Functional Specification

ALVN

Autonomous LiDAR and Vision based Navigator

By

Ciaran Maye

C00253212

Ollscoil Teicneolaíochta an Oirdheiscirt

South East Technological University

Table of Contents

Overview

This project aims to develop and examine various autonomous control systems for a ground vehicle using an onboard camera, lidar sensor and various algorithms. This project is intended as a proof of concept for the feasibility of fully autonomous vehicles.

Functionality

Lane Keep Assist System

The Lane Keep Assist System will utilize the OpenCV library to maintain the vehicle's position between two lane lines. The high-level plan for this system is as follows:

Acquire camera input: Capture input from the camera and assign each frame to a thread to be processed individually.

Region of interest: Define a region of interest in the image to focus on the lane lines.

Preprocess the image: Apply Gaussian blur using cv2.GaussianBlur, and detect edges using the Canny edge detection method from OpenCV (cv2.Canny).

Detect lines: Use the HoughLinesP method in OpenCV (cv2.HoughLinesP) to detect lines in the thresholded image.

Determine lanes: Combine lines into lanes and separate them into left and right lanes.

Average slope intercept: Calculate the average slope and intercept of the left and right lane lines separately.

Determine line points: Based on the calculated slope and intercept, compute the coordinates of two points on each line (left and right lane lines).

Calculate the lane line center: Determine the center point between the left and right lane lines.

Calculate steering angle: Utilize the lane line center to compute the steering angle based on the vehicle's position relative to the center point.

Smooth the steering angle: Implement a buffer to store a fixed number of previous steering angles and calculate the average of the stored angles.

Control the vehicle: Steer the vehicle's front wheels according to the smoothed steering angle.

Path Following System

The Path-Following System will use the OpenCV library to maintain the vehicles position by tracking a single lane line. The high-level plan for this system is as follows:

Acquire camera input: Capture input from the camera and assign each frame to a thread to be processed individually.

Region of interest: Define a region of interest in the image to focus on the lane lines.

Preprocess the image: Apply Gaussian blur using cv2.GaussianBlur, and detect edges using the Canny edge detection method from OpenCV (cv2.Canny).

Detect lines: Use the HoughLinesP method in OpenCV (cv2.HoughLinesP) to detect lines in the thresholded image.

Average slope intercept: Calculate the average slope and intercept of the left and right lane lines separately.

Determine line points: Based on the calculated slope and intercept, compute the coordinates of two points on each line (left and right lane lines).

Calculate the lane line center: Determine the center point between the left and right lane lines.

Calculate steering angle: Utilize the lane line center to compute the steering angle based on the vehicle's position relative to the center point.

Smooth the steering angle: Implement a buffer to store a fixed number of previous steering angles and calculate the average of the stored angles.

Control the vehicle: Steer the vehicle's front wheels according to the smoothed steering angle.

Lidar Obstacle Avoidance

The Obstacle Avoidance System will use a 2D Lidar sensor. This system is designed to function independently of other systems and serves as a proof of concept for the effectiveness of such systems, with the intention of eventually combining it with a lane keep assist system or path following system. The key aspects of this system are as follows:

1. **Initialize the system:** Set up the Lidar sensor, establish communication, and configure the vehicle's initial steering angle and speed parameters.
2. **Continuously monitor Lidar input:** Process and analyze data from the Lidar sensor to identify obstacles and their distances from the vehicle.
3. **Divide the Lidar field of view into sectors:** Create a specified number of sectors to simplify the analysis of the Lidar data and enable the system to focus on relevant areas.
4. **Determine the closest obstacle:** Identify the sector with the closest detected obstacle and calculate the steering angle required to avoid the obstacle.
5. **Steer the vehicle:** Adjust the vehicle's steering angle based on the calculated angle, incorporating a smooth transition to prevent abrupt changes.
6. **Detect imminent collisions:** If an obstacle is too close to the vehicle, stop the vehicle and perform a turn-around maneuver to change its course.

Target Market/Intended Use

This project primarily aims to demonstrate the potential of low-powered computing devices in the fields of Artificial Intelligence (AI) and Machine Learning (ML). By showcasing their effectiveness in a complex application such as autonomous navigation, it highlights the possibilities for these devices to be employed in various other domains beyond self-driving cars.

For instance, the lane-following and path-following systems developed in this project could be adapted for use in personal mobility devices, such as mobility scooters or electric wheelchairs. By implementing such technology, these devices could offer improved safety and convenience for users with limited mobility, enabling them to navigate more easily and with greater autonomy in their daily lives.

Furthermore, the warehouse and logistics industries could greatly benefit from the integration of low-powered AI and ML systems. Autonomous guided vehicles (AGVs) and autonomous mobile robots (AMRs) are already being used to streamline operations and increase efficiency in these settings. By leveraging low-powered devices, the cost**,** and energy consumption of these systems can be reduced, making them more accessible to small and medium-sized businesses.

Another potential application is in agriculture, where low-powered AI and ML systems could be integrated into farming equipment to perform tasks such as crop monitoring, precision spraying, or autonomous harvesting. This could lead to more efficient and sustainable farming practices, with a lower environmental impact.

Similar Projects

Nvidia Dave2

The Nvidia Dave2 project (Bojarski et al.) was created to investigate the effectiveness of convolutional neural networks in learning to drive, which entails detecting lane lines and adjusting steering angles appropriately as well as identifying landmarks, road markings, and signs. They concluded that it was quite effective, and not only that, but also effective in varied weather and lighting circumstances, using both a simulator and a real car.

DeepPiCarMicro

DeepPiCarMicro (Bechtel et al.) is a project from the University of Kansas, USA. This project was a case study from August 2022 with the goal of recreating the Nvidia Dave2 project on a smaller scale, using a Raspberry Pi Pico as the main component. This project is currently in pre-print, but the authors have reported that it was largely successful. Shared resource contention is a problem that needs to be considered to maintain good real-time performance on a relatively small, embedded device, according to a concern that was brought up throughout this project.

Bibliography

Bojarski, Mariusz, et al. "End to End Learning for Self-Driving Cars." *ArXiv.org*, 2016, https://doi.org/10.48550/arXiv.1604.07316.