Virtual Rail Autonomous Vehicle

19.02.2024

—

Lior Kozlov
Denis Kharenko
Tahel Karavani

Project Overview: Autonomous Line-Following Car

This autonomous virtual rail-following car project is designed to navigate predefined paths with the assistance of a computer vision algorithm. The vehicle's core system is dependent on a camera mounted on the car's chassis, which captures real-time images of the floor. These images are then processed using vision algorithms to detect and track the designated line along the path.

To interact with the vehicle, a web app has been developed, allowing users to instruct the car to travel from one point (point A) to another (point B). The web app is equipped with a server that employs a pathfinding algorithm to calculate the optimal route for the vehicle. This server acts as a central hub for receiving telemetry data sent by the car and to send navigation commands to the car.

The car communicates with the server by sending telemetry data, providing real-time information about its position, speed, and other relevant metrics. This data is then utilized by the web app to display detailed statistics and information about the vehicle's journey.

One of the key features of this project is the implementation of an algorithm that enables the vehicle to stay within the predefined path. This algorithm plays a crucial role in ensuring the car's accuracy and reliability in following the designated route. Overall, the combination of computer vision, pathfinding algorithms, and real-time telemetry creates a robust and efficient system for autonomous rail following.

Key Components

Our project consists of three main components: the vehicle with a computation module and a camera, a server, and a web application. Now, let's delve into the details of each of these sub-components.

Server (in page 3)

- Server path finding algorithm
- API for communication
- Telemetry management and storage

Vehicle computation module and camera(in page 5)

- Esp32 camera
- Line detection algorithm pipeline
- Rail following algorithm

Web application (in page 7)

- Road Map Display
- User-Initiated Route Start and End Points
- Transmitting to Server Route Start and End Points
- Telemetry and Statistics Display

Server subcomponents

Server path finding algorithm

The server employs a sophisticated data structure to store the graph representing our virtual rail network, where junctions serve as pivotal points. Users submit navigation requests through the web app, specifying the origin (point A) and destination (point B) for their vehicles.

Utilizing a shortest path algorithm, the server computes the most efficient route within the virtual rail network for the autonomous vehicle. The outcome is a set of instructions that the vehicle can subsequently request from the server, initiating its navigation along the rails.

This algorithm is designed to make optimal decisions at junctions, ensuring the vehicle reaches its destination in the shortest possible time.

API for communication with web application and autonomous vehicle

The server functions as an API intermediary facilitating communication between the web application and the autonomous vehicle. It offers specific routes to enable seamless interaction between these components.

Firstly, the server provides a route through which the vehicle can transmit responses to request a set of instructions for navigation along the virtual rail network. This ensures efficient communication and coordination between the server and the autonomous vehicle.

Additionally, the server includes a route dedicated to receiving telemetry data from the vehicle. This serves as a crucial channel for the continuous exchange of real-time information, contributing to effective monitoring and control.

For user-initiated actions, the server features a route accessible to the web application. Through this route, the web app can submit requests for navigation from point A to point B. The server, in response, initiates the utilization of a path-finding algorithm and stores the generated set of instructions. This strategic approach anticipates the vehicle's subsequent request for navigation guidance.

Furthermore, the server accommodates requests from the web app to retrieve all pertinent telemetry data submitted by the vehicle. This ensures that the web application has access to comprehensive and up-to-date information, enhancing the overall user experience.

Telemetry management and storage

Autonomous vehicles will transmit telemetry data to the server through designated routes. The server will receive, format, and store this information in a database, ensuring effective data management and reliability for future reference.

The server is equipped to distinguish between various types of telemetry and information sent by the vehicle. This capability is crucial as it enables the web application to request specific telemetry data, allowing for the display of insightful statistics and comprehensive information regarding the vehicle's performance.

Vehicle computation module and camera subcomponent

Esp 32 camera

Positioned to capture images of the floor, the ESP32 camera module is a crucial component of our autonomous vehicle project. Mounted on the vehicle's chassis and directed towards the floor, the camera captures real-time images of the floor surface, specifically focusing on the designated rail or path that the vehicle needs to follow.

These floor images serve as the primary data source for subsequent stages of image processing and analysis, guiding the vehicle's navigation along the predefined route.

Line detection algorithm pipeline

At the core of our computer vision system lies the line detection algorithm pipeline, tailored to process the images of the floor captured by our camera. This pipeline consists of interconnected stages designed to analyze the floor images and extract relevant features related to the rail or path.

From color space conversion to thresholding, edge detection, and subsequent analysis, each stage plays a vital role in identifying key elements within the captured images, facilitating accurate navigation along the ground surface.

Rail following algorithm

Serving as the cognitive framework for our vehicle's navigation system, the rail following algorithm utilizes insights derived from the floor imagery processed by the line detection algorithm pipeline. This algorithm interprets the detected features and generates refined control commands to steer and guide the vehicle along the predefined path.

By using techniques such as analyzing the distance from the center point of the image to the detected edges on both sides, the rail following algorithm ensures that the vehicle maintains alignment and trajectory, enabling seamless navigation along our designed path.

Web application

Road Map Display

The Road Map Display feature is crafted to provide users with a visually captivating interface for navigating routes on the map. Through this feature, users can effortlessly choose their desired starting and ending points directly on the map interface. This intuitive approach streamlines the route selection process, empowering users with the freedom to explore and customize their journey visually. By integrating interactive elements and clear visualization, the Road Map Display ensures a seamless and immersive experience, facilitating efficient route planning and enhancing overall user satisfaction.

User-Initiated Route Start and End Points

With user-friendly input mechanisms and intuitive interface design, users can easily specify their starting and ending locations. Clear prompts for reducing confusion and ensuring efficient route selection. This thoughtful approach prioritizes user experience, facilitating smooth and hassle-free route planning for enhanced usability and satisfaction.

Transmitting to Server Route Start and End Points

Facilitating seamless communication between the user interface and server backend, the Transmitting to Server Route Start and End Points feature ensures reliable data transmission for route processing. Error handling and validation procedures safeguard against data corruption or loss, maintaining data integrity throughout the transmission process. By offloading route processing tasks to the server, this feature optimizes system performance and scalability, enabling efficient handling of user requests and enhancing overall application responsiveness.

Telemetry and Statistics Display

Providing users with valuable insights into vehicle performance and navigation metrics, the Telemetry and Statistics Display feature offers a comprehensive overview of key data points. Through intuitive visualizations and customizable display options, users can monitor real-time telemetry data such as vehicle stability. Statistical analyses and trend visualizations enable users to track performance trends over time, identifying areas for improvement and optimization. By presenting actionable information in a clear and concise manner, this feature enhances user understanding and decision-making, ultimately contributing to safer and more efficient navigation experiences.