Automated Driving Using External Perception

Individual Lab Report - ILR04
March 22, 2023

Team E - Outersense

Author:
Jash Shah

Team Members:

Atharv Pulapaka
Dhanesh Pamnani
Jash Shah
Ronit Hire
Shreyas Jha

Contents

1 Individual Progress

For the MRSD project, I have been working majorly on two major fronts, the perception stack as well as the manufacturing of the track setup.

1.1 Perception

As for the perception domain, I mainly worked on two fronts. The first was camera calibration and the second was testing the robustness of our algorithm from multiple different heights.

1.1.1 Camera Calibration

In the previous progress review, we were able to get intrinsic camera matrices. However, the camera calibration for extrinsics was still pending. Extrinsic parameters map the world frame to the camera frame. This time, I was able to also retrieve the extrinsic parameters which would allow us to map any point in the world to the camera frame. Then using intrinsics, we can map the camera frame to the image frame and then to the pixel frame. The camera instrinsics are written to a .json file and the extrinsics are being printed such that they can be used for further calculations. Furthermore, I also created a package that can be quickly used for calibration whenever required. This is important for our project because we need to continuously re-calibrate the extrinsics every time we start our system. We need to do this because it is possible that the infrastructures move slightly between runs and our software needs to be able to account for this. As can be seen in Figure 1, the Github link to this repo can be seen.

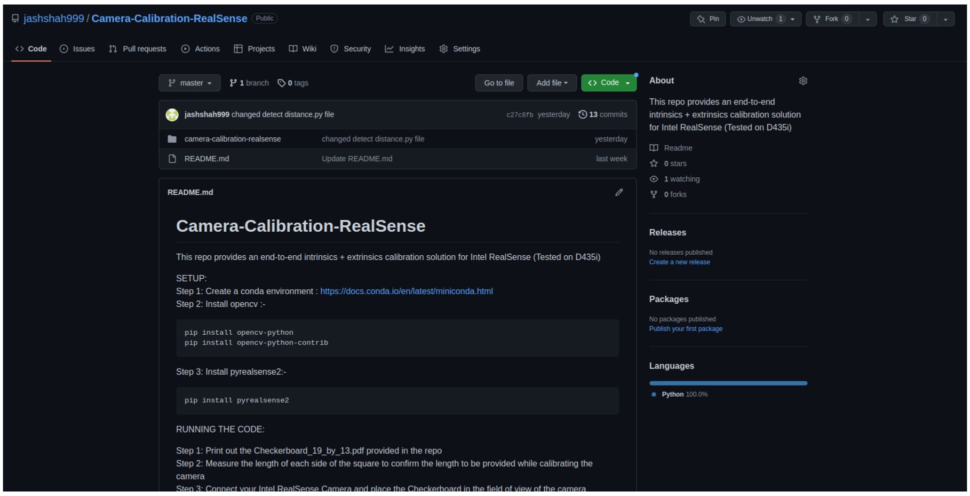

Figure 1: Github repo for end-to-end camera calibration of Intel realsense

1.1.2 Deviation from track algorithm and testing

As for the deviation from the track, I developed a simple geometric algorithm to find the distance between the center of the marker and the line joining the two ArUco markers. The results can be seen in Figure 2. The yellow line denotes the perpendicular distance between the center of the colored identifier and the line joining the two AruCo centers. This elucidates the deviation from one end of the track which is an important metric for us. Furthermore, the yaw estimate was made more robust from the last progress review and is providing more accurate results which is useful for the controls block.

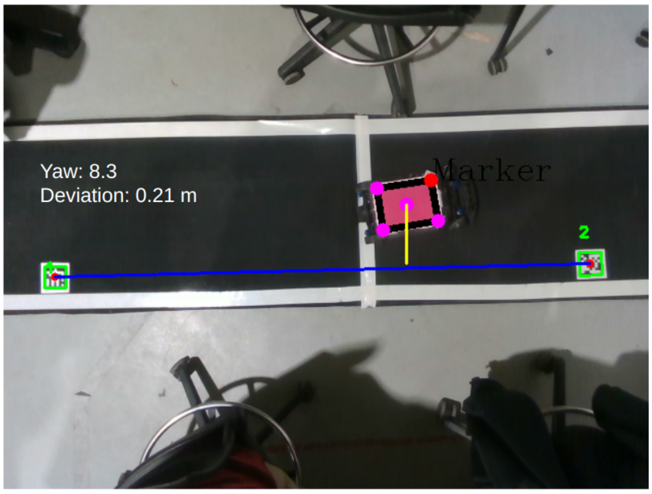

Figure 2: Deviation of RC car from track

Throughout the last few progress reviews, I have been testing my algorithm from a height of 2 meters. However, because for our final SVD, we will be testing from a height of 2.8 meters, I started testing the same algorithm from a height of 2.8 meters. From this height, there were a few issues with the same marker size. Hence, we mitigated this risk by increasing the marker size. Increasing the marker size improved the stability of detection and tracking. I tried keeping the size of the ArUco markers the same (75mm), which worked to an extent. However, to improve the stability of the detection, I switched to using 100m markers which worked better.

1.2 Track setup

As for the track setup, the work has been completed as of now. All the infrastructure units are up and ready. As can be seen in Figure 3, all units are up along with the mounts. The only facet remaining for this is the edge-compute module which needs to be mounted when it has been decided upon.

Figure 3: Infrastructure units ready with camera mounts

2 Challenges

2.1 Perception

In the perception subsystem, a major issue I faced was the flicker of the identifier under different lighting conditions and sizes. From the higher camera mount, I found that the same identifier and ArUco marker did not perform in the expected fashion. There was a fair bit of flicker from this height. To counter this, I increased the size of the ArUco markers as well as the identifier (scaled to the height).

Another major issue that I faced was with the calibration. Even though I was able to retrieve the intrinsics and extrinsics of the camera, there was no way of verifying whether these matrices were correct. Currently, we are still trying to corroborate these results.

2.2 Manufacturing

One of the significant challenges in manufacturing infrastructure units is the occurrence of vibrations due to the height of the structure, which may worsen when the cantilever arm and Intel Realsense are installed. To minimize this risk, we could incorporate weights such as sandbags to the base of the units to reduce vibration. Additionally, we are currently working on developing sturdy mounts for Intel Realsense to prevent camera shake and ensure accurate detection of identifiers and fixed markers on the cantilever.

3 Teamwork

As for teamwork, each of us is flexible and even though we have our own verticals, our team structure allows us to collaborate with each other to complete high-priority tasks faster.

- **Ronit Hire**: Ronit has been looking after the perception pipeline. He is currently working on integrating all the code for the perception subsystem. Moreover, he also worked on the PID under the controls subsystem. Further, Ronit is also looking after the logistics and budget of the team.
- **Shreyas Jha**: Shreyas worked on the multi-machine ROS architecture. He was able to control the speed and steering of the RC car from a different machine. Moreover, he set up the local wireless LAN system over which the data would be sent.
- **Dhanesh Pamnani**: Dhanesh mainly worked on the hardware aspect of the track and infrastructure. Because there were a lot of vibrations, Dhanesh worked on reducing these by stabilizing the base of the units. Moreover, Dhanesh also fixed the issues that had occurred during manufacturing. One of the infrastructure units was tilted which took time to fix. Further, he also collaborated with me to work on the calibration of the cameras. He is currently working on corroborating the results I got from the RealSense calibration.
- **Atharv Pulapaka**: Atharv worked on generating acceleration and steering profile for the RC vehicle. Moreover, he also ported all the Python code to ROS. This helped us make the code more modular and interface it with the perception block. Lastly, he also worked on Hardware in the Loop Simulation.

4 Future work

4.1 Personal

On a personal front, I will focus on the perception front. Mainly, we will be working on integrating all the perception subsystems. The perception node will provide x, y, yaw, velocity, and lane deviation. The controls block will subscribe to this data and use it as an input to MPC.

1. Explore new methods of feature detection that do not depend on lighting conditions, i.e. Haar cascaded classifier or SIFT
2. Be able to publish x, y. yaw. velocity, and lane deviation to the controls block
3. Corroborate calibration results with hand-eye methods or computing reprojection error accurately

4.2 Team

As for the future work of the team, we wish to achieve the following goals.

1. Perception block is able to detect pose of moving car
2. Fuse data from multiple infrastructure sensors
3. Display pose estimate from edge node on central server
4. RC car tracks motion cues sent from the central decision system
5. Software based E-stop and safe stop of vehicle on loss of communication