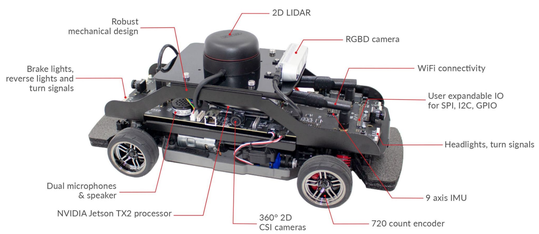
Real Time Lane Detection and Lane Following Algorithms for Autonomous Driving with Quanser QCar

Khush Lalchandani[1], Aarav Paryemalani[2], Gideon White[3]
[1]Liberal Arts and Science Academy, [2]BASIS San Antonio, [3]University of Texas at Austin College of Natural Sciences

Introduction

The development of autonomous vehicles (AVs) has advanced significantly due to progress in sensor technology, control systems, and computational power. Autonomous vehicles aim to improve road safety and reduce traffic congestion. Lane detection and lane assistance technology are important to ensure a vehicle can maintain its position and stay in its lane [1].

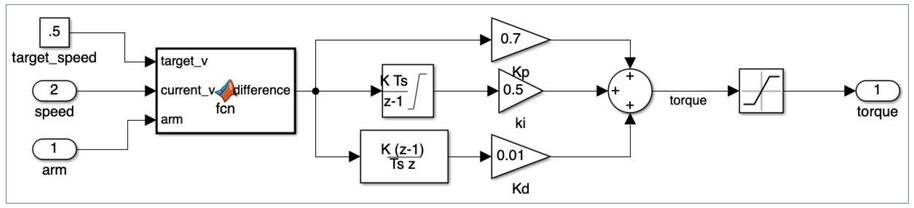
Current approaches in lane detection are categorized into model-based and feature-based methods. Model-based methods use predefined lane models, such as vanishing point-constrained methods or hyperbola-pair models and fit them to the lane line [2]. While very accurate, these models can be computationally intensive. Feature-based methods extract lane features such as color or edges from camera images. Techniques like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) are used to handle varying shapes and noise in the data, to improve lane detection [3]. For lane following, algorithms like Pure Pursuit, Stanley, and Model Predictive Control (MPC) are commonly used. Pure Pursuit is used for its simplicity in low-speed scenarios, whereas Stanley and MPC are preferred for higher speeds and complex maneuvers. This study integrates these methods with speed control techniques to improve real-time lane detection and following capabilities on the Quanser QCar.

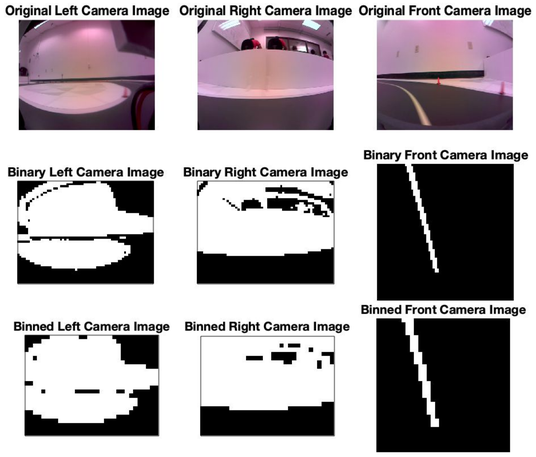


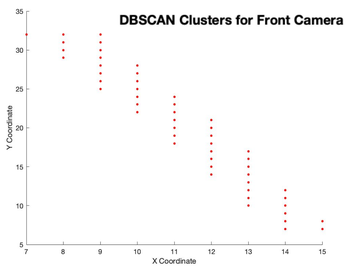
Research Goal

Our primary goal was to develop and implement an efficient algorithm for real-time lane detection and lane following using the Quanser QCar. We aimed to quickly process camera image data from the front and sides of the car to provide accurate steering commands and improve reliable lane detection and following.

Methods







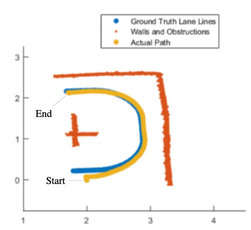
Speed control was managed with a PID controller, tuned via an ARX model trained on historical torque and speed data. The PID controller adjusted the vehicle's speed by calculating the error between the desired and actual speeds, applying proportional, integral, and derivative gains to minimize this error.

We used several image-processing techniques for improved lane detection. First, images were resized from 480x640 to 48x64 pixels to speed up processing. Then, the front camera feed was transformed into a top-down view using calibrated camera intrinsics and extrinsics in MATLAB.

A binary color mask was applied to isolate lane lines based on RGB values. The image was further reduced to 32x32 pixels through binning, where clusters of pixels were selected for their highest values. To detect lanes, we used the DBSCAN clustering algorithm. The cluster with the largest y-coordinate range that fit the cluster size restraints was selected as the lane line. A third-degree polynomial was then fitted to these lane line points for smooth lane following.

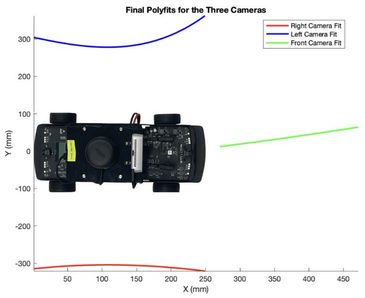
Side cameras provided additional information on the vehicle's distance to the lane edges. We detected the lowest y-coordinate in the side camera images to determine the distance to the lane edge, applying steering corrections dependent on the distance away from the edge. For steering control, we implemented the Pure Pursuit algorithm, using the polynomial-fitted lane line as the reference path. The lookahead distance was optimized to balance stability and accuracy.

Results



The lane-following algorithm for the QCar demonstrated high accuracy, deviating from the lane line by a maximum of 5 centimeters. The vehicle's speed was consistent and reached our target velocity of 0.5 m/s. We spent the majority of our time troubleshooting the process for the birds-eye view. Some problems we encountered included the birds-eye images being initially warped due to poor calibrations, incorrect measurement units, and broken camera extrinsics (MathWorks). Finally, we discovered that the processing time of the birds-eye function on the full size image was leading to lag and the car was receiving improper steering commands. By resizing the camera input before the birds eye function was called, the average processing time for each image went from 0.4 seconds to .01 seconds. This led to the QCar being able to properly correct its course to follow the center lane line on the Quanser track.

Conclusion



Our research demonstrated effective techniques for lane detection in autonomous vehicles. First, it was crucial to resize the QCar camera input to be small enough for our algorithm to process the image quickly. Our image was then converted from a fish-eye perspective to a birds-eye view and a binary mask of this birds-eye view was created. Clustering of the resulting points using a DBSCAN algorithm was completed at an average of 0.01 seconds per image. A pure pursuit algorithm used these clustered points to create and follow a line modeled after the lane. The final output of a steering angle was able to match the real-time location of the QCar and enabled it to follow the lane lines.

Acknowledgments

Thanks to the National Science Foundation and the Mobility Systems Lab at The University of Texas at Austin. Under the mentorship of Dr. Junmin Wang, Yung-Chi Kung, Robin Sambuis, and Ammaar Khatlani, we developed invaluable skills for our lives and future careers. We would also like to thank Shante Tyus for her coordination with the university and our fellow cohort members Shreehan Pippalla and Aayan Ghani.

References

[1] Tao, J., Shin, BS., Klette, R. (2013). Wrong Roadway Detection for Multi-lane Roads. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds) Computer Analysis of Images and Patterns. CAIP 2013. Lecture Notes in Computer Science, vol 8048. Springer, Berlin, Heidelberg.
[2] Jianwen Wang and Xiangjing An, "A multi-step curved lane detection algorithm based on hyperbola-pair model," 2010 IEEE International Conference on Automation and Logistics, Hong Kong, China, 2010, pp. 132-137, doi: 10.1109/ICAL.2010.5585398.
[3] K. Khan, S. U. Rehman, K. Aziz, S. Fong and S. Sarasvady, "DBSCAN: Past, present and future," The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Bangalore, India, 2014, pp. 232-238, doi: 10.1109/ICADIWT.2014.6814687.