COMPUTER VISION AND SENSOR FUSION FOR AUTONOMOUS VEHICLES

An Undergraduate Research Scholars Thesis

by

ABD-ALLAH EL-ATTAR[1], ABDELWAHID ELTAYEB[2], AND AHMAD AL-KHATEEB[3]

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Hussein Alnuweiri

May 2022

Majors: Electrical Engineering[1,2,3]

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

We, Abd-Allah El-Attar[1], Abdelwahid Eltayeb[2], and Ahmad Al-Khateeb[3], certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with our Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

ABSTRACT

Computer Vision and Sensor Fusion for Autonomous Vehicles

Abd-Allah El-Attar[1], Abdelwahid Eltayeb[2], and Ahmad Al-Khateeb[3]
Department of Electrical Engineering[1,2,3]
Texas A&M University

Research Faculty Advisor: Dr. Hussein Alnuweiri
Department of Electrical and Computer Engineering
Texas A&M University

Cars, particularly manually-driven cars, are one of the most commonly used modes of transportation today. However, millions of people are either killed or left with disabilities annually due to road traffic accidents caused by human error or sensor failures. Despite that, a lot of people seem reluctant to look into alternatives to manually driven vehicle transportation. This is understandable as driving cars has been the trustworthy mode of transportation for many years, and it is widely used in everyday life around the world. However, technological advances in the fields of machine learning and cyber-physical systems contributed to the emergence of nearly or fully autonomous vehicles, or driverless cars, as a true viable alternative for the current human-controlled driving mode. The technology still has a long way to go, mainly because the advances in vision and depth measurement sensors such as LIDARs can not achieve the levels of safety needed to make fully autonomous cars. Progress on this front is being made every day, and it seems inevitable that they will be readily available in the near future. Our team aims to further investigate the application of Computer Vision and sensor fusion to achieve independent self-driving without external guides. To accomplish this, we combine a depth camera with a LiDAR to provide better coverage of the

surroundings and allow more accurate detection and thus accurate avoidance of obstacles. We are mounting the vision system on a model driverless car and using the vision data to guide the car control system. A computer vision algorithm will be run by the NVIDIA Jetson Nano to determine what course of action the car should take. The final prototype should be capable of driving at a reasonable speed without colliding with any objects and making decisions such as braking or turning when necessary.

ACKNOWLEDGMENTS

Contributors

We would like to thank our faculty advisor, Dr. Hussein Alnuweiri, for his guidance and support throughout the course of this research.

Thanks also go to our friends and colleagues and the department faculty and staff for making our time at Texas A&M University a great experience. A special thanks to Mr. Wesam Mansour and Abdulrahman Sayedahmed for their help with this project.

Finally, thanks to Dr. Wael Alhajyaseen from Qatar University for his support.

The materials used for "Computer Vision and Sensor Fusion for Autonomous Vehicles" were provided by the F1TENTH organization.

All other work conducted for the thesis was completed by the students independently.

Funding Sources

Undergraduate research was supported by the Electrical and Computer Engineering department at Texas A&M University at Qatar. No other funding was given to this project.

NOMENCLATURE

SAE	Society of Automotive Engineers
CV	Computer Vision
AV	Autonomous Vehicle
YOLO	You Only Look Once
LiDAR	Light Detection And Ranging
ROS	Robot Operating System
VESC	Vedder Electronic Speed Controller

1. INTRODUCTION

Car accidents are one of the most dangerous risks many commuters face every day. This is the case for both pedestrians and passengers. Every year the lives of approximately 1.35 million people are cut short as a result of road traffic accidents, with up to 50 million people suffering injuries and many of those sustaining a disability as a result of their injury. In fact, road traffic injuries are the leading cause of death for children and young adults aged 5-29 years, as per the World Health Organization. [1] This is precisely why self-driving cars are becoming more of a reality every day, along with the fact that another major selling point is the comfort it provides to its passengers. Unfortunately, current self-driving technology is not quite complete and cannot be unequivocally trusted to take the wheel. Ideally, autonomous vehicles will be able to reduce car accidents significantly by eliminating its largest cause: driver error. However, true self-driving cars have a long way to go, even though they are starting to emerge as the more feasible option over normal manually-driven cars in many ways, as there is still the question of how safe it is to trust the vehicle with the life of a passenger or that of a pedestrian. The fatalities they have caused are not helping that case, as there have been quite a few incidents involving self-driving cars in the past that have cost people their lives. The race to reliable and truly autonomous vehicles has started, and thus different approaches to achieving this are being researched, developed, and tested.

To distinguish between the different degrees of automation in vehicles, the Society of Automotive Engineers (SAE) developed a classification system that defines the degree of driving automation a car and its equipment may offer, called the SAE Levels of Autonomous Driving, ranging from levels zero to five. The driving automation spectrum begins with vehicles without this technology and ends with entirely self-driving vehicles. Here is a brief description of each level of autonomy:

- Level 0: No automation. Full-time performance by driver

- Level 1: Driving Assistance. System can control either lateral (steering) or longitudinal (acceleration, braking) motion but not both. Examples: Adaptive Cruise Control, or Lane Keeping Assistance

- Level 2: Partial Automation. System can control both lateral and longitudinal motion, but driver must always pay attention
- Level 3: Conditional Automation. Same as Level 2, but also includes automated object and event detection response. It does not require constant driver attention, but the driver must be instantly ready to take control when the system experiences a failure
- Level 4: High Automation. Same as Level 3, but if the system fails it can automatically handle the emergency without a need for immediate driver control. So there are virtually no safety concerns if the driver is sleeping etc.
- Level 5: Full Automation. The vehicle will be able to autonomously drive in any condition without human interaction. There would not be a need for pedals or a steering wheel.

The objective of this project is to work towards levels 3 and 4, which requires accurate and reliable sensing. To get an accurate perception of the environment, multiple sensors can be used simultaneously. [2] Each type of sensor has its advantages, and thus using a variety of them helps compensate for the shortcomings of the others. For example, cameras can be used in conjunction with range sensors (LiDAR, SONAR, or RADAR) to achieve better coverage in different environments. The main goal of this project is to increase the accuracy of perception and obstacle avoidance in autonomous vehicles so that they are able to maintain a good estimate of depth, accelerating or decelerating whenever required. With this project, we hope to achieve an efficient computer vision and sensor fusion system. The implementation of such a system in real life on a larger scale, if successful, can significantly reduce the fatalities caused by autonomous vehicles, which in turn would contribute to the mitigation of the stigma surrounding autonomous vehicles.

1.1 Proposed Solution

Our project hopes to find a combination of sensors (including LiDAR) and cameras to create a computer vision system that can accurately detect and avoid obstacles. The onboard computer (Nvidia Jetson Nano) will run the computer vision algorithm and decide what course of action the car should take. The final prototype is expected to be able to drive at a reasonable speed without crashing into any obstacles, while making decisions such as breaking or turning in a reasonable time (comparable to the average driver). With this project, we hope to achieve an efficient computer vision and sensor fusion system that can guide an autonomous vehicle prototype without any trouble. If implemented on a larger scale, such as system could potentially significantly reduce the fatalities caused by autonomous vehicles.

2. LITERATURE REVIEW

Even though autonomous vehicles are seen as a viable next step in transportation, they face many challenges in public and governmental acceptance. To achieve our objective of increasing the trust of the public in such vehicles, we need to understand the background of this field, the progress being made on this front, and what contributions can be made. Therefore, we take a look in this section at the relevant literature to gain a better understanding of what projects such as this one entail and how some similar projects are being implemented. By doing this, we can find out how our project can be different and what it can do to improve. The relevant literature was collected through resources such as the IEEEXplore database and Google Scholar. This enables us to learn more about what we can do to gain an edge, overcome the difficulties that other projects might have faced, and perhaps take the necessary steps that will make our project stand out and contribute to the field of autonomous vehicles.

2.1 Related Works

Many attempts have been made throughout history to construct an autonomous vehicle that can operate without the assistance of a human driver, such as Leonardo da Vinci's self-propelled cart and Dickmann's VaMP self-driving car using 4D vision technology. However, a significant contribution to the initiative is that of the Defense Advanced Research Projects Agency's (DARPA) Autonomous Land Vehicle (ALV) project, which uses advanced sensors and is capable of travelling autonomously at high speeds. Between 2004 and 2007, DARPA launched a number of competitions, which prompted studies into the potential of self-driving cars. Following that, Google began creating its own self-driving car, and other businesses such as Tesla, Uber, and NVIDIA soon followed suit. [3]

Sensors detect the properties of an environment or changes to an environment and are essential for correctly perceiving the environment. When sensors are used in perception, there are different comparison metrics taken into account by sensors, such as resolution, the field of view, and dynamic range. A commonly used sensor for perception is the LiDAR. LiDAR stands for Light Detection and Ranging. This device uses light pulses to detect ranges. It works on nearly the same principles as RADAR, which is Radio Detection and Ranging. There are several varieties of LiDARs available, including 3D and spinning 2D LiDARs. By generating millions of data points per second, LiDAR sensors provide high precision and accuracy in object detection and recognition in Advanced Driver Assistance Systems (ADAS), creating a 3D point cloud reconstruction of the surrounding area. With a high data rate, LiDAR provides positional accuracy and precision. The bulkiness of LiDARs and the cost of employing them for ADAS applications are two major drawbacks. An alternative for this is the stereo camera sensor, which is a passive reflective device composed of two cameras. It can be used for 3D reconstruction of the environment, which reduces the need for the more expensive LiDAR. [3]

Cameras are an appealing choice for self-driving vehicles due to their relatively low price and range of potential applications. With two different points of view, it is possible to extract depth information by using stereo matching. [2] This method, along with Computer Vision, can help predict the distance and motion of objects detected in the scene. This means that cameras alone can prove to be very useful, but they cannot be fully depended on. Calibrating a multiple-camera set up to help establish a 3D representation can be challenging and typically requires a considerable amount of tweaking. In [4] Geiger et al. proposes a solution to automate the calibration of a system of cameras and sensors with relative ease.

The main purpose of cameras in such implementations is obstacle detection. The stereo camera initially generates a left image and a right image. After that, the two images are rectified. Given the extrinsic and intrinsic properties of the systems, rectification is a technique of computing the image transformation that turns epipolar lines collinear that are parallel to the horizontal axis. The disparity map is calculated using the camera characteristics extracted from the rectified pictures. The pixel-wise depth can be computed using the disparity map, which calculates the disparity between the left and right camera images. This implementation is vital in obstacle detection training, as is the case with Amara et. al's work where the algorithm is trained to detect a total of six classes, which are pedestrian, vehicle, bicyclist, cow, bus and dog classes. [3]

Information obtained through the camera used in autonomous vehicles could also use image recognition algorithms for Unmanned Ground Vehicles (UGVs) used to prevent accidents by measuring the distance between the vehicle itself and the vehicle in front of it, as per Mohamed et. al. [5] It may also be used to recognize lanes and prevent the vehicle from leaving its lane by measuring the distance between the car's wheels and the lane it was in. Furthermore, the automated driving system can respond appropriately to road signs by identifying them. [5] In the application of autonomous vehicles, after collecting the sensor and other perception data, it needs to be processed and used to guide the vehicle. This is basically where the "brain" of the vehicle comes in. In prototypes, certain computers are used for such purposes, such as NVIDIA's Jetson products, the Xavier or TX2.

A project was done in Misr International University where a prototype self-driving car was designed and built. [6] Their prototype was relatively small with a Raspberry Pi as the central processor of the system, along with an Arduino UNO in charge of motion and a L298N motor driver. The prototype had three main systems: depth perception using disparity maps, lane detection, and anomaly detection. [6] The video feed from one of the cameras gets put through multiple filters to reduce noise and prepare for edge detection and then a Hough Transform is used to identify the lane lines and the Raspberry Pi decides on the if the car needs to turn or change its speed based on its location and rotation in relation to the lane lines for lane detection. The project also utilizes two cameras with slightly different perspectives to generate a disparity map and thus a depth map. This requires some calibration beforehand, which is vital for good results when using two separate cameras rather than a single stereo camera. Furthermore, an accelerometer and gyroscope are used to detect anomalies in the car's motion. The accelerometer is used to measure the force exerted on the car and the gyroscope is used to measure the orientation of the car and its angular velocity. This information is valuable to determine the state of the car, and possibly road surface, such as road bumps, inclines, and other anomalies. [6] Based on their tests the disparity map proved to be very effective and worked well in both indoor and outdoor conditions. The anomaly detection system proved to have a 98.6% success rate in detecting anomalies. It is reported that the prototype performed as well as expected after some tweaking, with all tests being perfectly satisfactory. There is a lack of information on some of the technical details of the prototype and its performance, but judging from the design it seems to move at a relatively slow speed and is tested on a similarly small track. Therefore, the system might not be able to function as well at higher speeds, similar to those reached in other projects. It is only an assumption, but a lot of the components would have to be upgraded such as the Raspberry Pi for real-time processing at higher speed. [6]

Another project done on a larger scale was in the Worcester Polytechnic Institute in Massachusetts, USA. [7] The focus was to implement a miniature autonomous racer car, with a 1/10 scale, similar to our project. Actually, a great deal of the project was also on the technical aspects of the vehicle's motion: continuous variable transmission, speed control etc. The autonomous aspect of this project narrows down to implementing adaptive cruise control (ACC), trajectory generation, and a trajectory tracking controller. The ACC was achieved using some algorithms in combination with filtered LiDAR data to speed up or slow down depending on the readings for the region directly ahead of the car. The design also called for localization which was determined using Gmapping based on LiDAR data and odometry data calculated from the depth image produced by the Zed stereo camera, which does not seem to be used for any other purpose. Some of the main components of the project are: a Hokuyo UST-10X LiDAR, a NVIDIA TX2 as the main processor, a Zed depth camera for mainly odometry, and Teensy 3.2 boards used for low-level motor control. [7] Overall, the prototype seems to be effective based on the project's results, especially as a base for an ambitious goal. It is concluded that the car can generate a smooth trajectory and follow it smoothly with very small error, and ACC is implemented well with the absolute error smaller than 10 cm and angle smaller than 10 degrees. [7] Still, the prototype suffered from some issues. During implementation, the prototype was getting unreasonably heavy and the original acrylic base material was relatively fragile therefore it was replaced with ABS, but this necessitated the use of a PD controller to effectively control the speed. There were also some errors in localization due to incorrect odometer data from the zed camera, as it is not a consistent method to obtain odometry, but they had more success with it compared to measuring the RPM of the car's shafts. Also, the racing motor used could not be run at slow speeds and thus it would suddenly turn on and off when the car is following a slow moving object, which would lead to violating the safe distance. The LiDAR tended to overheat after long runtime, and the car seemed to have issues with driving perfectly straight. [7]

2.2 Analysis

Despite the advancements in research done in the literature cited, there are a number of aspects that are overlooked. In [3], what Amara et. al fails to consider what might occur if the vehicle is not able to detect other objects like walls or cones, with the number of categories identified by the algorithms particularly limited and quite specific and no mention of any other obstacles that can be avoided. On the other hand, there is the algorithm mentioned in [5], where it is capable of detecting lanes, vehicles, and road signs with no mention of any perception of pedestrians or any other objects. Perception in autonomous vehicles is a very big issue that needs to be solved.

There are similarities observed in various criteria when looking at the projects of MIU and WPI, which is not surprising as all of the projects, in one way or the other, are working towards the same objective which is essentially to increase trust in self-driving cars. Despite being mentioned in the context of race cars, the technology used in the WPI prototype conforms with what a standard autonomous vehicle prototype should use, validating its similarities with our project.

The WPI project has the most expensive prototype since it is designed to be an autonomous race car. Meanwhile, on the other end of the spectrum is the MIU prototype which is fairly simplistic based on its materials and the type of processor used. We anticipate that our prototype will be the most efficient one in that it optimizes the use and cost of its materials while still maintaining a good design, since we plan on using certain materials that are considered to be in the median range in terms of cost and performance. Therefore, it is estimated that the projected cost of a real vehicle based off of our prototype could be considered decent compared to how much a self-driving car on the market could usually cost, taking into consideration its design and its cost. In simple terms, our project aims to be a middle ground, cost effective solution with a reasonable price and good technology. Examining this literature enables us to learn more about our project and the possibilities for advancement and what sort of issues might arise.

While the general performance parameters of autonomous vehicles should not be taken lightly, the topic that requires more focus is perception in such vehicles. That is because inaccurate perception is the leading cause of accidents in autonomous vehicles. For example, the first death of a Tesla driver occurred when he collided with a truck while driving on autopilot. The car failed to detect the truck trailer as an impediment because of its “white color against a brightly lit sky” and “high ride height” despite it being the driver’s responsibility to keep an eye on the road and not let the car drive fully alone. In another incident, a woman named Elaine Herzberg was the first pedestrian fatality reported involving an Uber self-driving test car. [8] These cases are just a drop in the ocean in terms of the incidents that have occurred and a hint of what is to come if more extensive research into increasing the accuracy of the perception of autonomous vehicles is not done.

3. METHODS

In this section, we will take a look at the system architecture of the design and explain how the different components are configured and how they come together so they can be implemented into the final design. Next, we will examine the standards and constraints of the design so that we gain an understanding of the criteria that may be adhered to in order to make the project successful and feasible. After that, we will look at the system design overview that outlines exactly what the expected outcome of the project is and what it entails.

The design we are building for this project is called an F1/10 vehicle. These vehicles are 1/10 scale RC-based models of regular cars. These models are usually used for racing competitions. However, they can also be used to implement, verify, and test algorithms that can ultimately be applied to full-scale driverless automobiles due to their real autonomous car-like architecture. This project is one of them. Not only is it cheaper to use a model automobile instead of a real one, but because they can be produced on a 1/10 scale, all of the scenarios explored are considerably more manageable. Furthermore, all collisions involving a model automobile are likely to be far less serious than in real scenarios. This enables us to achieve our objectives more effectively, and at a rapid pace. It also helps us understand where mistakes might occur, and with little to no chance of casualties.

3.1 Components

The main components used to build the car comprise a car, an embedded computing board, LiDAR, and a depth camera. The car used is a Traxxas Slash 4X4 RTR 4WD Brushed Short Course Truck, the computing board is an NVIDIA Jetson Nano Developer Kit, the LiDAR is a Hokuyo 10LX, and the camera is a Intel RealSense D435i Depth Camera. We will now take a look at how each component was configured.

3.1.1 Car

The main use of the car is as a platform on top of which the other components will be mounted so that it can then utilize the components to run so that we can test the computer vision and sensor fusion system. We first start by setting up the lower level chassis. The way this is done is that all the original components within the car are disassembled, except for the motor and servo, to make way for the new components. After that, the autonomy elements are mounted onto the upper level chassis, which is essentially a laser cut platform deck. The elements mounted on top of the upper level chassis include a VESC 6 MkV Vedder Electronic Speed Controller, antenna standoffs, and cables, which are then be connected to the NVIDIA Jetson Nano, also mounted on the upper level chassis, and then finally, the powerboard and LiDAR. [9] As for the Intel RealSense D435i Depth Camera, it will be mounted on top of a 3D printed bracket. The bracket will be screwed onto the upper level chassis so that it is the highest point in the car, and the camera will then be screwed onto the top. This is done so that no other component on the car obstructs the camera's vision.

3.1.2 NVIDIA Jetson Nano

The Jetson Nano requires 5V to power on, and this could be supplied in two ways: using a micro USB connection or a barrel jack connection. For testing the Nano, the micro USB method was most used, although the maximum current that could be provided to the computer would only be 2A. When more power is needed, and when the Nano is fitted into the platform, a barrel jack connection is used to power the computer through the power board.

Using Python, the appropriate Machine learning (ML) Computer Vision platforms, such as OpenCV, TensorFlow, and other ML libraries, are used to manipulate the autonomy elements of the car through the Jetson Nano, which is connected to the LiDAR and Depth Camera. The Nano is also connected to the VESC controller to control the car.

Figure 3.1: NVIDIA Jetson Nano

It is worth mentioning that the targeted onboard computer was another, more powerful NVIDIA product, the Jetson Xavier. However, its lack of availability and constant price hikes led the team to work with the Jetson Nano, shown in Figure 3.1, instead to keep the cost of the design as moderate as possible.

3.1.3 LiDAR

Figure 3.2: Hokuyo UST-10LX LiDAR

The Hokuyo UST-10LX LiDAR, shown in Figure 3.2, is an accurate way to measure distances from the car to objects up to 10 meters away with millimeter accuracy. Before being able to use the Hokuyo sensor, however, some setup was required for the LiDAR's connections to be compatible with the other components. The LiDAR has 6 total wires bound to a connector, two of which are the power wires and should be isolated to power the system. The wires were separated from the connector using a wire cutter, and the 4 control wires are taped back using electrical tape. The 2 power wires, which were the brown and blue wires for 12/24V and GND (ground) respectively, are stripped from their leads and powered directly from a bench power supply or a power board when fixed on the vehicle.

The sensor must be configured with the correct network settings to be able to communicate with the computer, so a new network connection must be made. By setting the IP address to 192.168.0.15, the subnet mask to 255.255.255.0, and the gateway to 192.168.0.1, the LiDAR can be utilized with the factory connection settings. Since some models might have severely outdated firmware, an update is necessary before any connection is established. Hokuyo's Smart Updater tool was used to update the firmware, but since it can only operate in a Windows operating system, a laptop was used to install the update. After setting up the network settings for Windows, the LiDAR was recognized and its firmware was updated.

The next step in the LiDAR's setup includes the installation of the Robot Operating System (ROS) libraries responsible for controlling the car and visualizing the output of the LiDAR. The ROS version used is the ROS Melodic package, which is compatible with the Jetson Nano's operating system, Ubuntu 18.04 (Bionic). After troubleshooting and fixing an issue with the python_catkin package, required to build a work space for ROS, the Nano system was successfully able to run ROS on the terminal by executing the 'roscore' command. The result of running 'roscore' is shown below in Figure 3.3.

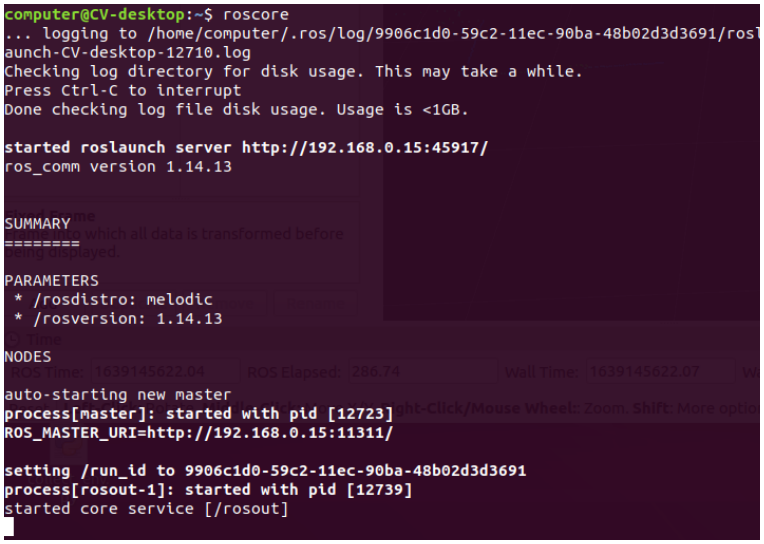

Figure 3.3: Terminal with ‘roscore’ command running

LiDAR data is then collected by opening a new terminal and running ’rosrun urg_node urg_node _ip_address = “192.168.0.10”’, this command establishes the connection with the sensor and the Nano starts receiving data from the LiDAR.

3.1.4 Depth Camera

Figure 3.4: Intel D435i Depth Camera

The Intel D435i depth camera used, shown in Figure 3.4, is a versatile and crucial component of the system. The D435i has an IR projector and three camera lenses: one RGB imager and two IR imagers used to produce the depth map. The D435i model, specifically, has an Inertial Measurement Unit (IMU) that can detect motion and rotation in 6 degrees of freedom.

In the project, the depth camera was used in a variety of ways. There are three different perception methods that utilize the camera. For the first one, the RGB frame obtained from the camera is fed through the YOLO object detection algorithm to detect all major identifiable objects in the frame, such as persons or cars. Then by comparing the position of those objects in the image with the depth map generated from the depth camera, the distance to that object is identified. Furthermore, this allows tracking the object by considering its change in position and thus also motion between frames.

The second method considers using the depth camera to generate a simple map similar to those produced by a Radar or LiDAR. In the sense that only the distances measured on the same horizontal plane as the camera are considered. This method allows for an accurate perception of obstacles at the height of the camera, and thus generally the car, without any complications from dealing with three dimensions, such as floors and overhanging objects like traffic lights/tunnels. The readings of the depth map are taken at exactly the halfway point of the depth image width. This functions similar to the LiDAR used but with worse performance. Still, the depth camera provides very accurate results, especially within about 6 meters, and its readings can be used to reinforce the data from the LiDAR. Furthermore, it will be mounted at a different height than the LiDAR, and thus it would provide a similar but different perspective on the environment. This approach introduces a significant amount of valuable data without being computationally intensive.

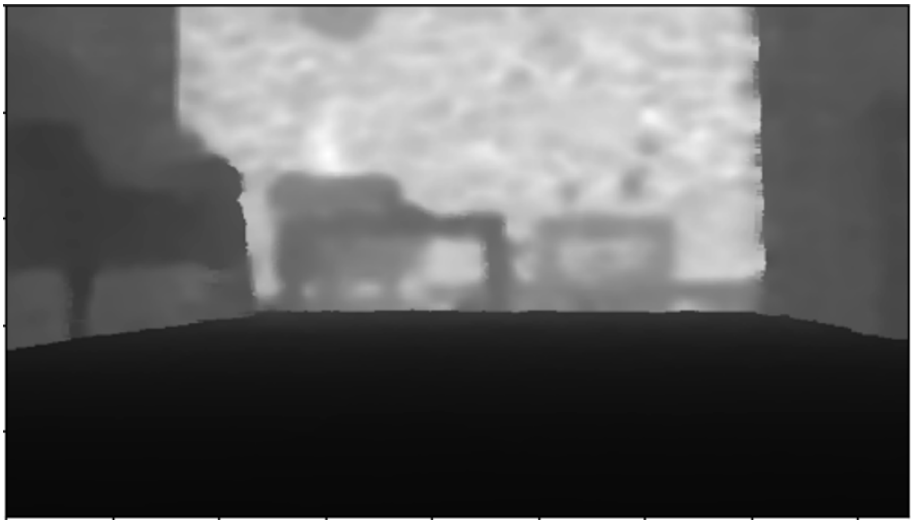

Figure 3.5: Calibration Depth Image

The third method focuses on the identification of anomalies directly ahead of the car at close range. The shortcomings of the LiDAR, and even the second method involving the depth camera, is that it does not detect everything in the car's path. They only perceive obstacles large/tall enough to show up on their plane of detection. Still, there are many potential obstacles to the car that could be small or short enough to go undetected by the LiDAR and even by the other methods using the depth camera. This is why short-range anomaly detection is needed. This detection is done based on a pre-determined depth map, as seen in Figure 3.5. Essentially, a depth map of an empty open floor is first captured by the camera. The data of that depth map will be taken as the ground truth of an empty, obstacle-free floor. Then during the operation of the AV system, the car will constantly compare the current depth map input to that of the previously captured depth map. Based on the variance from the original empty scene, it is possible to determine if there are anomalies in the car's path. As any significant change in the depth map will be detected. This method is not limited to detecting objects in the car's path, but it could also identify sudden changes in elevation, such as a drop-off that otherwise cannot be detected by the car's systems. However, this scan is only done for the lower portion of the image, as it requires accurate measurements that are only guaranteed in close-range. Otherwise, the variance of the measurements at further distances would constantly set off the system. An example of the depth anomaly detection method running can be seen in Figure 3.6. The closer than expected anomalies are highlighted in blue, while the sudden drop-off at the edge of the table is highlighted in red.

Figure 3.6: Depth Anomaly Detection Example

3.2 Standards and Constraints

While there are no established standards that fit this kind of autonomous vehicle project, there are other standards that can be followed to ensure that the project can be as successful as possible. One of these standards is the camera setup for computer vision algorithms, and according to this survey of other CV projects [9], the most used setup is the USB 3.0 setup as it provides wide compatibility and speeds, as well as it being the recommended method for the Intel depth camera used in this project.

As for technical constraints, the equipment used in the design of the car each have their advantages and disadvantages that had to be considered before being purchased. The Hokuyo UST-10LX LiDAR is a compact sensor, and its size is an attractive feature when platform size is limited. However, its small size comes with sacrifices in technology, as for example, the LiDAR's scan angle covers 270 degrees. [10] Since our project is on a small scale, the accurate range of the scanner can be relatively small. The UST-10LX can measure up to 10 meters with $\pm$ 40mm. [10]

As for the Jetson Nano, the size of the computer was compatible with the mount chosen for the platform, but compared to other Jetson products, the Nano falls short of the average computational power. For example, the Jetson Nano has a 128-core GPU [11], while the more advanced Jetson Xavier NX has a 384-core GPU. [12] The Nano has enough processing power for our purposes of computer vision and decision-making, and in addition to its lower cost compared to the Jetson Xavier NX, it was a worthwhile compromise for this project.

The Intel camera comes with a depth frame to measure distances from the lens, but according to tests done by the team (discussed later), the depth image is only reliable from 5 meters or less. This is far too short to accurately detect distances while the car is moving at any reasonable speed, and so the LiDAR's data can be given priority for farther ranges.

The constraints of the components on top of the car need to also be examined to ensure that it will work as expected. First, there is the LiDAR, which possesses a scan angle of 270 degrees and can obtain measurement data in a wide field of view up to a distance of 10 meters with millimeter resolution. It has a maximum detection distance of 30m. Next, there is the Jetson Nano which offers good performance, particularly in terms of speed and accuracy, for its cost, given how it is considerably cheaper than its counterpart, the Jetson Xavier, and its size is suitable for our purpose. After that, we have the Intel depth camera, which has an effective range of up to 5 meters and also has a built-in IMU (inertial measurement unit), which is suitable for our purposes because it provides an extra set of data allowing for better dense reconstruction.

In terms of software, there also some constraints that need to be taken into consideration, particularly with the computer vision algorithm. The algorithm used is called YOLO (You Only Look Once), and is trained to recognise a wide variety of objects and people. If a higher-quality image is fed into the algorithm, more objects might be detected, but the processing speed of each image would decrease dramatically. Since the objective of the project is to have a similar reaction time to human drivers, speed and accuracy are paramount, and so a balance between image quality and accuracy must be found.

3.3 System Design Overview

The final design should, in simple terms, be able to take input from the camera and LiDAR and run them through the computer vision (CV) algorithm to detect obstacles and identify objects from the input. The computer and its driving algorithm then make a decision whether to continue, decelerate or accelerate, or steer based on output of the CV algorithm. The goals of this project are to make the autonomous car be able to detect all obstacles in its range with enough accuracy to identify obstacles and avoid them, while making decisions in a time comparable to that of the average human driver.

To achieve this goal, the CV algorithm used is the YOLOv5 algorithm, and it will be fed the images from the depth camera and use them to identify people or obstacles to be avoided. As for the LiDAR, an algorithm will be developed to take its input along with the depth information from the depth camera and output a heat map that can tell the decision making algorithm how far the obstacles are and if the car is approaching a wall.

The computer used to process the CV, depth, and decision making algorithms is the Nvidia Jetson Nano. While more powerful models of the Jetson family exist, the Nano offers admirable power for its size, and because of weight and size restrictions in the vehicle, the Nano was the best fit for the project. Since it will be connected to many of the other components in the vehicle, it must be properly powered and ventilated in case the computer starts operating at full load.

Figure 3.7: Autonomous Vehicle Prototype

As shown in Figure 3.7, the vehicle serves as the platform for all the electronic components and the wiring, making it the most important decider of the limiting factors such as weight and size. The Traxxas car chosen has an engine powerful enough to go over 30km/h assuming all its parts are attached. Since the project does not require much of the car's original accessories to be attached, the maximum weight that can be loaded on the car increases. The motor used is also relatively small but powerful, which allows for more components to be placed in the vehicle's platform. The VESC controller will be used to control the car, and will be the interface between the electronic components and the vehicle.

4. FUNCTIONAL MODELING

In this section the upper level and detailed functional modeling of our design will be demonstrated and discussed. Functional modeling is an approach used to gain a better understanding of a project by looking at an overview of the system being designed and what it entails. The outcome of this procedure is an overall view of the system to be able to effectively clarify any ambiguous notions about the design while also being able to analyze and understand the different aspects of the project. In addition, this process also enables the team to see what could be done to improve the project or what it might be missing.

4.1 Upper Level Functional Model

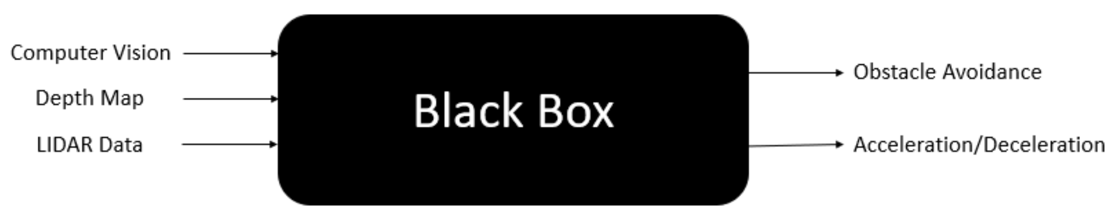

Figure 4.1: Upper level functional model

Figure 4.1 shows the upper level functional model of our proposed design. The black box part of the model, which indicates the system, is the car that will be designed to autonomously drive around. The input to the system is the computer vision algorithm, the depth map obtained from the camera used, and the data obtained from the LiDAR. These parameters enable the car to navigate its way around its environment. At the output, the design is expected to avoid different obstacles that it encounters and accelerate or decelerate depending on the circumstances it is faced with.

4.2 Detailed Functional Model

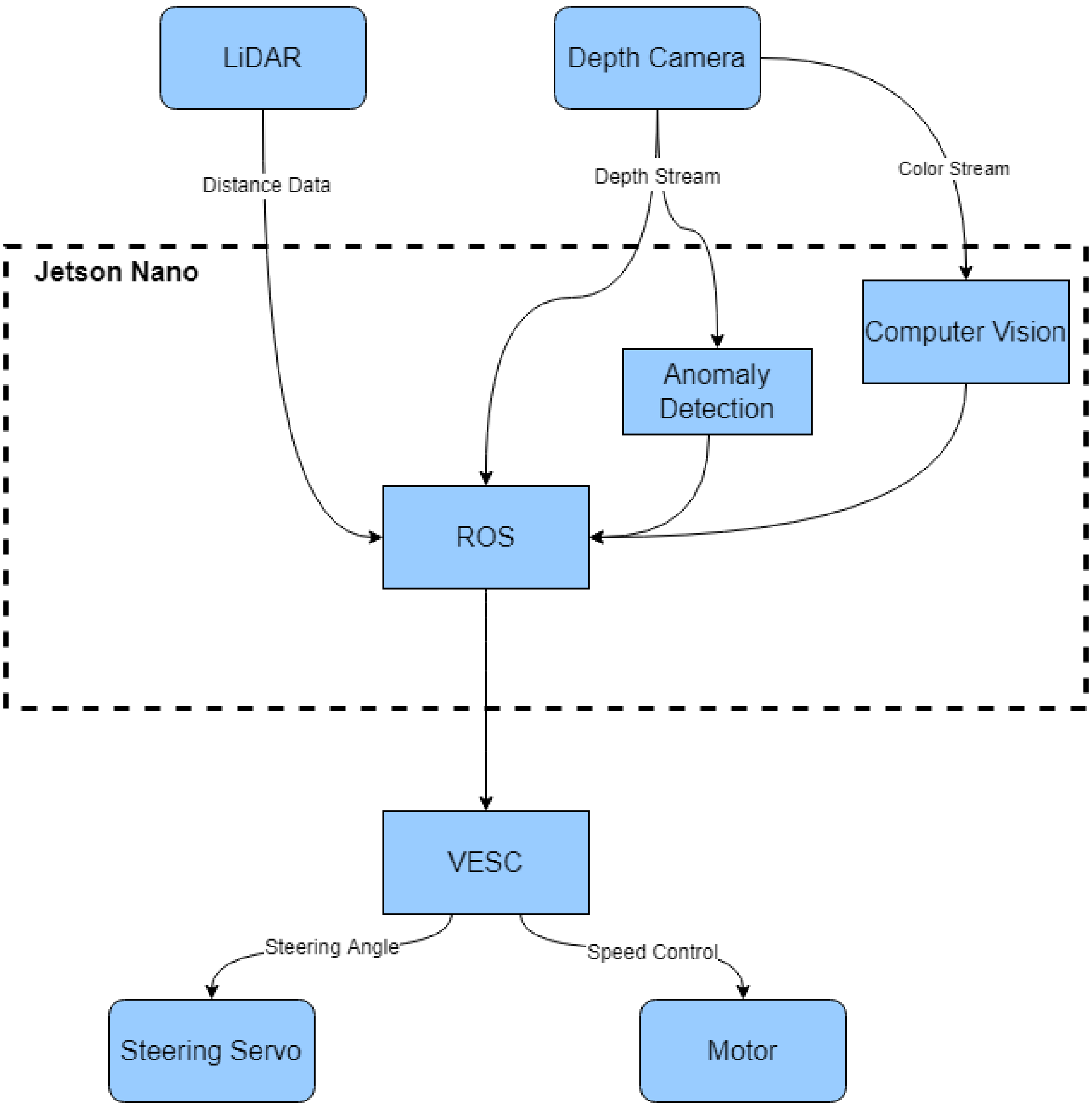

Figure 4.2: Detailed functional model

As shown in Figure 4.2, the LiDAR and the depth camera are the two data sources that will be taken as input for the car to make decisions. The depth camera is also used to acquire RGB data for the computer vision system which will also be taken into account for the decision making process. The decision making is mainly handled by ROS, which is run on the Jetson Nano. Finally, the motion decisions are transferred down to the VESC board, which will then handle the speed control for the motor and the signals sent to the steering servo.

4.2.1 System Operating Procedure

The ROS platform will receive all the input streams from the sensors and process them. Using python programs with the rospy module, the data is analyzed and composited together and ultimately make the decisions on where the car should go. These decisions will be based on autonomous driving algorithms, such as Follow the Gap. This algorithm focuses on avoiding obstacles by always heading towards the widest opening. This algorithm would be mainly dependent on the LiDAR data but also on the stereo camera, specifically the second method that utilizes the stereo camera explained earlier in (Section 3.1.4), as it functions in a similar fashion to the LiDAR. Furthermore, any objects detected by the CV algorithm and any anomalies detected by the anomaly detection function will be considered in the system's decision-making process. The results from the CV and anomaly detection algorithms will be interfaced with ROS through rospy. Using all this information, the system decides the actions that need to be taken to minimize the chance of colliding with an obstacle and sends the high-level decisions on acceleration/deceleration and steering angle to the VESC board. The VESC would then handle the speed control and provide the appropriate current to the motor. At the same time, it will send the signals to the servo motor to set the steering angle as desired.

4.2.2 Additional Features

Based on this model, it is possible to add many additional features by developing more ROS programs without making any changes to the physical design. The only potential change would be to upgrade from the Jetson Nano to a more powerful NVIDIA computer to better handle the extra processing required.

Using computer vision any lanes in the path of the vehicle would be detected, and decisions will be made to keep the car in a lane. Unlike the other sub-functions in our model, lane detection can only rely on the camera, therefore the process used must have a high reliability as sensor fusion cannot be used to verify the data entering the system and help justify the right decision. There are many approaches to lane detection using computer vision. Many of which rely on the Hough Transform. [6][13] First, some filters will need to be applied to the camera feed to improve the input for the lane detection algorithm, such as grayscale and Gaussian filters to reduce noise and smooth frame transitions. Then a Hough Transform is applied to extract features and determine the position of the lanes. It is also possible to use Convolutional Neural Networks (CNN) such as YOLO to detect the lanes similar to how object detection is done. [14] YOLO especially is quick and yields useful results in real-time detection. It may not be the best option for efficient lane detection, but it is an attractive choice given that the team is already planning to use YOLO for object/obstacle detection. Lane detection is an area that might be pursued in the future, but the main focus will remain on obstacle detection with the current design.

5. RESULTS

5.1 Depth Camera

The accuracy of the depth camera's distance measurement was tested. This was done by placing a uniform box at various distances ranging from 50cm to 6m in a well-lit room. The distance to that box was estimated using the depth map generated by the depth camera. The results, shown in Figure 5.1, verified that the camera's effective range ends at approximately 6 meters, and that it has very good accuracy in close-range.

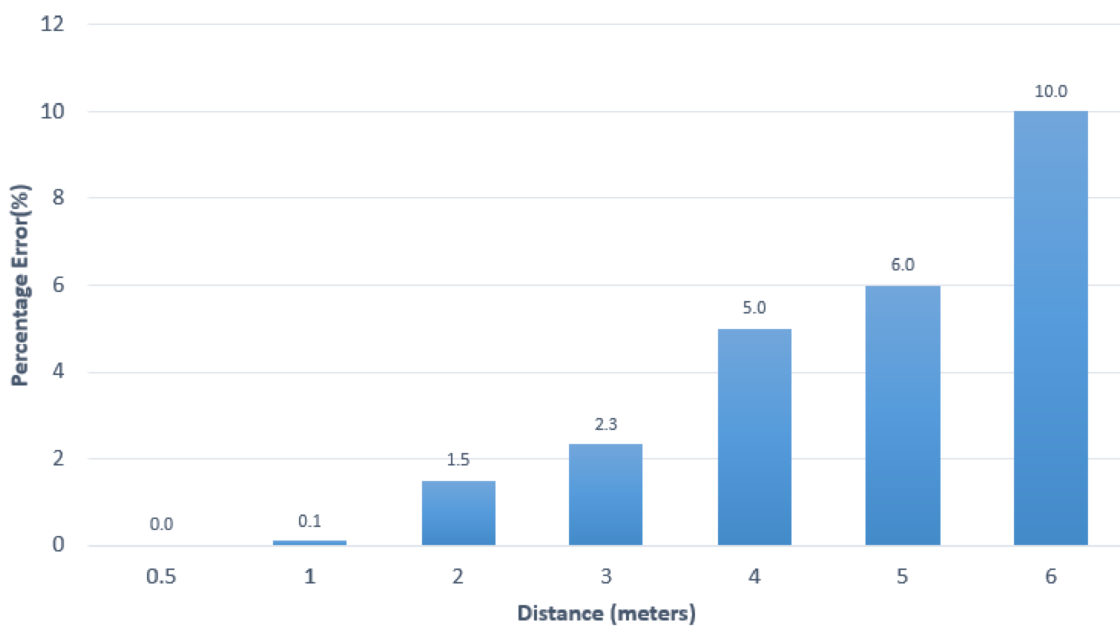

Figure 5.1: Accuracy of distance measurement from depth camera

Two more observations were made. The depth camera consistently overestimated the distance during the accuracy test. The variation in the estimated distance to the object was always significantly lower than the error for that estimated distance. Therefore, it should be possible to compensate for the error in the distance measurement by subtracting the expected error for that specific distance. More testing would need to be performed to accurately apply this, especially in different conditions. An easier and safer solution is to make the car behave more conservatively when encountering long-range data from the depth camera. Finally, a similar test was conducted in low-light conditions and it was found that the percentage error dropped by around half for the same distances used in the first test.

5.2 LiDAR

To visualize the data, the command ‘rviz’ can be run and a map of the LiDAR data will be displayed. By clicking the Add buttons, selecting LaserScan in the topics menu, and typing ‘laser’ in the map frame, the visualization of what the LiDAR can see will be displayed. Figure 5.2 shows an example of what can be seen using ’rviz’, where the red and dark blue lines reflect objects closer to the sensor, and green and cyan lines reflect objects further away from the LiDAR.

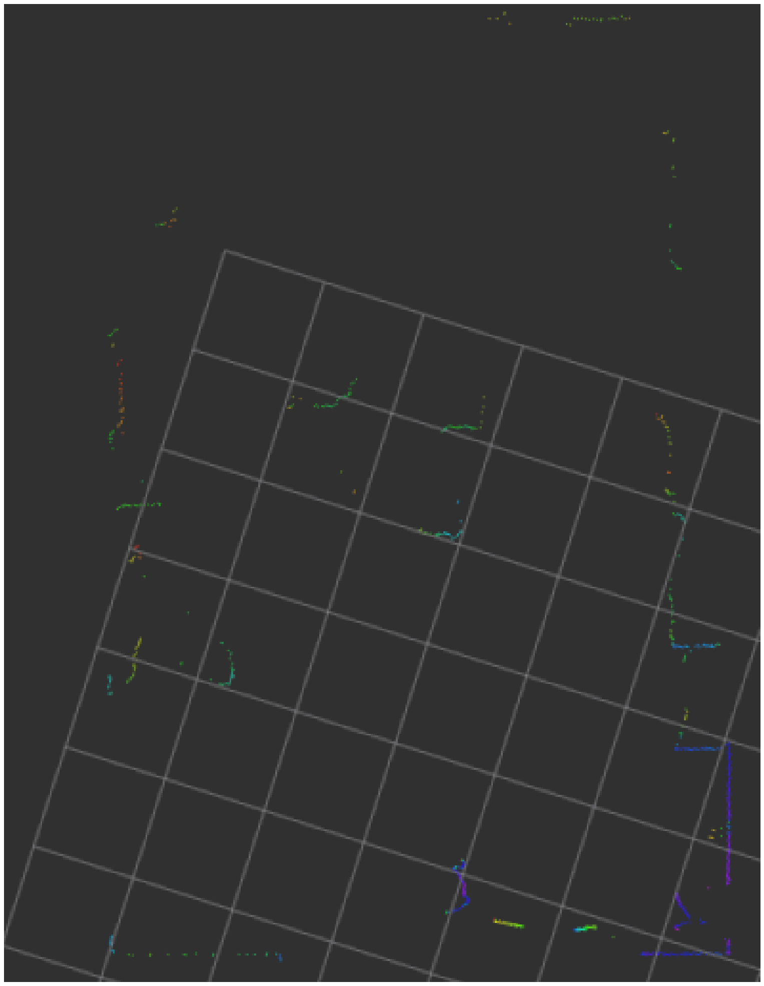

Figure 5.2: LiDAR map showing distances

Another program that can be used to visualize the LiDAR's data is URGBenriPlus, and it helps in better visualizing the scanning angle of the sensor. The program is installed, and with the correct Ethernet settings for the LiDAR connections, the program recognises the sensor with its default IP address.

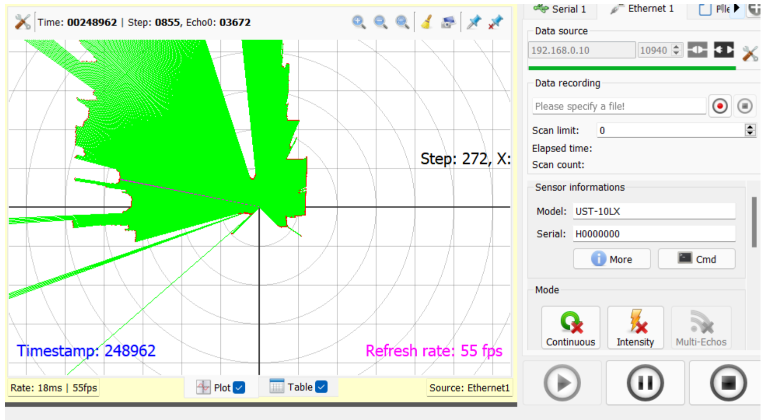

Figure 5.3: URGBenriPlus LiDAR Map

The LiDAR's blind spot of 90° can be seen in Figure 5.3. The red lines on the right side represent the sensor detecting objects close to it, and the lines on the top left that extend beyond the window show that there is no object obstructing the path of the LiDAR. By using both URG-BenriPlus and the rviz ROS module to visualize the LiDAR's output, the scanning angle can be adjusted to not strain the sensor and cause it to overheat.

6. CONCLUSION & FUTURE WORK

While autonomous vehicles seem to be the future of transportation, they are not reliable or safe enough for mass adoption. Our project aims to use a sensor fusion and computer vision system and implement it in a 1/10 scale car. We aim to achieve this by designing and building a mini autonomous vehicle prototype that will utilize both a depth camera and a LiDAR sensor. The project faced many compatibility issues when ordering replacement parts as those found in F1TENTH's bill of materials were either out of stock or overpriced to a point that they exceeded the budget. Given these issues, the project was set back a few times to wait for parts to make modifications to the platform or mount, and so some of the later tasks on schedule were not completed. One of these tasks was to use ROS to autonomously control the car, and to implement tests on the LiDAR to ensure that the best scanning angle and frequency are being used. There were some issues with installing and configuring the ROS packages on the Jetson Nano, and after they were solved, the team was able to start working on the movement code using ROS tutorials. Having already begun the implementation of the software aspects and assembly, the team is on track to set up and test the main sensors to successfully implement the prototype. This implementation, if successful, can then make way for future projects to take place, as many additional methods can be added with the current prototype, without needing to change the hardware. In fact, the team was considering to add some of these methods if there was extra time left after the main goal was achieved.

The main focus of the methods was to have the prototype act like a typical human driver, following road rules. One method is lane detection. As detailed earlier, it can be implemented by using the RGB camera feed, at the cost of some extra processing. Another feature would be stopping by stop signs. By training the object detection model on stop signs (or miniature stop sign props), it should be possible to have the car stop by any stop sign props near its path for a few seconds. It will then only start again when the path is clear. This should all be possible to implement with the current state of the prototype, but the system's performance may suffer, as the

Nano is reaching its limits. Therefore, the Jetson Nano could be upgraded to the Jetson Xavier, which was originally the choice for this project. Other implementations can also include adding more sensors, so that there are alternative navigation approaches, or other directions in which the vehicle can travel instead of just moving forward. This could be done by adding new sensors, such as ultrasonic sensors which were once considered by the team.

REFERENCES

[1] "Global status report on road safety 2018," *Geneva, Switzerland: World Health Organization*, 2018.

[2] A. B. J. Janai, Guney Fatma and A. Geiger, *Computer vision for autonomous vehicles: Problems, datasets and state of the art.* Boston: Now, 2020.

[3] A. D. Kumar, R. Karthika, and K. Soman, "Stereo camera and lidar sensor fusion-based collision warning system," *Advances in Computational Intelligence Techniques*, p. 239, 2020.

[4] C. A. Geiger, F. Moosmann and B. Schuster, "Automatic camera and range sensor calibration using a single shot," *2012 IEEE International Conference on Robotics and Automation*, 2012.

[5] R. J. G. M. L. H. Mohamed, A. and A. Ouda, "Literature survey for autonomous vehicles: sensor fusion, computer vision, system identification and fault tolerance.," *International Journal of Automation and Control*, vol. 12, no. 4, pp. 555–581.

[6] M. Fathy, N. Ashraf, O. Ismail, S. Fouad, L. Shaheen, and A. Hamdy, "Design and implementation of self-driving car," *Procedia Computer Science*, vol. 175, pp. 165–172, 2020.

[7] J. Chen, *SELF-DRIVING ON 1/10 RACER CAR*. PhD thesis, WORCESTER POLYTECHNIC INSTITUTE, 2018.

[8] H. A. W. A. S. T. Betz, J. and M. Lienkamp, "Autonomous driving—a crash explained in detail.," *Applied Sciences*, vol. 9, no. 23, p. 5126, 2019.

[9] "F1tenth - build documentation." Web, 2020.

[10] Hokuyo Automatic Co., *Scanning Laser Range Finder Smart-URG mini UST-10LX Specification*, 3 2015.

[11] NVIDIA, *NVIDIA Jetson Nano System-on-Module*, 2020. v1.0.

[12] NVIDIA, *NVIDIA Jetson Xavier NX Series System-on-Module*, 2021.

[13] A. N. Abbas, M. A. Irshad, and H. H. Ammar, “Experimental analysis of trajectory control using computer vision and artificial intelligence for autonomous vehicles,” 2021.

[14] B. T. Nugraha, S.-F. Su, and Fahmizal, “Towards self-driving car using convolutional neural network and road lane detector,” in *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, pp. 65–69, 2017.