Research Report

ALVN

Autonomous LiDAR and Vision based Navigator

By
Ciaran Maye
C00253212

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Table of Contents

Table of Figures

Abstract

This goal of this project is to develop a system for a vehicle to allow it to autonomously detect a 'Guideline' and its surroundings and navigate them accordingly. This project also aims to develop an 'Obstacle Avoidance System', by way of image processing, edge/line detection and LiDAR spatial mapping. This report summarizes the topics, methods, and technologies that have been investigated to find a solution for these systems.

Introduction

Autonomous vehicles, also known as self-driving vehicles, have the potential to revolutionise both personal and public transportation along with how we move goods and materials in a variety of settings. This project is intended to serve as a proof of concept for the potential amount of autonomy that can be achieved with a small-scale vehicle and limited sensors. Various fields and technologies have been investigated for the objectives of this project and are detailed below. Please keep in mind, as of this writing, these are the most relevant domains and technologies to this project, this list is not exhaustive.

Overview

What is an autonomous vehicle?

Autonomous vehicles are vehicles that are capable of sensing their environment and navigating without human input. They use a combination of sensors, cameras, and software to understand their surroundings and make decisions based on that information. Autonomous vehicles have many uses across a wide range of industries.

In a warehouse environment, autonomous vehicles are used to transport goods and materials from one location to another without the need for human assistance. They can be programmed to move along specific routes through the use of guidelines, stopping at various points to pick up and drop off goods. They can often communicate with other autonomous vehicles to coordinate their movements and avoid collisions, in conjunction with using path following and obstacle avoidance systems.

These technologies are also currently offered by many road vehicles on the market, including Adaptive Cruise Control (ACC) and different varieties of Lane Keep Assist Systems (LKAS) *(Kamerpower).* Adaptive cruise control uses Radar and LiDAR to detect and maintain a safe distance between lanes and from other vehicles. This feature has been offered from between 2012 and 2013. 'Lane Keep Assist' systems, a more recent development uses windshield mounted cameras to identify lane lines and then steer the car between the lanes. This is an extremely helpful tool for travelling on major roads such as motorways as it reduces fatigue.

There are different levels of autonomy for vehicles, as defined by the Society of Automotive Engineers (SAE). These range from Level 0, which has no automation, to Level 5, which is fully autonomous and requires no human intervention. Most current autonomous vehicles are at Level 2 or 3, which means they can handle some driving tasks, but still require human supervision*. ("SAE Levels of Driving AutomationTM Refined for Clarity and International Audience")*

Benefits of Autonomous Vehicles

Autonomous vehicles have brought significant changes to various industries, from commercial road transportation to warehousing and space exploration.

One of the primary advantages of autonomous road vehicles is the potential reduction in traffic accidents. In a world where all vehicles are autonomous, collisions could be nearly eliminated. Furthermore, the implementation of autonomous vehicles could lead to lower insurance costs due to fewer accidents and reduced fuel consumption and emissions, as artificial intelligence can drive more efficiently than humans. This efficiency translates to "reclaimed time," enabling individuals to engage in more productive or enjoyable activities during their commute. *(Abdallah)* Importantly, autonomous vehicles can also provide people with disabilities, such as blindness or tetraplegia, greater freedom, and independence by allowing AI to control the vehicle partially or entirely, depending on their ability.

In the warehousing and logistics industry, autonomous vehicles offer several benefits, such as enhanced productivity, improved safety, optimized space utilization, and reduced labor costs. *(Stefanini)* Automated Guided Vehicles (AGVs) and autonomous forklifts can work tirelessly, increasing efficiency in warehouses and distribution centers. Replacing human-operated vehicles with autonomous systems significantly reduces the risk of accidents and injuries, creating a safer work environment. Additionally, these vehicles can move more precisely and efficiently, allowing for better space utilization in storage facilities, ultimately reducing the need for manual labor and cutting costs for businesses.

The agriculture sector can also benefit from autonomous vehicles. Precision farming, enabled by advanced sensor-equipped vehicles, can perform tasks like planting, harvesting, and monitoring crops more accurately and efficiently than human-operated machinery. *(Vrchota, Pech and Švepešová, 2022)*. These vehicles can optimize resource usage, such as water, fertilizers, and pesticides, contributing to more sustainable and environmentally friendly farming practices. Moreover, the use of autonomous vehicles can protect agricultural workers from hazardous tasks and strenuous labor, reducing the risk of accidents and injuries.

In space exploration, autonomous vehicles like rovers can explore distant planets and celestial bodies without human intervention, reducing the risks and costs associated with manned missions. These vehicles can navigate and analyze complex terrains more effectively, enabling comprehensive data collection and a better understanding of extraterrestrial environments. Additionally, autonomous vehicles can operate for extended periods, allowing for more thorough and continuous exploration of space.

Despite these advantages, there are several concerns regarding the adoption of autonomous vehicles across various industries. Job displacement is a significant concern, as human-operated vehicles are replaced by automated systems, potentially leading to job losses in warehousing, logistics, and agriculture. The development and implementation of autonomous vehicle technology can be expensive, which may limit its adoption by smaller businesses or organizations with limited budgets. Technological limitations, such as adverse weather conditions, low-light environments, or complex terrains, could impact the effectiveness of autonomous vehicles in certain industries or applications. Security risks, particularly in critical industries like space exploration, make autonomous vehicles vulnerable to cyber-attacks or hacking, potentially compromising mission objectives or data breaches.

Ethical considerations arise as increased reliance on autonomous vehicles brings up questions about responsibility and accountability for accidents or unintended consequences, as well as the potential impact on human employment and well-being. Lastly, the adoption and integration of autonomous vehicles across industries may require new regulations and standards to ensure safe and responsible use, presenting challenges for businesses and policymakers.

Hardware

PiCar-V

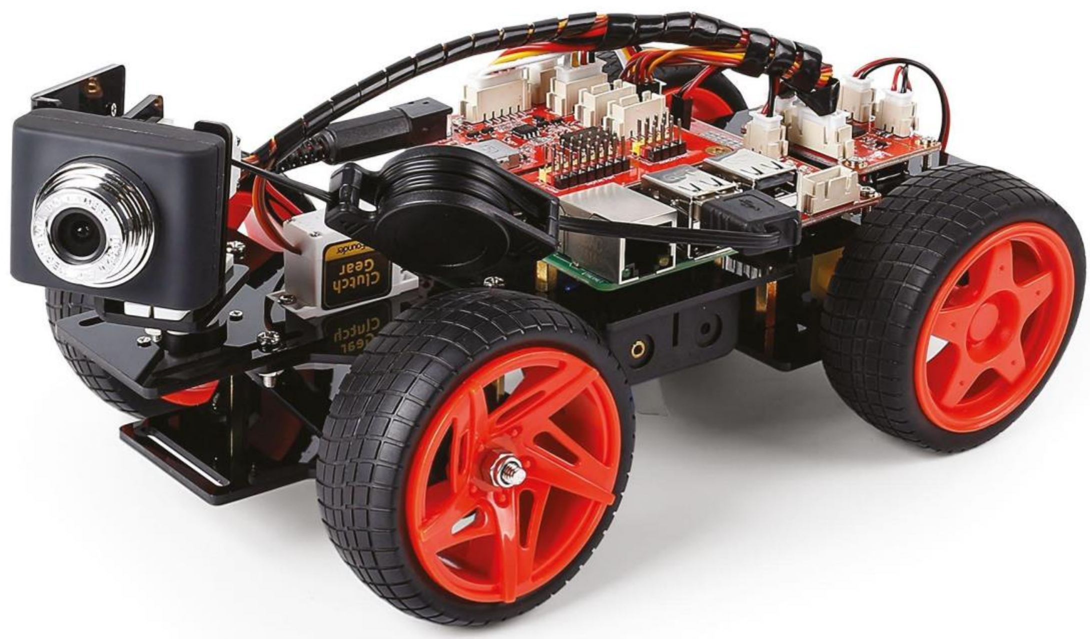

Figure 1: Sunfounder Picar-V

The vehicle used in this project is a combination of a Raspberry Pi 4B and a robotic car kit from the company 'Sunfounder' (*"SunFounder Focuses on STEAM Education with Open-Source Robots"),* known as the PiCar-V. This kit includes 3 servos (180° rotating, high precision motors), 3 separate PCB's, a battery mount, 2 front wheels, 2 back wheels, a 120° wide angle camera along with all the necessary mountings and parts to construct the car.

The raspberry Pi 4B uses a Quad core ARM based 1.5 GHZ processer, with 8GB of ram. The PI is renowned for its portable form factor, variety of ports given its size and relative power based on its form factor. It has two USB 3.0 ports, two micro-HDMI ports, a two lane MIPI DSI display port, a two lane MIPI CSI camera port, stereo audio and composite video port, a micro-SD card slot, a USB-C power port and an ethernet port.

RPLidar

Figure 2: Slamtec RPLidar A2M8

The Lidar sensor used for this project is a RPLidar A2M8, produced by the company "Slamtec" *(Huang).* This sensor is a 2D Lidar sensor with 360° of rotation. The sensor has a max range of 16 meters. Slamtec provide an SDK to interface with the sensor, this SDK is intended for use with C++. As detailed below there are community supported python implementations of this SDK which will be used for this project.

In order to complete this project, it will be necessary to attach the Lidar sensor to the Picar. To do this, a mount specifically for the Picar will be designed and 3D printed. The software program for doing this is detailed below.

Software and Technologies

Operating Systems

For the purposes of this project, various operating systems were considered, with power consumption, resource consumption and available tools being considered.

Ubuntu

Ubuntu (“Enterprise Open Source and Linux | Ubuntu”) is a Linux distribution based on Debian. It's well suited for cloud computing, running servers and desktop computers.. Ubuntu's interface is based on Linux desktop environments. GNOME has been the default since Ubuntu 17.10.

Ubuntu is open-source software that is constantly checked and reviewed by its users. As a result, any security flaws can be rapidly recognized and remedied. When compared to other operating systems.

Ubuntu is not a resource-intensive operating system; it runs nicely on low-end hardware. The gnome desktop environment requires just under 1 GB of RAM to function.

Robot OS

The Robot Operating System (ROS) *(“ROS.org | Powering the World’s Robots”)* is a framework and set of tools that install on top of Ubuntu to provide the functionality of a single operating system on a distributed computer network. Its primary applications are for robotics however most of the tools provided are geared toward working with external device of all types.

ROS provides hardware abstraction, device drivers, communication between processes across many machines, testing and visualization tools, and more features to facilitate the development of software for machines.

ROS is a collection of resources, tools, and packages that acts as a common ground for all developers. For example, if someone has created a program that interacts with a driver or performs a specific algorithm, ROS provides an interface that enables other programmers to utilise it. ROS supports all programming languages, meaning you can communicate with nodes that are written in any language. ROS enables the connection of a network of nodes to one central hub. Nodes can run on several devices and all of them can communicate with the hub in different ways.

The basic methods of establishing the network are to provide requestable services or to establish publisher/subscriber links with other nodes. Both techniques communicate using the same message types.

As ROS is less an entire operating system and more of a feature pack for Ubuntu installations, the resource usage is the same as Ubuntu.

While the features of ROS may be useful for this project, the Software Development Kit provided by Sunfounder for the PiCar-V renders them unnecessary.

Raspberry PI OS

The Raspberry Pi's official operating system is called Raspberry Pi OS (formerly ‘Raspbian’).
It is based on Debian and adheres to the same stability and performance principles. While it may not have the most bleeding edge software packages available, stability is farm more important for this project.

There have been numerous enhancements since the product's initial release in 2012.
A Raspberry Pi may be used as a regular desktop computer thanks to the PIXEL desktop (based on LXDE/LXQT), which is included with Raspberry Pi OS.

While the interface of the operating system is largely irrelevant to this project, it is still required as monitoring and testing camera functions is easiest on the raspberry pi itself, due to inconsistencies with remotely accessing a camera.

One of the main advantages of using the PIXEL desktop is how little resources it uses. LXQT uses 400MB of ram to run the entire desktop and its process.

Since LQXT is based on Debian Linux, the default package manager is APT (Advanced Package Manager). APT is one of the longest standing package managers and as a result, is highly stable and reliable. Additionally, a dependency resolver is included with APT, ensuring that all necessary packages are installed during the process.

Languages

Python

In recent years, Python has grown to be among the most used programming languages worldwide. It can be utilised in many scenarios from testing to web development and machine learning.

Python is a programming language designed to analyse data, automate processes, and create software and websites. Modern python has many packages and features meaning it may be used to create many different types of programs and applications and isn't specifically designed to solve a particular problem. Its beginner-friendliness and versitility has helped it to reach grow to its levels of usage today, which in turn means that there is extensive support and documentation for its functionality.

Python is easy to read and understand because of its relaxed syntax, which is similar to natural English. Development times are sped up as a result.
It is adaptable. Python can be used for a large range of projects but excels in machine learning and data science. Because it is open source, python can be used and distributed freely. Python has a large and growing library of modules. Python also boasts a large and active community that promotes, contributes modules and libraries, and serves as a vital resource for other programmers. Crucially, both “OpenCV2” and the “PiCar SDK” both have libraries built for the current Python version (3.10).

Libraries/Tools

OpenCV2

OpenCV2 is an open-source computer vision library. Computer vision is a method that allows us to visualize how media is stored and how to gather data from them. The main tool used in AI is computer vision. Automated cars, rely heavily on computer vision, along with robotics and photo editing programs.

The main features of OpenCV2 include.

- *Image/video I/O, processing, display (core, imgproc, highgui)*
- *Object/feature detection (objdetect, features2d, nonfree)*
- *Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)*
- *Computational photography (photo, video, superres)*
- *Machine learning & clustering (ml, flann)*
- *CUDA acceleration (gpu)*

("OpenCV - Overview - GeeksforGeeks")

The main features from this library used in this project are image/video processing and object/feature detection. Open CV2 implements various algorithms for this purpose which will be discussed below.

PiCar-V SDK

The PiCar-V SDK is provided by Sunfounder to remotely control the PiCar from another device. Much of the code within this package is for this purpose, or for the purposes of demoing its functionality. The SDK provides various pre-built functions for controlling each component of the PiCar separately. This SDK was originally provided for Python version 2.7 but is fully compatible with Python 3.10. To move the Picar with reference to the logic offered, the functions provided for each component can be implemented as a local library.

RPLidar Python Library

The RPLidar Python Library is a library of functions developed by the user Hyun-je *(Hyun-je)* for the Slamtec RPLidar range of sensors. Along with functions to carry out Lidar Scans, it offers a number of functions to manage the sensor and retrieve sensor-related data. This library will be used to implement the SLAM methodology, in conjunction with the LKAS to navigate around obstacles, while staying within the bounds of the “lane”.

Skoltech Robotics rplidar library

The rplidar library from Skoltech Robotics is an open-source software package that provides an easy-to-use interface for working with the RPLidar series of LIDAR sensors developed by Slamtec. The library is designed to simplify the integration and management of RPLIDAR sensors by providing a comprehensive set of functions to control and retrieve data from the devices. Written in Python, the library is compatible with multiple platforms and offers efficient, easy access to the sensor's features. It includes support for sensor initialization, motor control, data acquisition, and error handling.

Tinkercad

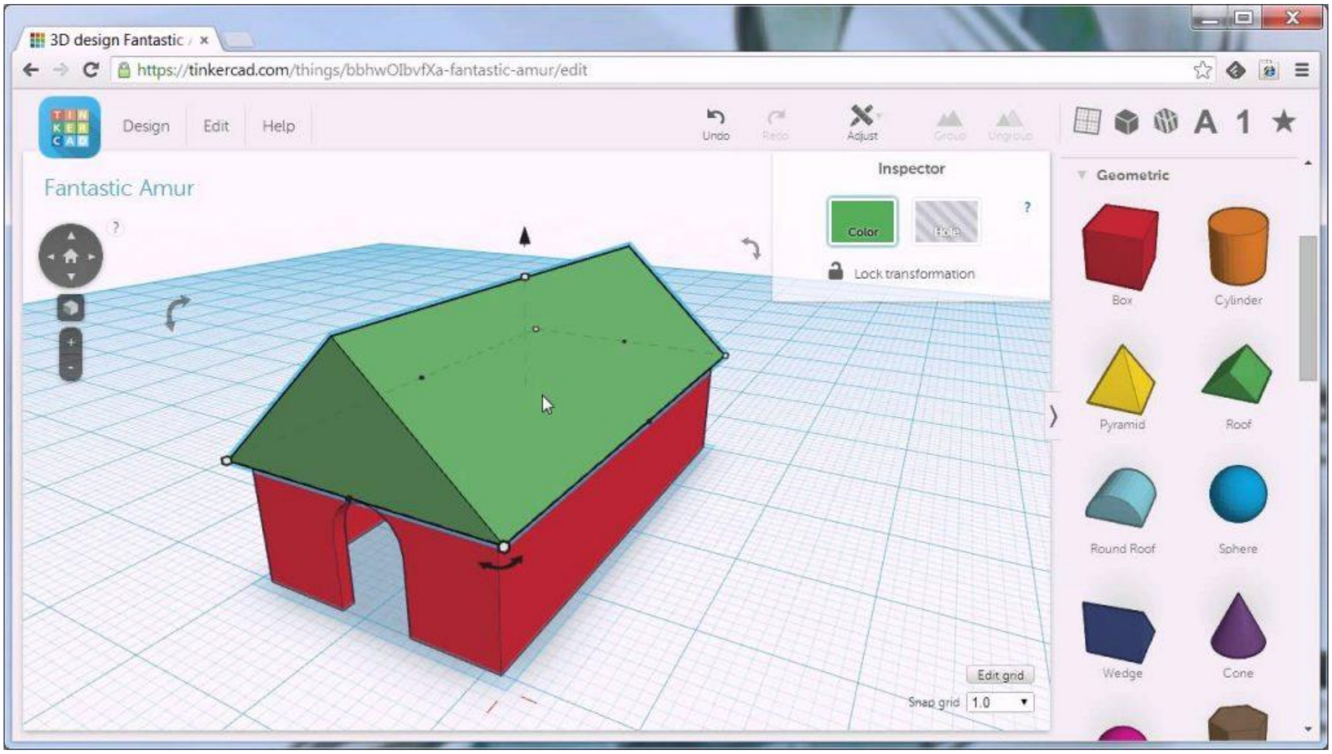

Figure 3: Example of Tinkercad interface

Tinkercad (*"Tinkercad | from Mind to Design in Minutes")* is a free to use online modelling tool. It caters to users with little prior experience in 3D modelling. Although it is the most accessible option, it is limited in terms of complexity. Given the simplicity of the proposed model along the time constraint, Tinkercad is highly suited to this project.

Algorithms

Haar Cascade

A Haar cascade is an algorithm that detects objects in an image, regardless of their size or location. Haar cascade is a classifier. It works to distinguish positive data points that are a part of the object discovered and negative data points that are not.

This algorithm operates in real-time as it is not too complex. This is a major benefit in context of this project, given the compute power of the Raspberry PI. A haar-cascade detector can be trained to recognise most objects and items given the correct training of the cascade. A haar cascade is trained by providing 1 or more “positive” images (images that do contain the object) and as many negative images (images that do not contain the object) as possible.

The Haar Cascade uses the cascading window technique, which detects features in a grid rectangular regions in a window. Individual frames of the video feed from the Picar's USB camera will serve as the detection window for this project.

Canny Edge Detection

The Canny edge detector employs a multi-stage method In order to identify edges in images,. It was invented in 1986 by John F. Canny. The method works in multiple stages. *To calculate the intensity of the gradients, a filter based on a Gaussian derivative is used. The Gaussian minimises the impact of any noise that may be present in the image. Then, by deleting gradient magnitude non-maximum pixels, potential edges are reduced down to 1-pixel curves. Using hysteresis thresholding on the gradient magnitude, edge pixels are then preserved or eliminated.* (Ugarte)

This method will be utilized to begin the process of lane detection in context of this project. Because there can be many lines detected in 1 image, the algorithms parameters need to be fine-tuned, filtered, and tested extensively to ensure accuracy, before attempting to single out lane lines using the Hough Line Transform method.

Hough Line Transform

The Hough Transform and was originally invented by Paul V. C. Hough to detect complex lines in photographs (Lee). Since its creation, the algorithm has undergone modifications and improvements that enable it to recognise other shapes, including particular types of circles and quadrilaterals. In the context of this project, the Hough line transform will be used to single out lane lines from other detected edges within a frame of video.

Convolutional Neural Networks (CNN)

This project intends to develop a system that enables the Picar to follow lanes autonomously with reasonably adequate precision using the aforementioned algorithms and methods. After achieving this, the system may be used to gather a significant amount of sample data, which can then be used to train a neural network model that can subsequently teach itself to drive more accurately.

A CNN is a subtype of the many artificial neural network models. A CNN is a type of network design with deep learning algorithms that are utilised for tasks like image recognition and data processing.

A CNN has three layers: the convolutional layer, the pooling layer and the fully connected (FC) layer. The complexity of the network increases as it moves from the convolutional layer to the FC layer. The CNN is able to identify increasingly larger and more intricate details of an image until it successfully recognises the complete object.

SLAM

SLAM (simultaneous localization and mapping) is a term used for autonomous vehicles that refers to the simultaneous construction of maps and localisation of the vehicle within that map. The vehicle is able to map uncharted environments using SLAM algorithms.

Braitenberg Algorithm

The Braitenberg Algorithm is a behavior-based algorithm on the work of the neuroscientist Valentino Braitenberg. The algorithm uses simple rules that are based on the behavior of insects to control the movement of the robot based on the sensor readings. It is a reactive approach that does not require a map of the environment or a planned path. The Braitenberg algorithm is implemented by assigning weights to the sensor inputs and using them to control the speed and direction of the robot. The algorithm is simple and efficient, but it is limited in its ability to handle complex environments and obstacles.

Artificial Potential Field

The Artificial Potential Field (APF) algorithm is a path planning method used in robotics and control systems. It is based on the principle of simulating a virtual potential field in the environment where the robot is operating, where the robot is attracted to the goal and repelled from obstacles.

The APF algorithm works by defining an artificial potential field, where attractive forces pull the robot towards the goal and repulsive forces push it away from obstacles. The resulting motion of the robot is the sum of these attractive and repulsive forces. The algorithm is simple and effective, and it can be used to plan collision-free paths in real-time.

One of the key advantages of the APF algorithm is that it can be used to plan paths for a wide range of robots, including wheeled and legged robots, as well as drones and other aerial vehicles. However, the algorithm can also suffer from some limitations, such as getting stuck in local minima and being sensitive to the shape of the obstacles.

Vector Field Histogram

The Vector Field Histogram (VFH) method is a path planning technique used for obstacle avoidance and navigation in dynamic environments. It is based on the concept of building a polar histogram of the environment and then calculating the robot's movement based on the detected obstacle distribution.

The VFH algorithm works by dividing the robot's surroundings into a grid and collecting sensor data to detect obstacles. This data is then used to create a two-dimensional histogram of the environment, where each cell represents a particular direction and distance. The histogram is then smoothed and reduced to a one-dimensional polar histogram that represents the angular distribution of obstacles around the robot.

The robot's motion is then determined by selecting the least occupied sector in the polar histogram, which corresponds to the direction with the lowest obstacle density. The VFH algorithm ensures that the robot moves toward the goal while avoiding obstacles, adapting its motion in real-time as the environment changes.

One of the key advantages of the VFH method is its ability to handle dynamic environments with moving obstacles, making it suitable for applications such as autonomous vehicles and mobile robots. However, the VFH algorithm can be computationally intensive, particularly for high-resolution grids or complex environments, which may limit its real-time performance.

Evaluation

In order to assess the performance of the autonomous vehicle system, a variety of test scenarios will be designed to simulate diverse driving conditions, environments, and challenges.

The test scenarios may include:

1. Straight pathways featuring clear lane markings.
2. Curved pathways.
3. A range of lighting conditions, such as lit, and shadowy areas.
4. Different surfaces.
5. Dynamic environments containing moving and unmoving obstacles.

By utilizing these test scenarios, the system's performance can be evaluated under a wide range of circumstances.

Limitations

Despite the objectives of this project, there are several limitations that should be acknowledged:

Scalability

The project is based on a small-scale vehicle model, and the results may not directly translate to full-sized vehicles or complex environments. Scaling up the proposed system to larger vehicles or more dynamic environments may require additional sensors, hardware, and computational resources.

Limited Sensors

The project relies on a single LiDAR sensor and a 120° wide-angle camera for spatial mapping and obstacle detection. This may limit the system's ability to detect and avoid obstacles in certain situations, such as in low light conditions or when dealing with reflective surfaces.

Computational Power

The Raspberry Pi 4B, while suitable for this project, may not offer sufficient computational power for more advanced autonomous vehicle applications, such as real-time machine learning or complex decision-making algorithms.

Safety and Security

The project does not address safety and security concerns associated with autonomous vehicle technology, such as potential vulnerabilities to hacking or system malfunctions.

Ethical Considerations

The project does not delve into the ethical considerations related to autonomous vehicles, such as their impact on employment, data privacy, or the decision-making process during critical situations.

Legal and Regulatory Factors

This project does not consider the legal and regulatory aspects of autonomous vehicle deployment, which may vary across different jurisdictions and could affect the adoption and implementation of the proposed technology.

By acknowledging these limitations, future research and development can focus on addressing these challenges to improve the performance, safety, and applicability of autonomous vehicle systems.

Similar Established Projects

Nvidia Dave2

Figure 4: Nvidia Dave-2 in action

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.

Summary and Conclusions

In conclusion, the purpose of this project is to develop a small-scale self-driving car using a front-facing camera feed and a combination of manually programmed and more precise deep learning solutions. The self-driving system will also be combined with an obstacle avoidance solution that utilizes a Lidar sensor for a more precise performance. It is believed that the previously discussed technology, software, tools, and techniques are the most effective ways to implement these solutions.

Glossary

ACC- Adaptive Cruise Control

APF- Artificial Potential Field

CNN – Convolutional Neural Network

LKAS – Lane Keep Assist System

LIDAR – Laser imaging detection and ranging

PCB – Printed Circuit Board

ROS – Robot Operating System

SLAM – Simultaneous Localization and Mapping

Bibliography

Abdallah, Abdelrahman. “Benefits of Vehicle Automation for Public Transportation Operations.” *Current Trends in Civil & Structural Engineering*, vol. 6, no. 5, Iris Publishers, Dec. 2020, pp. 1–5, irispublishers.com/ctcse/fulltext/benefits-of-vehicle-automation-for-public-transportation-operations.ID.000646.php. Accessed 11 Apr. 2023.

Bechtel, Michael, et al. “DeepPicarMicro: Applying TinyML to Autonomous Cyber Physical Systems.” *ArXiv.org*, 2022, https://doi.org/10.48550/arXiv.2208.11212. Accessed 17 Nov. 2022.

Bojarski, Mariusz, et al. “End to End Learning for Self-Driving Cars.” *ArXiv.org*, 2016, https://doi.org/10.48550/arXiv.1604.07316. Accessed 17 Nov. 2022.

“Enterprise Open Source and Linux | Ubuntu.” *Ubuntu*, 2020, ubuntu.com/. Accessed 14 Nov. 2022.

Huang, Tony. “RPLIDAR-A2 Laser Range Scanner_ Solid Laser Range Scanner|SLAMTEC.” *Slamtec.com*, 2016, www.slamtec.com/en/Lidar/A2. Accessed 16 Nov. 2022.

Hyun-je. “Hyun-Je/Pyrplidar: Full-Featured Python Library for Slamtec RPLIDAR Series.” *GitHub*, 12 Aug. 2019, github.com/Hyun-je/pyrplidar. Accessed 16 Nov. 2022.

Jaiswal, Abhishek. “Object Detection Using Haar Cascade: OpenCV.” *Analytics Vidhya*, Apr. 2022, www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/#:~:text=Haar%20cascade%20is%20an%20algorithm,%2C%20buildings%2C%20fruits%2C%20etc. Accessed 17 Nov. 2022.

Kamerpower. “10 Best Self Driving Cars with Self-Driving Features 2022.” *Kamerpower™*, Kamerpower™, 4 May 2022, kamerpower.com/10-best-self-driving-cars-with-self-driving-features-2022/. Accessed 13 Nov. 2022.

Lee, Socret. “Lines Detection with Hough Transform - towards Data Science.” *Medium*, Towards Data Science, 2 May 2020, towardsdatascience.com/lines-detection-with-hough-transform-84020b3b1549#:~:text=The%20Hough%20Transform%20is%20an,and%20quadrilaterals%20of%20specific%20types. Accessed 17 Nov. 2022.

“OpenCV - Overview - GeeksforGeeks.” *GeeksforGeeks*, 23 Sept. 2019, www.geeksforgeeks.org/opencv-overview/. Accessed 15 Nov. 2022.

“ROS.org | Powering the World’s Robots.” *Ros.org*, 2020, www.ros.org/.

“SAE Levels of Driving Automation™ Refined for Clarity and International Audience.” *Sae.org*, 2021, www.sae.org/blog/sae-j3016-update. Accessed 16 Feb. 2023.

SkoltechRobotics. "SkoltechRobotics/Rplidar: Python Module for RPLidar A1 and A2 Rangefinder Scanners." *GitHub*, 9 Apr. 2019, github.com/SkoltechRobotics/rplidar. Accessed 17 Apr. 2023.

Stefanini. "Increased Productivity and Other Benefits of Using AI in Warehouse Automation - Stefanini." *Stefanini*, 13 Apr. 2022, stefanini.com/en/insights/articles/benefits-of-using-ai-in-warehouse-automation-and-increased-produ. Accessed 11 Apr. 2023.

"SunFounder Focuses on STEAM Education with Open-Source Robots." *SunFounder*, 2022, www.sunfounder.com/. Accessed 14 Nov. 2022.

"Tinkercad | from Mind to Design in Minutes." *Tinkercad*, 2022, www.tinkercad.com/. Accessed 17 Nov. 2022.

Unihost.FAQ. "How to Choose Linux Desktop Environment (RAM Usage)." *Unihost.FAQ*, Unihost.FAQ, 23 Apr. 2021, unihost.com/help/how-to-choose-linux-desktop-environment-ram-usage/. Accessed 14 Nov. 2022.

Vrchota, Jaroslav, et al. "Precision Agriculture Technologies for Crop and Livestock Production in the Czech Republic." *Agriculture*, vol. 12, no. 8, July 2022, p. 1080, https://doi.org/10.3390/agriculture12081080. Accessed 11 Apr. 2023.