. These services have associated callback functions which assess the state of the parking lot and process all the raw information before sending it to the planners.

- 1) **global_state_render**

Request: Boolean

Response: List of spots with their state (empty, claimed, or occupied)

- 2) **global_state_backend**

Request: Boolean

Response: Number of vehicles in the queue

3) local_state

Request: Boolean

Response:

- List of spots with their state (empty, claimed, or occupied)
- Density of vehicles in the parking lot

- **Testing**

After setting up these services, we were in a position to initiate the testing of our final system. As seen in Figure 5, vehicles were able to plan their way from entrance to parking spot. This is pretty close to our final scope for the simulation, as envisioned by the team. We'll be keeping a log of our test data to analyze and draw some conclusions about our system.

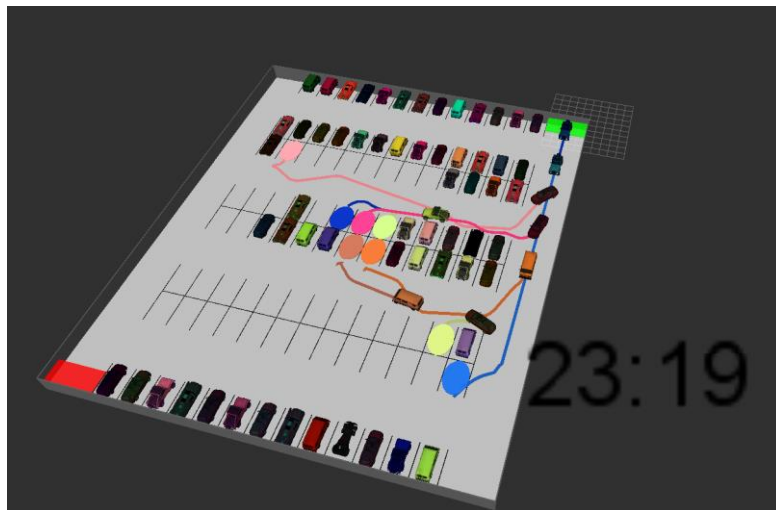


Figure 5 Simulation Environment in Action

Physical System

- **Navigation**

The physical system has been close to completion since a couple of weeks now and only required slight tweaks to make it perform robustly. I worked towards iterative testing and tweaking on the navigation related aspects of the platform. As per the test plan, I created a log of the errors in the final and desired position. As seen in the following table, the platform tends to deviate from its desired position and heading by a certain amount, which eventually adds up and either causes the platform to fail in planning a path or collide with infrastructure.

Test Spot	Deviation in Position	Deviation in Orientation	Collision
Intermediate Waypoint 1	12cm Forward	Negligible	No
Parking Spot 1	4cm Right	NA	Yes
Exit Spot	2cm Forward	Negligible	No
Parking Spot 1	8cm Right	NA	Yes
Exit Spot	2cm Right	5 degrees	No

I used this data and my previous routine to create a final system which performed with higher reliability. Through this testing, I could ascertain that there needs to be minimum of 30cms gap between intermediate spots and parking spots for the platform to enter the spot without collision. Also, the inaccuracies in the heading of the vehicle, which anyways are slight, don't cause any issues since these continuously get corrected by the platform. As can be seen in Figure 6, a network of waypoints has been created in a strategic manner such that the platform can traverse through them robustly and without colliding with the infrastructure.

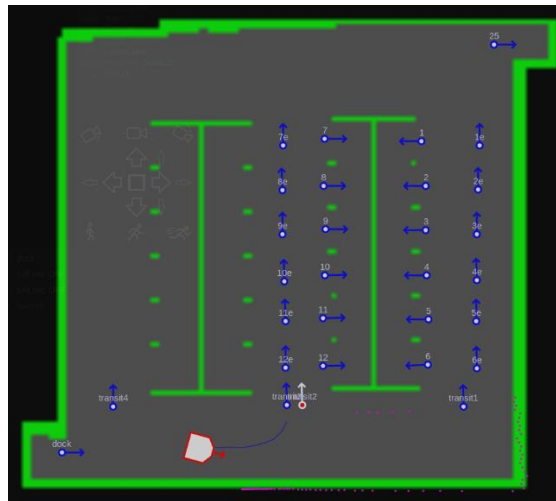


Figure 6 Waypoints for Navigation (Collaborators: Shivam and Pranav)

As per the routine that was created previously, there are interim waypoints for the platform to go to before it enters or exits a spot. These spots face the exit zone and assist the navigation stack to plan path with lower number of recovery rotations and failures. The map has also been edited to be in line with our modified parking lot which is now devoid of any partition between parking spots. It now has some virtual boundaries which ensure that the planner doesn't make the platform travel through the spots and at the same time don't obstruct the platform from entering the spot.

SVE Demos

- **Demo #1**

Setup:

The setup for Demo #1 requires the assembly of the parking lot and placing the platform in the entrance region. A laptop is started to run the User Interface, SSH in to the platforms, and execute the Launch files. Two phones are needed to connect our mobile app to the platforms.

Performance:

Owing to some hardware issues, we couldn't use the mobile app for this demo, which otherwise is fully functional and has been used in the past. Barring this issue, both the platforms were able to:

- 1) Communicate with and update the user interface
- 2) Communicate with each other
- 3) Take into account the virtual vehicles
- 4) Avoid obstacle
- 5) Completely park and return from the spot

Future Work:

Since we have received a new platform now, we'll be using the two fully functional platforms for our future tests. We plan to improve the robustness of our system.

- **Demo #2**

Setup:

The setup for Demo #2 simply requires to start a Launch script on a laptop which has all the necessary packages.

Performance:

The test went fairly well. We were able to meet all our performance objectives, which were:

- 1) Render the environment in RViz
- 2) Use a global planner to assign spots
- 3) Use a local planner to plan path till those spots
- 4) Avoid collision between vehicles

Future Work:

We'll be maintaining logs of our test runs to help guide us while we tune our heuristics and cost functions. We aim to draw comparisons in the performance our system vis-à-vis any normal non cooperative system.

2. Challenges

The work on the simulation environment has been relatively easy, barring few issues with setting up the services and fixing the local planner. The physical system on the other hand has been much more difficult to work with. We had to deal with plenty of hardware issues while readying our system for the progress review. The connector for the Solid State Drive on one of the platforms was coming loose, making the platform reboot abruptly. We had to dismantle the platform multiple times to fix that. The depth sensor was also giving us erroneous values and we eventually replaced it.

Our current configuration of the parking lot is also a bit tricky to work with for multiple platforms because the entrance queue only supports a single platform. This on few occasions made the platforms either collide with each other or struggle with localizing themselves.

Apart from these few challenges, things have been fairly smooth and we plan to rectify all our issues with iterative testing.

3. Teamwork

Shivam and I worked on the Navigation and Localization of the physical platforms. Mohak and I worked on the local planner. Mohak, Shivam, and I also worked on readying all the demonstrations before the progress review. Shivam worked on the global planner which is now mostly complete. Dorothy and Richa worked on implementing a modified A* to our system and they have a working version in the simulation which they'll implement in the coming weeks on the physical platforms.

4. Plans

All our systems are ready now and only require testing and tweaking before the SVE. Mohak, Shivam, and I will be working towards collecting and analyzing data from the simulation. Dorothy and

Richa will be working towards implementing their improved multi agent planner. We are on schedule and in a good position to deliver all our promised work.