Autonomous Obstacle Avoidance Vehicle using LIDAR and an Embedded System

Nikolaos Baras, Georgios Nantzios, Dimitris Ziouzios, Minas Dasygenis
University of Western Macedonia
Department of Informatics and Telecommunications Engineering
Kozani 50131, Greece
nbaras@outlook.com, gnantzios@gmail.com, dziouzios@uowm.gr, mdasyg@ieee.org

***Abstract*—Autonomous miniature vehicles are widely used to test many types of algorithms and simulate the driving behavior as in the real world. In the last decade, several approaches have been proposed for obstacle avoidance in self-driving vehicles. In this paper, we present our autonomous vehicle implementation which gives a solid solution to this problem in an efficient manner, using a Raspberry Pi and a LIDAR module for indoor navigation. Our vehicle is capable of navigating in an unknown environment while avoiding obstacles. Unlike other implementations, our vehicle doesn't use Computer Vision (CV) techniques for obstacle detection but only a single LIDAR sensor and thus can safely navigate in low luminosity enviroments. We perfomed various experiments and verified the success of our implementation.**

***Index Terms*—Autonomous vehicle, Raspberry Pi, LiDAR, obstacle avoidance, Indoor Positioning System**

I. Introduction

Autonomous vehicles seem to be the next big thing in the automotive industry [1]. The continous expansion of the self-driving technology demands better methods that could reduce the number of accidents [2]. The biggest challenge that researchers are facing in this field, is the safety of the passengers, pedetrians and equipment. One of the biggest difficulties of autonomous vehicles is the ability to react correctly and safely to the eventualities that may occur during driving. Consequently, these vehicles must be equipped with sensors capable of responding very quickly and a processing unit capable of interpreting all this data in real time. Among these sensors are cameras, radars, Global Positioning System (GPS), LIDARs and infrared modules. From all these sensors, LIDAR carries a high implementation complexity that makes it particularly suitable for autonomous driving. The growing interest in selfdriving vehicles has caused researchers to once again seek a solution to the problem of Simultaneous Localization And Mapping (SLAM), a problem that appeared in robotics 25 years ago [3]. The fundamental technology for achieving safety in Autonomous Vehicles is Obstacle Avoidance (OA). We can seperate the OA methods in two main categories: The first category is based on the already known environment (offline processing algorithms) while the second one is based on the gathered sensor information in real time. Our implementation falls in the second category and utilizes an RPLidar[1] and a Raspberry Pi[2].

The proposed methodology provides a solution to the navigation and obstacle detection problem. However, as in every online algorithm, due to the lack of pre-existing knowledge regarding the environment and the obstacles, the optimal path is not guaranteed. The main contributions of our research are: (a) the use of a perception system (LIDAR) for gathering measurements regarding the surrounding environment and obstacle detection, (b) the use of existing Wi-Fi infrastructure for indoor positioning and localization, and (c) a novel algorithm capable of driving the vehicle in real time based on the rapid processing of all gathered information.

The rest of the paper is structured as follows: in section II, we give an overview of reasearch work similar to ours. In section III we present the hardware of our implementation. In section IV, we analyze how our algorithm works and the benefits that it has. Finally, the conclusions and future plans of our research are outlined in section V.

II. Related Work

A vehicle's navigation in a dynamic changing environment, from a starting to an ending point while avoiding obstacles, can be used in many different applications like robotic transfer of goods or taking measurements in an unknown environment. In the last decades, many scientists, researchers, students and hobbyists proposed their different implementations [4]–[8]. In the last few years, this was leveraged by the recent development and improvement of the hardware and software available. Even though some of these are insightful approaches, they have some serious drawbacks such as the high computational cost and high energy requirements. These drawbacks prohibit them from being used on embedded system.

An obstacle avoidance robot using only a single stereo camera as a sensor is proposed by Taihu Pire et al. [9]. A stereo camera is taking left and right images which produces depth maps. Using the captured depth images, the distance of nearby objects is possible to be determined. This research has three major drawbacks as it requires a powerful processor (Intel Core i5 2.67 GHz), a minimum size of object that can be detected and a high latency process.

[1]http://www.slamtec.com/en/lidar/A2
[2]https://www.raspberrypi.org

978-1-7281-1184-1/19/$31.00 ©2019 IEEE

Yan Peng et al. [10] have proposed an algorithm based on 2D LIDAR and verified it on a MATLAB simulation. The basic idea is to use three steps before deciding whether an obstacle is real or not and how to avoid it. These three steps include filtering, preprocessing and clustering. The obstacle avoidance algorithm continually calculates the angle at which the vehicle should turn. The LIDAR SICK LMS511 that is used is very expensive and the MATLAB simulation makes that implementation unsuitable on real environment conditions. Compared to the aforementioned authors, we use a much cheaper and portable LIDAR device and a small embedded platform with low energy consumption.

Another obstacle avoidance module using LIDAR is proposed by Takahashi et al. [11] in which they emphasize on position prediction of moving obstacles. It is based on an embedded microcontroller that predicts the trajectories of obstacles and detects a particular type of objects. The maximum relative speeds of the obstacles towards the mobile robot is limited to 5 km/h.

Widodo Budiharto [12] proposed a surveillance robot, based on a Raspberry Pi and three ultrasonic distance sensors. The obstacle avoidance method is based on a backpropagation neural network, a gradient descent process which brings the network closer to a minimum error with each change. Ultrasonic sensors have a very limited coverage compared to LIDAR, which has 360 degrees of coverage. Also, the training of the robot in this implementation is very slow, causing the robot to malfunction in the early execution.

It is evident from the previous references, that there are multiple approaches from low to high cost hardware. They typically bear high implementation cost and cannot be utilized in a small embedded systems. In our proposed implementation, we utilize an efficient algorithm combined with inexpensive and reliable hardware.

III. System Architecture: Hardware

There are two main hardware components in our implementation for an autonomous vehicle: Raspberry Pi and RPLidar A2 (Figure 1). For obstacle detection, we used a Light Detection and Ranging module (LIDAR). LIDAR is a surveying method that continuously sends pulsed light into every direction (360°) and measures the reflected pulses at every angle with a sensor. Based on the time and wavelength of the received pulse, the distance of the nearest obstacle is calculated. All these measurements need to be processed in order to identify obstacles and perform navigation. For this processing, a Raspberry Pi 3 board is used. Raspberry Pi 3 is a lightweight, powerful and low cost Single Board Computer (SBC) that is very customizable due to its 40 General Purpose Input Output (GPIO) pins. The low power consumption and the plethora of available software make the Pi the best candidate. Furthermore, our full design consists of a small vehicle with motors, motor driver (L293D)[3], Li-Po battery and a USB power bank. Since DC motors use a lot of energy, we intentionally decided to use seperate power sources for the Raspberry Pi and the DC motors, so that in a case of an emergency where the Li-Po battery is completely discharged, we can still communicate with the Raspberry Pi, find its location and receive measurements regarding the surrounding environment. Average power consumption for each component during operation is: 0.65 A for the Raspberry Pi, 1.5 A for the LIDAR module and 750 mA for each DC motor.

[3]https://www.arduino.cc/documents/datasheets/H-bridge_motor_driver.PDF

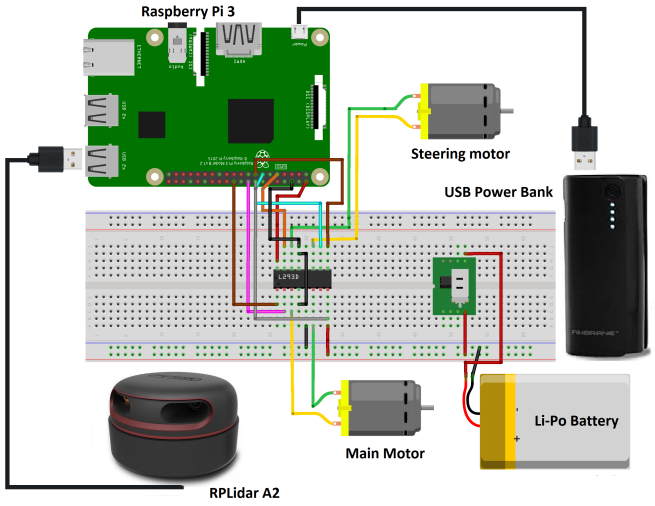


Fig. 1. The main components of our hardware are Raspberry Pi 3 and RPLidar A2.

IV. System Architecture: Software

The Raspberry Pi board supports many operating systems, including Raspbian, Ubuntu and Windows IoT. We decided to use Raspbian because it is specifically developed for the Raspberry Pi hardware and offers better performance compared to the other operating systems. We developed a five stage algorithm for the localization and navigation of the vehicle. The five main stages that constitute our methodology for an autonomous vehicle are: Initialization, Localization, Obstacle Detection, Decision Making and Control (Figure 2). In the first stage, the initialization of our algorithm is performed. Route-specific metrics (starting, intermediate and final waypoints) and environment profiling are loaded as command line arguments. During the algorithm initialization, using as input the relative locations of the vehicle and the installed WiFi stations a calibration procedure is being executed This process occurs only once per mission. In the second stage, our Indoor Positioning System (IPS) implementation provides the position of the vehicle as explained in subsection IV-A. In the third stage (subsection IV-B), we use the installed LIDAR module on the vehicle to detect surrounding obstacles. In the fourth stage (subsection IV-C), gathered information about obstacles is processed by our Decision Making algorithm. The algorithm decides which is the best move for the vehicle. In the last stage (subsection IV-D), the algorithm controls the physical parts of the vehicle (DC motors) in order to move the vehicle in the desired direction.

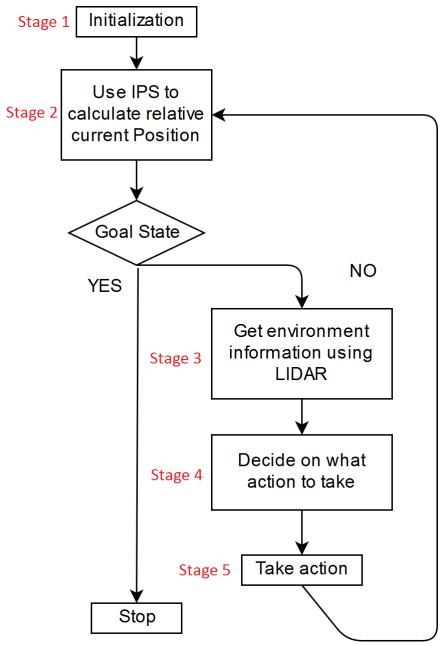


Fig. 2. Our proposed 5 stage algorithm for an autonomous vehicle.

technique as proposed in [14], we can calculate the location of the vehicle relative to the installed BSs. If more than 3 BS are broadcasting at the location of the vehicle, the 3 BS with the highest signal strength are being chosen. In order to calculate the direction of the vehicle, a memory buffer is used. The algorithm compares the position of the vehicle to the previous calculated position every few seconds and determines the direction of the vehicle. For example, at t_0 the vehicle has coordinates (x_0, y_0) and at t_1, (x_1, y_1) (Figure 3). Using our proposed method, we can calculate the direction of the vehicle.

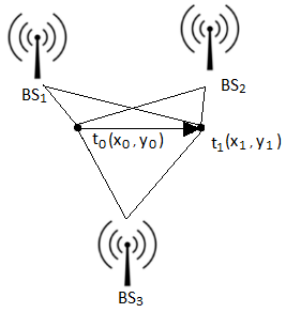


Fig. 3. Calculating the direction of the vehicle based on the change of its position from t_0 to t_1.

A. Indoor Positioning System

In order to compute the vehicle's position and navigate properly, it is necessary to implement an Indoor Positioning System (IPS). Our goal is to avoid the added cost and complexity of extra equipment, so using existing Wi-Fi infrastructure for navigation is the best option. Multiple Wi-Fi Broadcast Stations (BS) are located in the Area of Interest (AoI) that the vehicle has access to. We consider a typical scenario in an indoor environment that has transmissions of multiple BS from different locations. This scenario is not far from reality because modern workplaces house multiple SSIDs (Service Set Identifier). In order to calculate the exact position of the vehicle at any given time, it is necessary that the vehicle has contact with three or more BS. The relative position of all the BS, the Starting Position (SP) and the Goal Position (GP) of the vehicle are known. Received Signal Strength (RSS) is the actual signal strength at the receiver, measured in miliWatts (mW) or decibel-milliwatts (dBm). Higher RSS values mean smaller distance between Transmitter (Tx) and Receiver (Rx). RSSI distance measurement generally uses the logarithmic distance path-loss model [13]. It is expressed in (Eq. 1).

$$RSSI = -10n\log\left(\frac{d}{d_0}\right) + A + X_\sigma \tag{1}$$

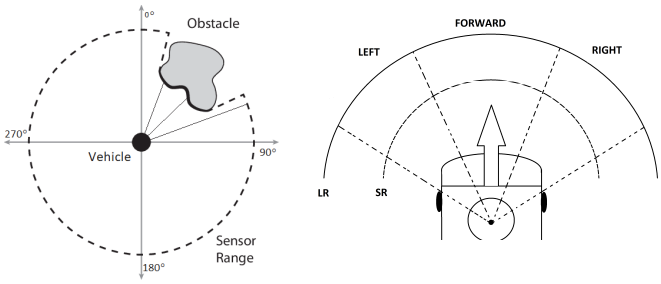
where d is the distance between the transmitter and the receiver, n is a path-loss parameter related to the specific wireless transmission environment, X_σ is a Gaussian-distribution random variable and A is the RSSI with distance d_0 from the transmitter. The more obstacles there are, the larger n will be. The RSSI approach provides us only with the distance of the vehicle relative to each BS. Using a simple triangulation

B. Obstacle detection

The RPLidar A2 module we used, captures 8,000 measurements per second from 0° to 360° with a range up to 18 meters. LIDAR returns these measurements in a point cloud. Every data point is the reflection of an obstacle and carries three numbers: degrees, distance and signal strength. Based on the degrees, we categorize measurements in 4 classes: Forward, Left, Right and Back. For each class, we divide measurements in 2 categories: Long Range (LR) and Short Range (SR) measurements. Measurements between 0 cm and 110 cm are placed in the SR category while measurements between 111 cm and 1800 cm are placed in the LR category (Figure 4). Measurements in SR are weighted more compared to those in LR. We perform a preprocessing filtering to speed up the computations by removing measurements that are not currently required. For example, when the vehicle is moving forward, the important measurements are located in the front category. Measurements and obstacles in other directions are not used, therefore we have to filter them to improve the processing speed of the algorithm.

C. Decision making algorithm

The goal of the algorithm is to provide the best decision possible for the vehicle at any given time in order to reach its destination. It takes into account multiple factors, such as: vehicle direction, motor state, direction of GP and surrounding obstacles. The algorithm drives the vehicle to the direction of the GP until an obstacle is detected. When the vehicle cannot keep going forward due to a detected obstacle, the algorithm then checks left and right to determine the possibility of a



(a) (b)

Fig. 4. In (a) we see LIDAR detecting an obstacle. In (b) we categorize collected data based on measurement location.

turn. If both left and right directions are free of obstacles, the algorithm favors the direction closer to the Goal Position. If every direction is blocked, the algorithm drives the vehicle in the reverse direction while simultaneously checking for a different possible turn (Figure 5).

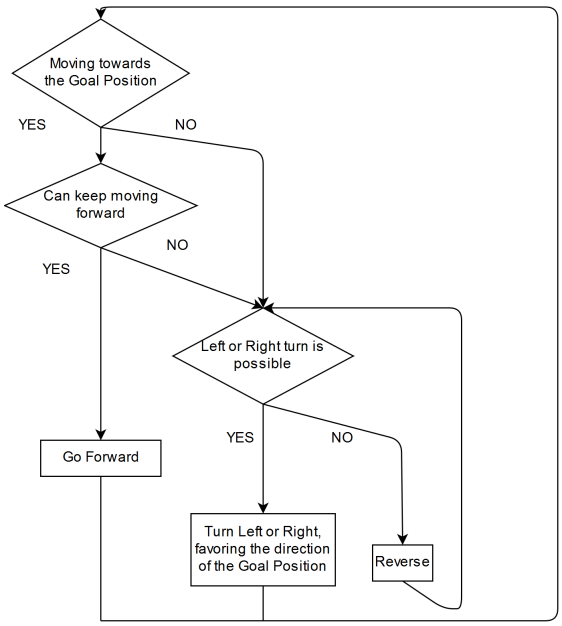


Fig. 5. Our Decision Making Algorithm.

D. Taking action

In the end, every decision the algorithm makes comes down to the movement of the two DC motors installed on the vehicle. In order to control these motors, we use the RPi.GPIO Python Library to set the desired values on the appropriate GPIO pins. These pins are directly connected to the installed L293D motor driver which controls the DC motors.

V. Experimental Results

In order to evaluate the functionality of vehicle, we performed various tests in our University's laboratory. The dimensions and the area of the testing environment were $5m \cdot 14m$ and $70m^2$ respectively. In total, there were 6 BS installed: one at the SP, one at the GP and 4 more strategically placed in order to guarantee the complete coverage of the area. There were multiple obstacles of various sizes located in random places in the environment. The vehicle managed to navigate from the SP to the GP in $5min$ and $31sec$. The average speed of the vehicle was $0.31m/sec$. The average power consumption was 9.5 W for the Raspberry Pi and LIDAR, and 7.31 W for the installed DC motors.

VI. Conclusions and Future Work

In this paper, an autonomous vehicle using Raspberry Pi and LIDAR is presented. The vehicle is capable of utilizing existing Wi-Fi infrastructure in order to locate its position in the environment and navigate safely from a starting point to a destination point while avoiding obstacles. Our work could be further enhanced by implementing a mapping feature in order to generate an image of the environment during navigation. Modifications on the algorithm could also be made in order to predict obstacle movement and navigate in a more efficient manner in dynamic environments.

References

[1] R.-S. Run and Z.-Y. Xiao, "Indoor autonomous vehicle navigation—a feasibility study based on infrared technology," *Applied System Innovation*, vol. 1, no. 1, 2018.

[2] T. Litman, *Autonomous Vehicle Implementation Predictions: Implications for Transport Planning*, ser. desLibris: Documents collection. Victoria Transport Policy Institute, 2013.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] K. Bimbraw, "Autonomous cars: Past, present and future - a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings*, vol. 1, pp. 191–198, 2015.

[5] Q. Memon, M. Ahmed, S. Ali, A. Rafique, and W. Shah, "Self-driving and driver relaxing vehicle," 2016, pp. 170–174.

[6] H. Flämig, *Autonomous Vehicles and Autonomous Driving in Freight Transport*, 2016, pp. 365–385.

[7] Q. Memon, M. Ahmed, S. Ali, A. Rafique, and W. Shah, "Self-driving and driver relaxing vehicle," 2016.

[8] B.-C.-Z. Blaga, M.-A. Deac, R. W. Y. Al-doori, M. Negru, and R. Danescu, "Miniature autonomous vehicle development on raspberry pi," in *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2018.

[9] T. Pire, P. De Cristóforis, M. Nitsche, and J. Jacobo Berlles, "Stereo vision obstacle avoidance using depth and elevation maps," 2012.

[10] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-d lidar," in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 1648–1653.

[11] M. Takahashi, K. Kobayashi, K. Watanabe, and T. Kinoshita, "Development of prediction based emergency obstacle avoidance module by using LIDAR for mobile robot," in *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, 2014.

[12] W. Budiharto, "Intelligent surveillance robot with obstacle avoidance capabilities using neural network," *Computational Intelligence and Neuroscience*, vol. 2015, pp. 1–5, 2015.

[13] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, "Indoor positioning algorithm based on the improved rssi distance model," *Sensors*, vol. 18, p. 2820, 2018.

[14] N. A. Mahiddin, "Indoor position detection using wifi and trilateration technique," *The International Conference on Informatics and Applications (ICIA2012)*, pp. 362–366, 2012.