Automated Driving Using External Perception

Individual Lab Report - ILR09
November 2, 2023

Team E - Outersense

Author:
Ronit Hire

Team Members:

Atharv Pulapaka
Dhanesh Pamnani
Jash Shah
Shreyas Jha

Contents

1 Individual Progress

The goal of this PR was to focus on full scale testing of our system – 2 controlled cars driven by 3 infrastructure sensors. Along with the global mission planner, the local planner was also in the loop and generating path for the controls to follow. My efforts for this PR were focused on enabling obstacle detection with the help of Outersense markers and testing the robustness of the perception pipeline.

1.1 Perception

1.1.1 Obstacle Detection

To enable static and dynamic obstacle avoidance in our system, the planner needs information of all the objects present on our track. Our bird's eye view perspective coupled with a niche object type-RC cars prevent us from using any off-the-shelf classification models. As showcased in our previous PR's we have pivoted to using Aruco makers to detect and track our controlled vehicles and since already had the code to detect our custom Outersense logo from the SVD, we decided to use it as a marker for obstacles. So I focused on modifying the perception system to accommodate these two separate detection modules. The results can be seen in the figure Fig.[1]. The outersense logo (obstacle) is near the top of the image and aruco marker is tracked in the bottom half.

1.1.2 Robustness Testing

As per our planned schedule for the fall semester, we will soon be freezing the development of our perception system and to that end I have started its rigorous testing to identify the critical failure scenarios. Of the 3 major sub-modules of perception – detection, tracking and pose estimation, I concentrated my efforts on testing the tracking and pose estimation modules. We use optical flow to track SIFT features in a region of interest around the car. I tested the quality of optical flow by repeatedly moving the car in a straight line and observing the drift accumulated and concluded that the shift is in an acceptable range . To test the reliability of pose estimation, I compared it against different ground truth locations which were manually marked on the track. Over repeated runs, the maximum eucledian error was observed to be $< 20cms$ which is well within our tolerance limits. I also observed some problems in the system which are discussed in the challenges section.

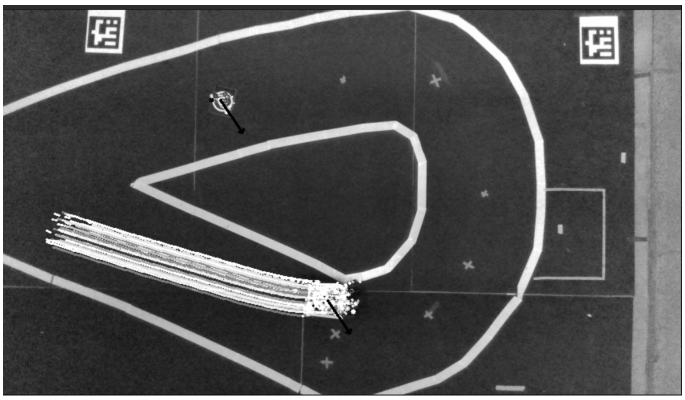

Figure 1: Detection and Tracking of two different markers

2 Challenges

2.1 Pose Latching

If the perception system has been active for over 30 mins with the RC cars exiting the field of view of one camera and entering another, we observe that perception sometimes continues to report pose from the previous camera. We suspect this to be a quirk of using python ROS as it does not allow for manual garbage collection. In our system, each detected vehicle creates a “VehiclePose” type object which is responsible for publishing its own pose. This object is supposed to be destroyed as soon as the vehicle leaves the field of view but python does not allow a manual call to the object destructor. We are currently investigating methods to avoid this behaviour and in the worst case we might have to rewrite some parts of the pipeline. This is not a critical issues as of now because 30 mins well over our desired operational time but we still want to mitigate any risk.

2.2 Noisy Pose

Of the three Realsense D435 cameras which we are mounted on our infrastructure units, we observed a substantially higher noise in pose estimates coming from the third camera. The reason for this seems to be a combination of frequent frame drops and a higher detection failure rate. We suspect this to be a hardware issue coupled with a poor calibration. Our plan is to recalibrate this camera with the proprietary Intel Realsense software and retest for improvements. Currently the increased noise is dealt by our State Estimation block which fuses odometry and IMU data from the RC car to improve the heading estimates and reduce reliance on perception.

2.3 Slow planning around obstacles

As we started testing the planning pipeline, we noticed that the planner functions at the desired frequency in normal in-lane driving scenarios. But as soon as an obstacle in introduced on the track it takes a long time to come up with feasible plan respecting the kino-dynamic constraints of the car leading to a collision. Our initial approach to tackle this challenge is to reduce the map resolution. We will also be trying out better heuristics for the current Hybrid A* implementation.

3 Team Work

Along with individual perception tasks, as a team we also made progress on aspects like Cruise Control, Data Association and Planning.

- **Jash Shah**: Jash worked on adapting a off-the-shelf Hybrid A* planner to run in an online manner for our use case. He collaborated with Dhanesh to add ROS bindings and integrated it into our system. He is now extending it to support multiple cars.

- **Shreyas Jha**: After developing the initial state estimation block, Shreyas worked on tuning the UKF filter which fuses pose from perception with the onboard odometry and IMU. He is also working on visualizing the pose estimates in RVIZ for easing the debugging processing.

- **Dhanesh Pamnani**: Dhanesh collaborated with Jash, to successfully integrate the planning block in our system. He worked extensively on map generation and a goal-publisher for the planner. He is currently working on incorporating the RC car parameters in the A* implementation and reducing the planning time around obstaclese.

- **Atharv Pulapaka**: Atharv tuned the cruise control module to deal with noisy pose. He is working on handling the behavior of the cars in the intersection region of the track and improving the MPC controller to follow the planned path closely.

4 Plans

Moving ahead, my individual goals are:

- Investigating the pose latching issue.
- Testing the robustness of Aruco and Outersense marker detection modules.
- Adding a smoother to improve the pose in the handover region of two cameras.

As a team our goals are:

- Test the full system for robustness and identify failure points of the new planning and cruise control module.
- Update track configuration to handle static obstacles and accommodate the staging behaviour of the data-association pipeline.
- Incorporate safety behaviours in the cars to deal better with communication loss and high latency cases.