Automated Driving Using External Perception

Individual Lab Report - ILR05

April 6, 2023

Team E - Outersense

Author:

Dhanesh Pamnani

Team Members:

Atharv Pulapaka
Ronit Hire
Jash Shah
Shreyas Jha

Carnegie Mellon University

Contents

1 Individual Progress

1.1 Lane detection

As the infrastructure and the track are ready and most of the mechanical work of the system is completed I worked on the perception stack starting with the Lane detection. I used 2 methods to detect the lane the first was a Sliding window method and the second was using canny edge detection and then extracting the features to obtain the lane. The main challenge was to have a detection which did not assume that the entire tract was within the FOV of the camera and that the lane was not directly below the camera. The sliding window approach starts with cropping the image and creating a mask which has just the track and not the surroundings. It then takes the sum of all the pixels along the width of the image to obtain the histogram. The max values of the histogram on either side of the center of the track then gives the starting point of the respective lane. The lane detection starts from the center and then goes to either side of the image frame. The sliding window allowed the detection to be robust and also allowed to have curves and different angels in the track. After the lanes were detected the coordinates were used to obtain the midpoint of the lane and then an equation was fit to obtain the equation of the track. The pixel coordinates were then converted to world frame and given to the perception system. The Fig 1 below shows the output of the lane detection and the next section talks about the transforms of converting pixel coordinates to world frame.

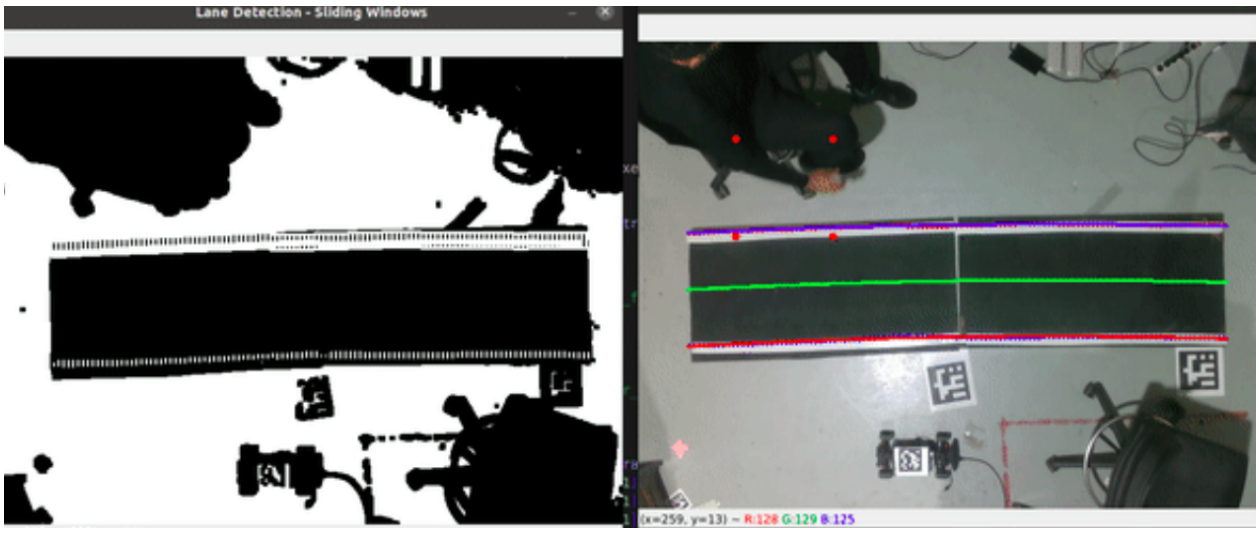

Figure 1: Lane detection using Sliding Window

1.2 Accurate transforms between image frame to world frame

Secondly, I worked on implementing accurate transforms from the image frame to the world frame. This was used for handover between the 2 cameras and to get the accurate position of the RC car and hence the velocity. Getting accurate values in the world frame is extremely important for us as the controls of the car is written in the world frame and he performance requirements set by the team are also in the world coordinates. i implemented the inbuilt Realsense function to get the transform however this gave very inaccurate results for our scale. The challenges section below describes the method I used to overcome this and the final result. This was then used for handover and the difference can be seen in the Figure 2 below. This figure shows the difference

in the estimated position of the RC car from 2 cameras. The error is almost 5cm still and we will be working on making this better by calibrating the cameras again and tuning the deprojections of the 2 cameras from the color to the depth frame.

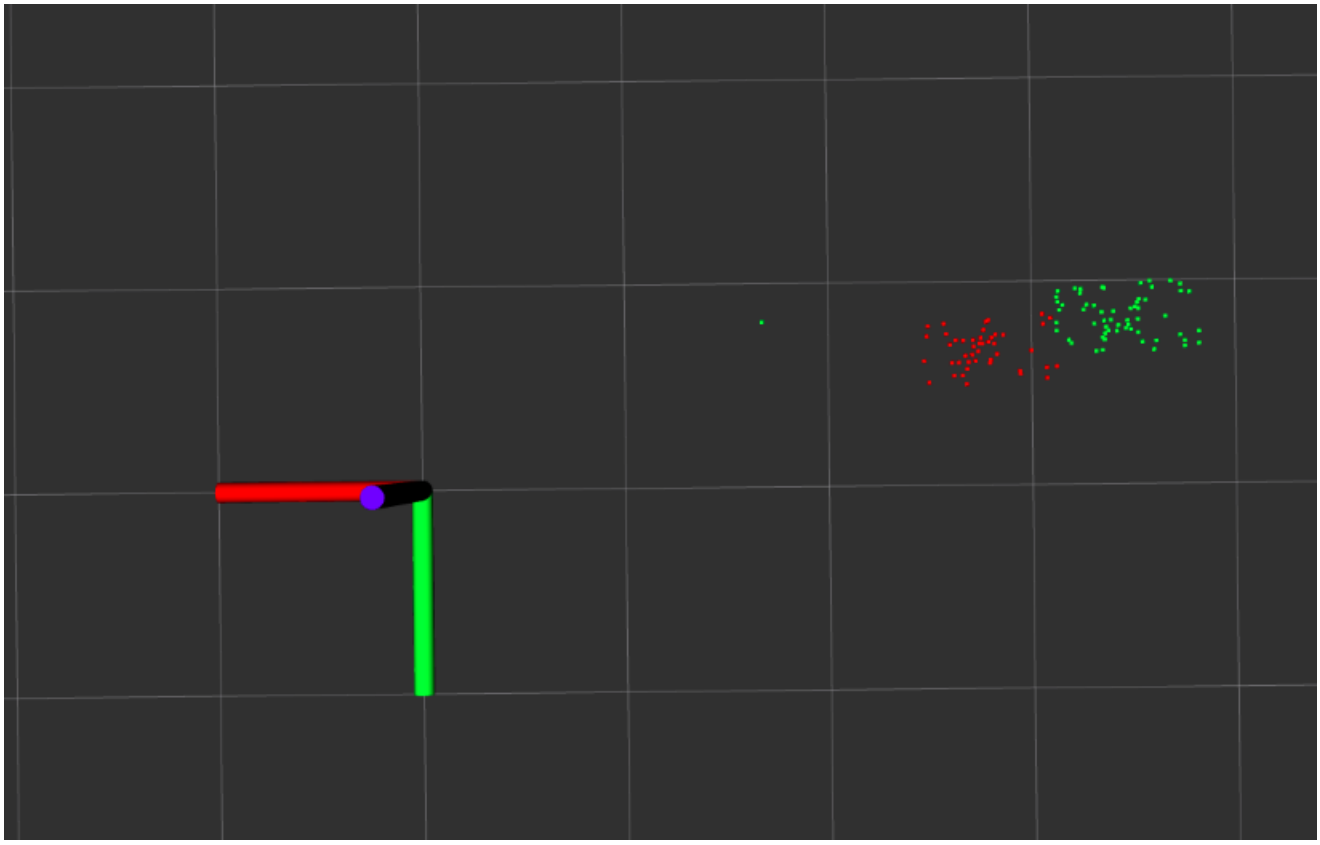

Figure 2: Transforms between 2 cameras

1.3 Mechanical changes in the RC car

I also worked on getting the RC car ready for the integration tests. This involves the Mechanical upgrades of the car, the procurement, the wire management and other factors. I designed 3D printed structures on which the electrical components could be housed. The structure as shown in Fig 3 below, had 2 levels, the first was were the VESC and the Rpi were placed. The motor battery and power bank (which powered the Rpi) were placed on the bottom level. The top level is where the marker for detection will be placed for perception. The center of the marker gives an estimate of where the car is, a mask is then applied which gives a bounding box of the car. The key points are then detected within this region to obtain interest points on the car for perception. I also worked on rectifying certain camber problems in the RC car wheels which had come about after an accident in which the car ramped into a wall at high speed. Finally we can say taht we have 1 car ready for the SVD and will be working on the spare vehicle for risk management.

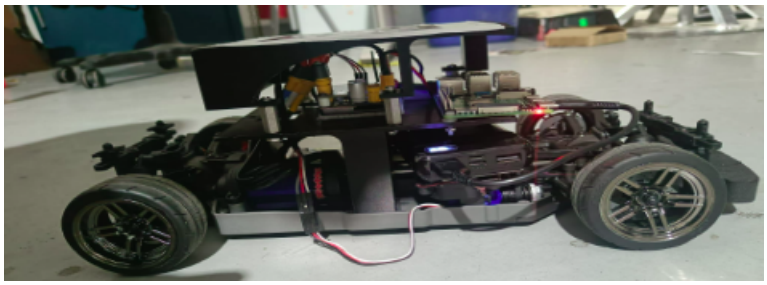

Figure 3: Mechanical setup of the ready RC car

1.4 Basic Implementation of detection and velocity

Lastly, I also worked on the basic implementation of the perception system. This was to integrate the data from the camera to the control system and check for communication delays. I implemented the detection of the vehicle based on the colored marker on it. This was a HSV based implementation which was very sensitive to the lighting conditions. Further, I detected velocity of the car by detecting the 4 corners of the HSV marker in every frame. The angle or pose of the vehicle was obtained by getting the angle of the HSV marker from a datum obtained by aruco set on the track. Essentially it was the entire perception system but a very unreliable ”dirty” implementation. It helped us check for communication delays while Ronit worked on getting the velocity from Swift features and Jash worked on car detection based on rotation invariant template matching.

2 Challenges

2.1 Lane Detection

I started the lane detectio with the sliding window approach which assumed that the whole track will be in the FOV of the camera and the track would be directly below the camera. These assumptions made the system very unreliable under different conditions. The first step was to crop the frame by using a mask for just the track and not the entire frame. this allowed me to focus the detection on the region that has the track if the whole track is not in the frame. Next challenge was making sure that the detection does not detect any parts of the car and only detects the track even if the track was curved or had edges in different angels. The sliding window approach helped in this. I also implemented edge detection for the lane detection in the case where there are more than one lane.

2.2 Accurate transforms between image frame to world frame

Calibrating the Realsense RGB camera leads to various errors. This makes handover very challenging and also getting the accurate transform between the image and the world coordinates difficult. The inbuilt method by realsense for this gives an error of around 5cm. This is a lot for the scale we are working at. To overcome this I wrote a function that de-projects the pixel from the image frame to the real world and re-projects it in the 2 stereo cameras in the depth frame. It then runs a matching algorithm to find the pixel in the depth frame that is closest to the pixel in the RGB color frame. The depth is then obtained by getting the BufData and the depth scale. The depth along with the extrinsic and intrinsic are used to then obtain accurate X

and Y coordinates of the desired point in the color frame. This reduced the error to within 1 cm . However the function sometimes gets stuck in the while loop which acts as a blocker and that needs to be worked on.

3 Team Work

This section talks about the work the team members have been doing in the project.

- **Jash Shah**: Jash worked on Aruco marker detection and making the identification of the car more robust by shifting from hsv based detection to rotation invariant template matching. He tested the implementation from different heights and for different lighting conditions. He also worked on the embedded system and setting up ros and OpenCV on the various embedded systems.

- **Shreyas Jha**: Shreyas worked on getting the RC car ready for the integration test. He replaced the ESC with the VESC. The VESC allowed him to control the car below 80cm/sec. He then integrated the RC car with control inputs and implemented npt on the RC car.

- **Ronit Hire**: Ronit worked on the perception system. He was mainly responsible for the velocity and pose estimation of the car through perception. He also worked on the handover with me and on developing accurate transforms between the 2 cameras. Further Ronit worked on trying to implement the perception stack on the embedded system but faced a lot of challenges in running the stack on the embedded systems which he has highlighted in his ILR. He has also worked on getting the stack ready for the integration test of the system

- **Atharv Pulapaka**: Atharv has been working on the PCB for the RC car along with other factors. He worked on writing a different node for the plotting as running on the same road was making the controls slow because of matplotlib. He also worked on making the controls more robust and on the integration testing of the system with the RC car and on the integration test of the perception with the Controls.

4 Plans

The path forward for the team focuses on integration testing and preparation for the SVD

- **My Plans**: The next step for me is to work on the integration and testing of the various sub-systems. This is in accordance with what the team is targeting too as we head towards the SVD. We will be integrating the controls with the RC cars and integrate the perception with the controls. We will also be improving the handover between the 2 cameras and the visualization in Rviz. Lastly we will be working on the SVD presentation and the demonstration. We will start by testing the system outside the Navlab in the lighting conditions of the basement. For this we will start with setting up the track and the infrastructure units at the desired position. This will be followed by the RC car setup along with making a spare of the car. Lastly we will be testing all the units on their own before we can begin the integration test of the system and the SVD preparation.