Practical Implementation of Obstacle Detection and Obstacle Recognition

Datta Lohith Gannavarapu
University of Maryland
College Park
gdatta@umd.edu

Vishnu Mandala
University of Maryland
College Park
vishnum@umd.edu

Kasturi Charan
University of Maryland
College Park
charan03@umd.edu

Hamza Shah Khan
University of Maryland
College Park
hamzask@umd.edu

Rishabh Namdev
University of Maryland
College Park
rnamdev@umd.edu

Abstract

Obstacle detection and recognition are critical tasks in robotics and automation, ensuring safe and efficient operation across various industries. In this project, we propose the development of an integrated obstacle detection and recognition system using a customized Obstacle Detection pipeline. The system utilizes advanced algorithms, such as YOLO (You Only Look Once), to detect and identify obstacles in real-time within indoor environments. By combining custom Obstacle detection and recognition techniques, the mobile robot will be able to not only detect obstacles but also accurately classify and recognize them. The project aims to leverage computer vision and the creation of a custom dataset to enhance the robot's perception capabilities, enabling it to navigate complex environments effectively and avoid potential hazards.

Keywords: obstacle detection, obstacle recognition, custom obstacle detection, custom obstacle recognition, YOLO, real-time applications, computer vision, dataset creation.

1. Introduction

Object detection and recognition are essential tasks in various domains, including robotics, surveillance, and autonomous systems. In this project, we introduces a novel approach for real-time object detection and recognition using custom datasets and the YOLO (You Only Look Once) algorithm. Custom datasets are created with annotated bounding box coordinates and class labels for diverse obstacles of interest. The YOLO algorithm, known for its speed and accuracy, is employed to train the obstacle detection model on these datasets. The hardware setup includes a high-performance single-board computer integrated with cameras and sensors, while software implementation utilizes computer vision libraries and YOLO-specific frameworks.

In this project, we implement the YOLO algorithm for real-time obstacle detection, leveraging its high accuracy and processing speed. YOLO treats object detection as a regression problem, predicting bounding boxes and class probabilities in a single pass over the input image. This unified detection approach eliminates the need for iterative computations or region proposal methods, resulting in real-time performance and efficient resource utilization. This approach allows for real-time performance, making YOLO suitable for applications that require quick and accurate object detection. The integration of YOLO into our project will enable precise object recognition in images and videos, paving the way for advancements in robotics, surveillance, and autonomous systems.

The development of robust and efficient object detection algorithms has revolutionized various domains, including robotics, surveillance, and autonomous systems. In this project, we focus on implementing the You Only Look Once (YOLO) algorithm, a state-of-the-art real-time object detection algorithm, to enable precise object recognition in images and videos.

The practical implications of this project are significant, as it enables the creation of custom object detection and recognition systems tailored to specific application domains. The developed system can be deployed in various real-world scenarios, including robotics, surveillance, and interactive systems, where real-time object detection and recognition are crucial.

1.1. Pipeline

Our project aims to leverage the capabilities of the YOLO algorithm in the context of a 4-wheeled robot equipped with object detection and recognition capabilities. By integrating YOLO into the robot's system, we can enhance its ability to perceive and interact with its environment autonomously. The fast and accurate object detection provided by YOLO will enable the robot to identify and track objects of interest in real-time, facilitating tasks such as obstacle avoidance, object manipulation, and environment mapping.

In this report, we present the implementation and evaluation of the YOLO algorithm within the context of our autonomous 4-wheeled robot. We discuss the mechanisms and mechanical aspects of the robot, including the drive mechanism, steering mechanism, wheel configuration, and suspension system. We also highlight the key principles and characteristics of the YOLO algorithm and explain how it enhances the robot's object detection and recognition capabilities. The Pipeline for the project is depicted in Figure 1.

By leveraging the power of the YOLO algorithm, we aim to develop an autonomous 4-wheeled robot that can efficiently detect and recognize objects in real-time, enabling it to navigate and interact intelligently with its surroundings. Through this project, we aim to contribute to the advancement of object detection techniques and their practical applications in autonomous systems.

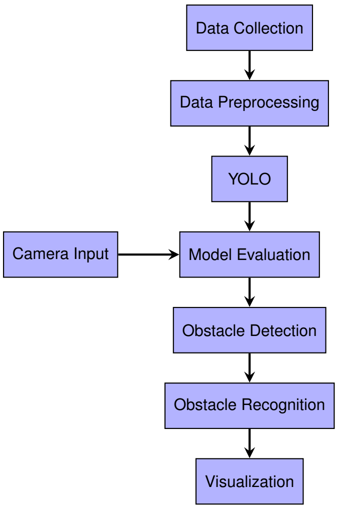

Figure 1. Obstacle Detection and Recognition Pipeline

2. Hardware Implementation of a Mobile Robot using Differential Drive

In this project, we implement the hardware implementation of a mobile robot using a differential drive system. The differential drive mechanism allows the robot to navigate and control its motion by independently controlling the speed and direction of its two wheels.

2.1. Differential Drive System

The differential drive system consists of two independently controlled wheels, each driven by a separate motor. The robot's motion is controlled by varying the speeds of the two wheels in different directions. The speed and direction of the robot can be controlled using the following equations:

$$V = \frac{R}{2}(v_r + v_l) \tag{1}$$

$$\omega = \frac{R}{L}(v_r - v_l) \tag{2}$$

where V is the linear velocity of the robot, ω is the angular velocity of the robot, R is the radius of the wheels, L is the distance between the wheels, v_r is the speed of the right wheel, and v_l is the speed of the left wheel. An illustration of the Kinematics of the mobile Robot is depicted in the Figure 2.

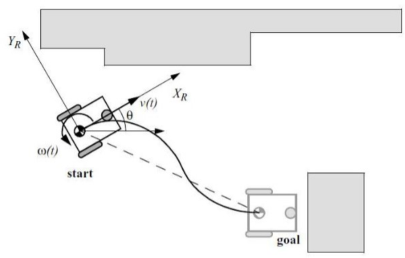

Figure 2. Kinematics of the Mobile Robot

2.2. Robot Design

Figure 3 illustrates the design of our mobile robot. The robot consists of a chassis, four motorized wheels, a Raspberry Pi for running the obstacle avoidance algorithm,an onboard microcontroller for IMU data, and various sensors such as Ultrasonic Sensor for perception and environment interaction. The wheels are driven by DC motors, and the speed and direction of each wheel are controlled using pulse-width modulation (PWM) signals from the Raspberry Pi using a L298N motor driver.

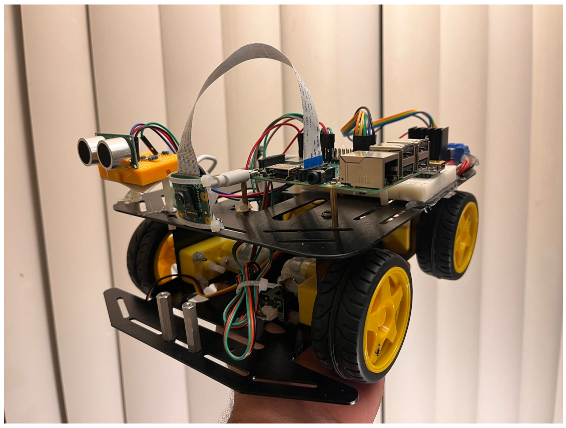

Figure 3. Hardware design of the mobile robot using differential drive.

2.3. Implementation

To implement the differential drive system, we used four DC motors, with an encoder on one of the motors on each side. The encoders provide feedback on the rotation of each wheel, which is used for closed-loop control of the robot's motion. The microcontroller reads the encoder signals, calculates the required speeds for each wheel based on the desired linear and angular velocities, and generates the PWM signals to control the motors. We integrated various sensors, such as an RGB camera(RPi Camera v2.1), an Ultrasonic proximity sensor(RCWL-1601), and an inertial measurement unit (IMU - BNO055), to enhance the robot's perception capabilities and improve its navigation and obstacle avoidance capabilities.

The Control Loop is as depicted in Figure 4. The IMU provides real-time data on the robot's orientation, including its roll, pitch, and yaw angles, as well as its linear accelerations. This information is fed into the control algorithm running on the Raspberry Pi, which calculates the desired wheel speeds for the differential drive system. The control algorithm compares the desired orientation and accelerations with the measured values from the IMU, and adjusts the wheel speeds accordingly to maintain the desired motion.

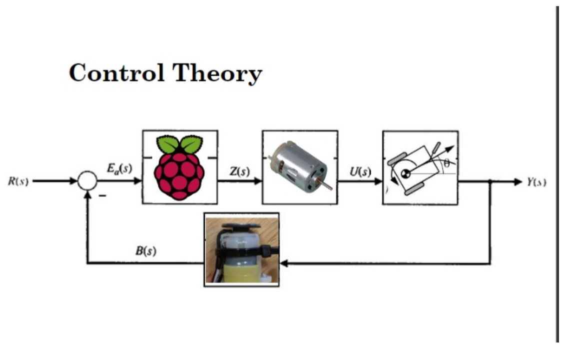

Figure 4. Control loop of the Mobile Robot

2.4. Experimental Results

We conducted several experiments to evaluate the performance of our mobile robot. Figure shows the trajectory of the robot as it navigates through a predefined path. The robot successfully follows the desired path, demonstrating accurate control of its motion using the differential drive system.

We also measured the robot's odometry and compared it with the ground truth. Figure 5 shows the comparison between the estimated position and the ground truth position of the robot. The results indicate that the differential drive system provides reliable odometry estimation for the mobile robot.

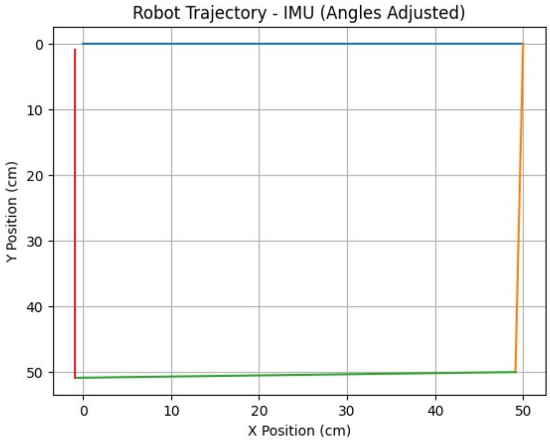

Figure 5. Robot Trajectory using IMU

Our hardware implementation of a mobile robot using differential drive has shown effective control of the robot's motion. The differential drive system allows for smooth and precise maneuvering, enabling the robot to navigate through complex environments.

By integrating various sensors, we enhanced the robot's perception capabilities, enabling it to sense its surroundings and interact with the environment effectively. The use of encoders for closed-loop control improved the accuracy of the robot's motion and provided reliable odometry estimation.

The experimental results demonstrate the successful implementation of the differential drive system. The robot accurately followed predefined paths and achieved consistent odometry estimation, showcasing the effectiveness of the hardware design.

3. Conclusion

Future work will focus on extending the capabilities of the robot by integrating additional sensors, improving perception algorithms, and implementing higher-level control strategies to enable autonomous navigation and intelligent decision-making.

4. Code and Datasets

Please refer to the github link: Code and Datasets