Obstacle Detection and Avoidance Vehicular Bot

Mrs. Shruthi.J
Asst Professor, Department of Computer Science
BMS Institute of Technology and Management
Bangalore, Karnataka 560064, IN

Mr. Ravi Kumar B N
Asst .Professor Department of Computer Science
BMS Institute of Technology and Management
Bangalore, Karnataka 560064, IN

Mrs. Sumathi M S
Asst .Professor Department of Telecommunication
BMS Institute of Technology and Management
Bangalore, Karnataka 560064, IN

Mrs. Suma Swamy
Professor *Department of Computer Science*
Sir M Visveswaraya Institute of Technology
Bangalore, Karnataka 560064, IN

***Abstract* - This paper presents a unique design of a driverless, collision protected bot that can navigate across any given space without any human intervention. The bot is built on the raspberry pi board using the Raspbian OS. The bot uses the servo motor board for movement from one place to another. The camera module is responsible for detecting obstacles on the path. The steering module, comprising of servo motor and wheels is used for positioning the wheel in normal cases and in case of any obstacle detection. The camera module is trained using the tensorflow zoo model on the modified YOLO algorithm.**

***Index Terms* - Self-Driving Bot, YOLO, obstacle detection, avoidance**

I. INTRODUCTION

The usage of motor vehicles has increased exponentially in the past few decades. With increasing traffic, commute timings have risen rapidly along with the number of accidents everyday. One solution for these issues would be an autonomous car or a vehicular bot. Throughout the past decade, we have witnessed one of the greatest strides in automobile technology with the focus on autonomous cars. Recently self-driving cars are coming up in the news, though the researchers have been experimenting with self-driving vehicles for over 45 years. These vehicles require a broad range of technologies and infrastructures to operate properly.

A vehicular bot (driverless, autonomous, robotic car) is a vehicle that can sense its environment and navigate without human input. Vehicular bots can detect environments using a variety of techniques such as radar, GPS and computer vision. Such advanced systems use data to find the best route, detect obstacles and suggest the best route around these obstacles. This is very useful in planning a path to the desired destination. The vehicular bot uses AI technology. Developers of such bots use vast amounts of data from image recognition systems, along with machine learning and neural networks, to build systems that can drive autonomously. The neural networks identify patterns in the data, which is fed to the machine learning algorithms. That data includes images from cameras on from which the neural network learns to identify as traffic lights, trees, curbs, pedestrians, street signs and other parts of any given driving environment.

For any mobile bot, the ability to navigate in its environment is one of the most important capabilities. The navigation task can be divided into three main tasks: Obstacle detection, obstacle avoidance, and path planning. The potential application areas of the autonomous navigation of mobile robots include automatic driving, guidance for the blind and disabled, exploration of dangerous regions, transporting objects in factory or office environments, collecting geographical information in unknown terrains like unmanned exploration of a new planetary surface, etc.

Movement is taken care of by the servo motor board which has a DC motor and is connected to the wheels. The servo motor aligns the wheels according to the required wheel odometry. Obstacles are detected using sensory input from the raspberry Pi camera attached to the raspberry pi board. The input is fed to the YOLO algorithm which tries to identify the obstacle. Ultrasonic sensors present are used to measure the distance between the bot and the obstacle.

II. BLOCK DIAGRAM

The architectural design of the system is shown in the following block diagram shown as figure1.

The architecture of the proposed system has three modules. These modules are used to detect objects present in the immediate vicinity and plan out a path which circumnavigates around the obstacles thereby allowing the bot to move without any human intervention.

The three modules are: Obstacle Detection module, Obstacle Avoidance module, and steering module. Like the name suggests Obstacle detection is responsible for obstacle detection i.e to determine the presence of an obstacle. Obstacle avoidance is used for distance calculation and circumnavigation. The steering module is used for determining the path to take to avoid collision.

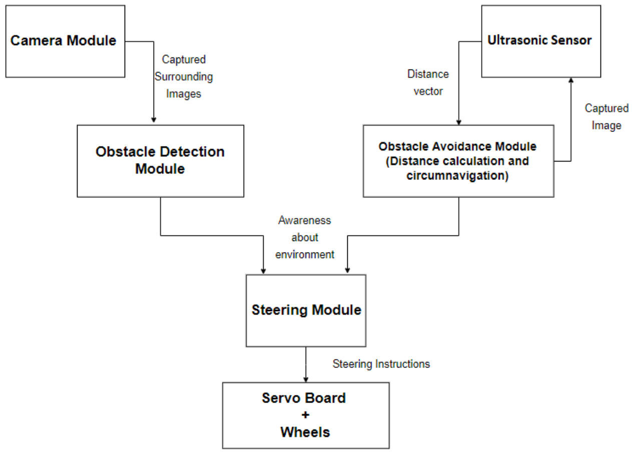

Fig. 1 Architecture Diagram of the prototype

A. *Obstacle Detection module*

The Obstacle detection module is the main module of the prototype. The Obstacle detection modules uses machine learning techniques to identify the presence of an object and to classify the object type.

It uses the modified YOLO algorithm. CNN's were not used as we cannot predetermine the number of output neurons. Also, since the prototype is built on a simple Raspberry Pi board, a CNN would prove much more computationally vexing and would require lots of power, especially during training.

The machine learning model, the modified YOLO is trained using the google tensorflow model zoo. The model Zoo provides us with around 5 datasets with around 10,000 images in each. The YOLO algorithm, upon identifying an obstacle, frames the obstacle, provides the name of the obstacle if it can recognize the object.

B. *Obstacle Avoidance module*

The obstacle avoidance module is the module responsible for the next set of actions to be taken once an obstacle is detected. As soon as the obstacle detection module detects an obstacle, the obstacle avoidance module triggers the ultrasonic sensor. The ultrasonic sensor has parameters set to observe the distance between the prototype and the obstacle.

The parameters change with the change in distance. Once the parameter reaches a certain threshold, the obstacle avoidance module gives control to the servo motor to decide the path to choose. The ultrasonic sensor parameter should be set with a little buffer since controlling the prototype at certain speeds can be challenging.

C. *Steering module*

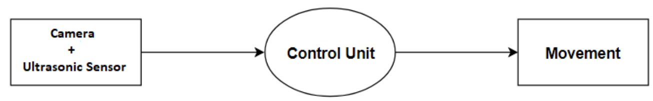

Fig. 2 Level 0 DFD

The Steering module consists of two major components: Servo motor board and wheels. The ultrasonic sensor at a certain distance will trigger the path planning program which is used to determine the path that the prototype has to take.

Here we have input from the previous two modules, which is, the obstacle ahead of us from the obstacle detection module and the ultrasonic sensor gives us the distance from the obstacle. With this data the prototype can build an awareness of the surrounding which can then be used to make decisions about the direction in which it should move.

The servo motor makes use of the regular DC motor attached to it and couples it with the given sensory information for position feedback. Depending on its awareness of the surroundings the position is decided and sent to the wheels for alignment.

III. Component Design

The system is divided into the following components: Sensory inputs - Pi camera and Ultrasonic Sensor input, Obstacle detection, distance computation and Obstacle avoidance.

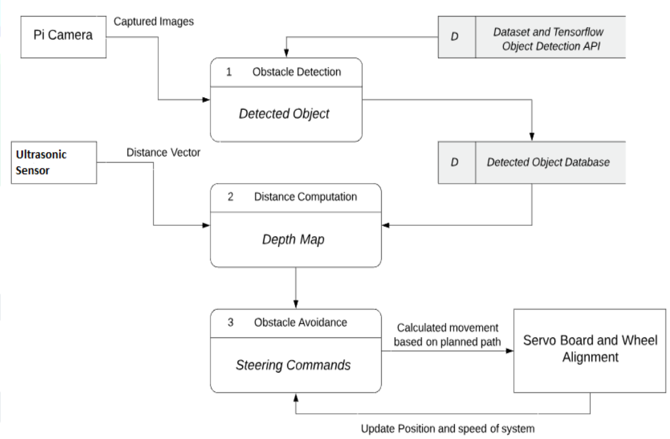

Fig. 3 Level 1 DFD

The above figure illustrates the flow of information between the various processes and components present in the system in the form of a data flow diagram.

The following figure gives the overall component design. The sensory input from the Pi camera is given to the obstacle detection module. The obstacle detection module trained using the Tensorflow obstacle detection training model using the COCO dataset. The captured image is given to this trained module which then identifies the object. Information about the obstacle is given to the steering module.

The ultrasonic sensor computes the distance between the obstacle and the prototype. A distance vector is formed which contains the distance of the object from various obstacles present in its surroundings. Using this a depth map is formed. A depth map, is like a heat map which is used to build an awareness of the surrounding.

The data from both these modules is then fused and the data is used for path planning. The servo motor control board sends instructions to the wheels arranging its wheel odometry. The motor makes decisions as to move right or left.

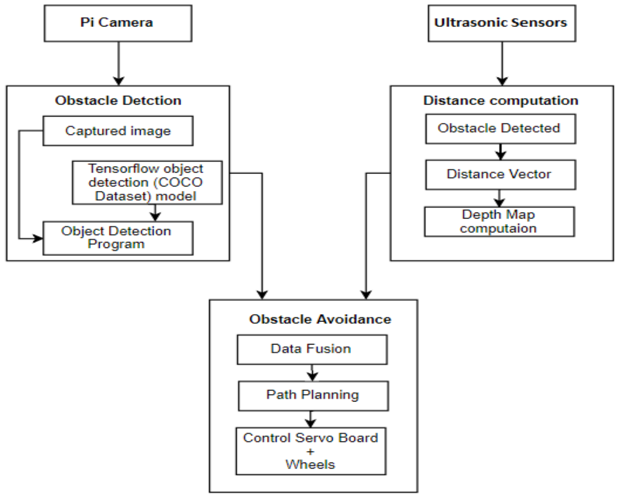

Fig. 4 Component Design

IV. Results

The result is a prototype that can move on its own in any given space. The prototype can detect obstacles on its way and can make intelligent decisions to avoid collisions. The prototype can also detect multiple obstacles at once by creating distance vector and by computing depth maps.

V. Conclusions

The project was successful in achieving its aim of developing a low-cost, economical, and algorithmically strong obstacle detection and avoidance vehicular bot.

The output of the obstacle detection module is a set of frames outlining the obstacles, which we fuse into a consistent grid representation that indicates free and occupied space. The set of detected obstacles along with the understanding of the surroundings (found by gauging the distance between the obstacles and the system, using the ultrasonic sensor) can be used for detecting collisions and thereby plan out a path of movement. Due to the nature of the input peripherals, especially the camera, this system is extremely time sensitive so that computations had to be optimized as much as possible for the car to be able to react and respond with proper movements in real time.

The project can be further expanded to incorporate geo-navigation and give it the ability to be also controlled by instructions from the user. As of now the default direction of movement is forward. This needs to be configured such that the user can have a hand in guiding the direction. The instructions can be of any form, preferably voice.

We can also try tackling a much larger scale of project and obstacles. Currently the system focuses on low lying small sized obstacles. Work can be done to expand the system's line of sight, along with making it more reliant to all kinds of travel terrain.

References

[1] Christian Hane, Torsten Sattler, and Marc Pollefeys, "Obstacle Detection for Self-Driving Cars Using Only Monocular Cameras and Wheel Odometry", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept 28 - Oct 2, 2015

[2] Jing Tao, Hongbo Wang, Xinyu Zhang, Xiaoyu Li, and Huawei Yang, "An Object Detection System Based on YOLO in Traffic Scene", in 6th International Conference on Computer Science and Network Technology (ICCSNT), 2017

[3] Lohit Ujjainiya and M. Kalyan Chakravarthi, "Raspberry pi-based cost-effective vehicle collision avoidance system using image processing", in ARPN Journal of Engineering and Applied Sciences, vol 10, No 7, 2015

[4] R. Surya Kumar, P. Kingston Stanley, A. Sanjeevi Gandhi, "Automated Vehicle Collision Avoidance and Navigation Using Raspberry Pi", in International Journal of Engineering and Techniques - Volume 3 Issue 4, July-Aug 2017

[5] Yijia He, Yue Guo, Aixue Ye, and Kui Yuan, "Camera-Odometer Calibration and Fusion using Graph Based Optimization", in Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics December 5-8, 2017, Macau SAR, China

[6] Sebastian Ramos, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother, "Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling", in 2017 IEEE Intelligent Vehicles Symposium (IV) June 11-14, 2017

[7] Hailan Kuang, Kaiwei Zhang, Ruifang Li, and Xinhua Liu, "Monocular SLAM Algorithm Based on Improved Depth Map Estimation and Keyframe Selection" in 10th International Conference on Measuring Technology and Mechatronics Automation, 2018

[8] T.-J. Lee, D.-H. Yi, D.-I. Cho, "A Monocular Vision Sensor-based Obstacle Detection Algorithm for Autonomous Robots", in Sensors, Vol. 16, No. 3, 311, 2016.

[9] Yasir Dawood Salman, Ku Ruhana Ku-Mahamud, & Eiji Kamioka, "Distance Measurement for Self-Driving Cars using Stereo Camera" in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 235-242), 2017

[10] Christian H¨ane, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, Marc Pollefeys, "3D Visual Perception for Self-Driving Cars using a Multi-Camera System: Calibration, Mapping, Localization, and Obstacle Detection" in Image and Vision Computing, September 1, 2017.