Available online at http://icsejournal.com/index.php/JCSE
Journal of Computer Science an Engineering (JCSE)
e-ISSN 2721-0251
Vol. 5, No. 1, February 2024, pp. 20-38

Color-Based Image Processing for Autonomous Human-Following Trolley Robot Navigation with Camera Vision

Budi Artono[1]*, Widya Nugroho[2], Rizki Wahyudi[3]

[1,2]Department Electrical Engineering, State Polytechnic of Madiun, Madiun, Indonesia.
[3]Department Informatics, Faculty of Computer Science, Universitas Amikom Purwokerto, Purwokerto, Indonesia.
[1]budiartono@pnm.ac.id*; widndutgendut@gmail.com; rizki.key@gmail.com;
*corresponding author

ARTICLE INFO

Article History:
Received September 8, 2023
Revised January 21, 2024
Accepted February 6, 2024

Keywords:
Image Processing
Robotics
Trolley Robot

Correspondence:
E-mail: budiartono@pnm.ac.id

ABSTRACT

The rapid advancements in the field of robotics have spurred intensive research, particularly in the industrial sector, aiming to develop robots that can assist in simplifying daily human tasks. One emerging area of research involves the design of a cargo-carrying robot trolley. This trolley robot has the capability to follow a person carrying items by recognizing the color of the clothes worn by that person through image processing. The objective of this research is to facilitate the transportation of goods, especially in airport environments, by enabling the robot to identify and follow human objects with a minimum distance of 30 centimeters and a maximum distance of over 3 meters. The design system of this robot trolley utilizes a camera sensor to detect the object to be followed through image processing using OpenCV on the Microsoft Visual Studio 2012 platform. The image processing results in PWM values sent to the Arduino to drive DC motors. Additionally, ultrasonic sensors are employed to restrict the robot's movement in its surroundings, preventing collisions. The robot's speed can adjust according to the walking speed of a person. If the robot is moving too fast, it will be stopped by the ultrasonic sensor when the distance between the robot and the person being followed is less than 30 cm, avoiding collisions between the robot and the person.

1. Introduction

In the rapidly advancing era of robotics, continuous innovation is actively sought to make significant contributions to everyday life. One innovative concept currently under development is the human-following robot [1], [2], a mechanical entity with the ability to adapt its movements to those of a human. Several studies related to the development of a human-following robot trolley have been conducted with the aim of assisting in carrying excess items that are challenging for humans to transport.

The presence of human-following robots is becoming increasingly important, especially in the context of transporting excess goods [1]. By utilizing image processing technology using cameras, hardware such as the Arduino Mega 2560, and ultrasonic sensors, we designed a robot trolley that can automatically follow humans [2]. The primary application of this robot can be found at airports, where flat and smooth terrain allows the robot to operate efficiently, assisting passengers in carrying heavy luggage [3].

As a development foundation, the research by Suprobo et al. [3] discusses the development of an electric trolley prototype at the airport terminal to enhance baggage transportation efficiency. This trolley can carry a load of up to 130 kg with a 5 AH battery that lasts approximately 4 hours and 27 minutes without a load and 3 hours and 5 minutes with an 89 kg load. The low speed of the trolley is 3.3 km/h, and it can stop within a distance of less than 100 cm. The focus of this research is to facilitate passengers in efficiently carrying large amounts of baggage, contribute to the

http://dx.doi.org/10.36596/jcse.v5i1.727

development of baggage transportation at the airport, and enhance passenger experience while reducing fatigue for manual trolley users..

A study by Ananda et al., [4] discusses the development of an automated image processing and microcontroller-based robot trolley to enhance customer convenience in supermarkets. By integrating automation technology into conventional shopping carts, the research creates a trolley that can automatically follow user movements. The trolley can move automatically by processing the color information of designated objects using additional devices such as a camera, microcontroller, DC motor, servo motor, motor driver, and battery. The final results of the study show a success rate of 80%, with a maximum distance between objects and the trolley of 5.65 meters. The implementation of automated trolleys is expected to enhance the shopping experience for customers by providing comfort and convenience during the shopping journey.

The study by Ambikapathy et al., [5] discusses the development of an object-following robot with multi-sector applications, including hospitals, hotels, tourist attractions, defense, and industrial automation. This robot is designed to follow and assist humans, addressing the lack of medical care in countries like India and providing support in the defense sector. The success of the robot is evident in its ability to assist customers during busy hours in hotels, act as a guide for lost tourists, and offer effective solutions for transporting goods in various contexts. This innovative solution aims to integrate automation into daily activities, enhance user comfort while shopping in modern markets, especially in retail environments, and align with consumer needs.

In previous research by Poornima et al., [6] focuses on the development of a robot that can track humans and is designed as a support for daily life. The main goal is to create a robot capable of tracking and moving according to specific targets, utilizing barcode technology. Videos captured through a camera are converted to grayscale and displayed frame by frame using OpenCV. Once the barcode is detected, the robot is commanded to move according to the calculated center of mass position. Ultrasonic sensors are used to avoid collisions between the robot and obstacles. This robot can be used as an assisting system for disabled individuals in carrying luggage or applied at airports or train stations as a luggage carrier.

The research by Amangesti et al., [7] introduces an automatic shopping cart to facilitate shopping without the need to push the cart. The design involves a Raspberry Pi module as the controller, DC motors, motor drivers, and a servo motor as actuators, along with a camera as a sensor to detect surrounding objects. Image processing is carried out by detecting HSV color using OpenCV on the objects. The cart can move forward when the object is at a distance of 22 cm ≤ object ≤ 30 cm, move backward at a distance < 18 cm, and stop at a distance of 18 cm to 21 cm or when the object is > 30 cm away. Additionally, the cart can move to the right when the object is on the right side and to the left when the object is on the left side.

Overall, the previous research illustrates efforts in developing robotic technology to enhance customer service and comfort in various contexts. The study by Ananda and colleagues focuses on automatic shopping carts in supermarkets, while Ambikapathy et al. explore multi-sector applications for object-following robots. Poornima et al.'s research centers on utilizing barcode technology in developing human-following robots. On the other hand, Amangesti et al. design an automatic shopping cart with a Raspberry Pi module to facilitate shopping without the need to push the cart. The success of these studies creates innovative solutions expected to improve the shopping experience for customers, provide convenience, and have a positive impact on various aspects of daily life.

In contrast to previous studies, this research focuses on implementing image processing using a camera to detect the presence of humans, based on the recognition of the colors of clothing worn by humans. This robot is not only capable of automatically following humans but also integrated with ultrasonic sensors to monitor the distance between the robot and the person it is following.

http://dx.doi.org/10.36596/jcse.v5i1.727

2. Method

To achieve good results in a research study, it is essential to determine the methods employed in the research. Several methods are utilized during the implementation phase in the design of Image Processing on Human-Follower Robot based on Object Color Using a Camera: (i) Literature Review: This method is used to gather data from various sources such as books, literature, and others. (ii) Design: This method is implemented to design the system to be created. (iii) Experimentation: This method is conducted to test the devices used. (iv) Construction: This method is carried out to create the tool or assemble all the devices. (v) Testing: This method is performed to test the created tool.

The specifications of the devices used for image processing in the human-following trolley robot based on object color using a camera are as follows: (i) 1 camera as an object detector, (ii) 3 HC-SR04 ultrasonic sensors [7], (iii) 1 sensor at the front center to avoid collisions with objects, (iv) 2 sensors on the right and left sides to avoid hitting walls or other obstacles. (v) 1 laptop for digital image processing, (vi) 1 Arduino Mega 2560 board, Zhang et al. [8], as a data processor, (vii) 1 L298n DC motor driver [9], and (viii) 2 DC motors as actuators [10]. The specifications for the design of this trolley robot are illustrated in Figure 1.

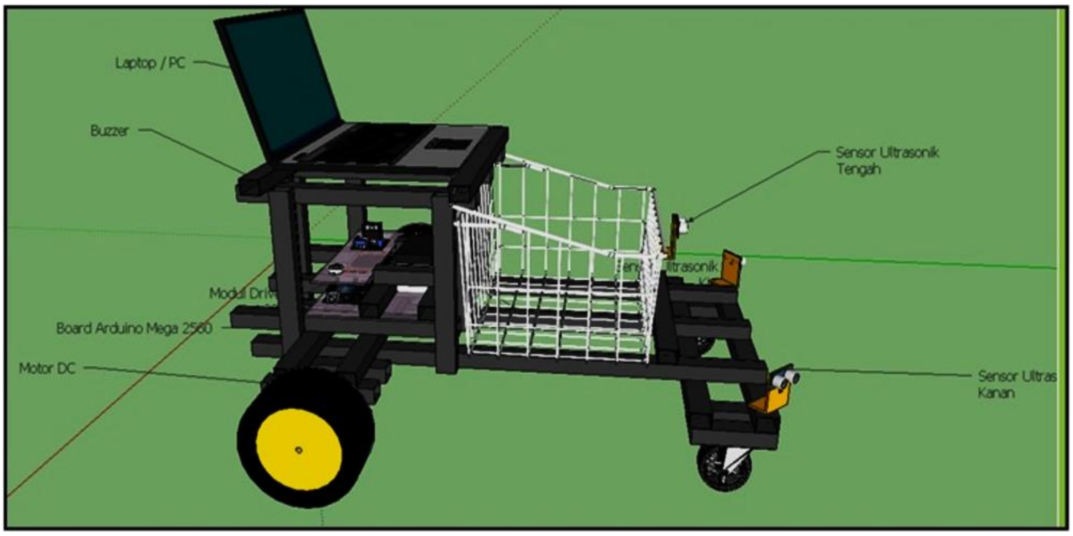


Figure 1. The design of the trolley robot

In Figure 1, there is an explanation regarding the layout of components or circuits for the specifications of the tools used in the design of the trolley robot. Main components, such as ultrasonic sensor HC-SR04, DC motor, laptop, Arduino Mega 2560, and camera, are arranged systematically to operate the system that will be created. This arrangement is necessary for the system to operate according to the established plan.

Design of Robot Trolley Workflow Diagram

The block diagram of Image Processing on Human Follower Robot Based on Object Color Using a Camera can be seen in Figure 2. The explanation of the system workflow diagram is as follows: (i) Objects are recorded using a camera as a sensor, (ii) The image captured by the camera is transmitted to a laptop for image processing to detect relevant objects, (iii) HC-SR04 is used to help determine the distance between the robot and objects in front, as well as to avoid obstacles around the robot, (iv) The results of digital image processing and HC-SR04 data are then processed by Arduino Mega 2560 to control the robot's movement, (v) The processed data from Arduino Mega 2560 is then forwarded to the DC motor driver L298n to drive the DC motor, enabling the robot to move.

http://dx.doi.org/10.36596/jcse.v5i1.727

 e-ISSN 2721-0251

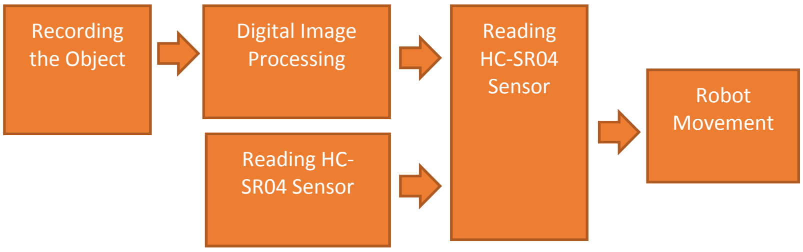


Figure 2. Block Diagram

The working principle of "Image Processing on Human Follower Robot Trolley Based on Object Color Using Camera" is as follows: (i) Initially, the system captures an object using a camera functioning as a sensor. (ii) After capturing the image, color calibration is performed to determine the color to be detected by the robot trolley. (iii) The calibration results provide the object's area and its position on the x and y axes, which serve as references for the trolley's movement. (iv) The decision for right or left movement is based on the object's midpoint on the x-axis, enabling the robot trolley to move horizontally, either to the right or left. (v) Data from image processing on the laptop is then sent to Arduino Mega 2560 to control the movement of the DC motor by sending PWM to the DC motor driver L298n. (vi) To assist the trolley's movement and avoid collisions with objects or obstacles, an ultrasonic sensor HC-SR04 is implemented.

Software Planning

The program is the most crucial component in the design of the robot trolley because all system conditions can be controlled through programming. The programming language used is C [11]. To understand the system in detail, the planning of the system is divided into specific parts. In this planning, creating a system flowchart is essential to support testing and equipment analysis.

Planning the pin configuration on the microcontroller

Before diving into the planning for the entire system or the system in each part, the planning of this system needs to initialize the pin in its addressing. The microcontroller in this system functions as a data processor. Additionally, it also functions to store the programmed data written in the programming language. The microcontroller used is the Arduino Mega 2560. The pin configuration on the Arduino Mega 2560 Microcontroller board is illustrated in Figure 3.

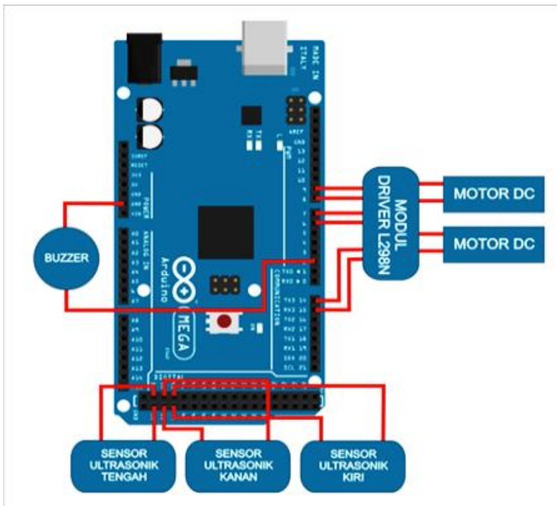


Figure 3. Arduino Pin Configuration

On Figure 3 (Arduino Mega 2560 board pin configuration), the used pins on the robot trolley are as follows: PIN 2 is used for the Buzzer output, PIN 6 is used for input 1 of the L298n Motor Driver, PIN 7 is used for input 2 of the L298n Motor Driver, PIN 8 is used for input 3 of the L298n Motor Driver, PIN 9 is used for input 4 of the L298n Motor Driver, PIN 14 is used for Enable Driver

http://dx.doi.org/10.36596/jcse.v5i1.727

 e-ISSN 2721-0251

Motor L298n (B), PIN 15 is used for Enable Driver Motor L298n (A), PIN 48 is connected to the echo pin of the ultrasonic sensor HC-SR04 (left), PIN 49 is connected to the trigger pin of the ultrasonic sensor HC-SR04 (left), PIN 50 is connected to the echo pin of the ultrasonic sensor HC-SR04 (right), PIN 51 is connected to the trigger pin of the ultrasonic sensor HC-SR04 (right), PIN 52 is connected to the echo pin of the ultrasonic sensor HC-SR04 (center), PIN 53 is connected to the trigger pin of the ultrasonic sensor HC-SR04 (center), and the USB port is connected to the laptop [12].

Initialization of the DC Motor Driver

This design uses the L298n IC as the DC motor driver. There are two DC motors on this trolley robot, located on the right and left sides. The movement of the trolley robot is limited to three directions: forward, turn right, and turn left. The pin configuration for the DC motors can be seen in figure 4.

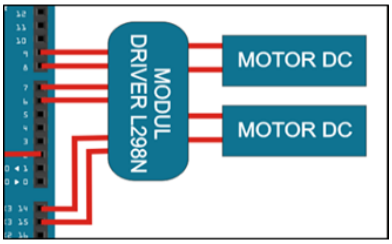


Figure 4. Configuration of DC Motor Pins

The motor driver controls two DC motors with details of 6 input pins and 4 output pins. This design is intended to drive the two rear wheels.

```
// Left Motor Initialization
pinMode(6, OUTPUT);
pinMode(7,OUTPUT);
pinMode(15,OUTPUT);
// Right Motor Initialization
pinMode(8, OUTPUT);
pinMode(9,OUTPUT);
pinMode(14,OUTPUT);
void stopMotor (void){
     analogWrite(14,0);
     analogWrite(15,0);}
     void majuka(int x){
     digitalWrite(8,LOW);
     digitalWrite(9,HIGH); analogWrite(14,x);}
void majuki (int x){
    digitalWrite(6, LOW);
    digitalWrite(7,HIGH);
    analogWrite(15, x); }
```

In the above code, it can be seen that each motor input is connected to enable A, enable B, input 1, input 2, input 3, and input 4. Pins 14 and 15 are Enable pins that provide PWM signals to the DC motor.

Table 1. Rotation state of DC motor

Motor DC	Motion	Condition
Right	CW	PIN 8 (LOW) PIN 9 (HIGH)
Left	CCW	PIN 6 (LOW) PIN 7 (HIGH)

http://dx.doi.org/10.36596/jcse.v5i1.727

Table 1 shows the direction of rotation of the DC motors controlled by the DC motor driver. Each DC motor rotates in only one direction, preventing the robot from moving backward.

Initialization of the HC-SR04 Ultrasonic Sensor

Ultrasonic sensors in the robot trolley system serve as barriers to prevent collisions between the robot trolley and objects or obstacles around the robot. Figure 5 shows the pin configuration of the ultrasonic sensor on the Arduino Mega 2560 microcontroller.

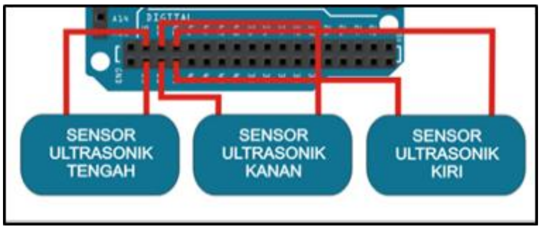


Figure 5. Ultrasonic Sensor Pin Configuration

The three HC-SR04 ultrasonic sensors are connected to digital pins ranging from pin 48 to pin 53. The configuration of the ultrasonic sensors requires initialization on the Arduino Mega 2560 by adding the ping sensor library to the program list in the Arduino IDE. Each ultrasonic sensor has four pins that need to be connected, namely, pins for VCC, ground, echo, and trigger.

Table 2. Configuration of the ultrasonic sensor.

Parts	Pin trigger	Pin echo	Maximum distance
Right	51	50	3 m
Center	53	52	3 m
Left	49	48	3 m

Buzzer Initialization

The buzzer functions as an alarm that provides a signal when the robot stops or when the distance between objects is too far for the robot to detect. The buzzer has only 2 inputs, namely vcc and ground. The pin configuration of the buzzer on the Arduino board can be seen in figure 6.

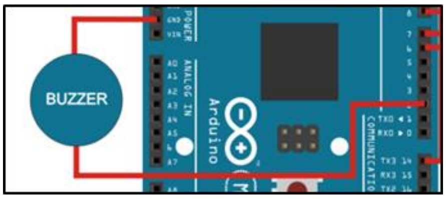


Figure 6. Configure the buzzer pin

Pin 2 is connected to the VCC part of the buzzer, and the Ground pin is connected to the Ground of the buzzer. The configuration of the buzzer requires initialization on the Arduino Mega 2560 by defining the buzzer in the program list in the Arduino IDE. The buzzer is set to turn on and off with a 100ms delay, providing effective warnings for objects.

Laptop Initialization

The laptop is used for digital image processing recorded by the webcam. The software used for image processing is Microsoft Visual Basic 2012, utilizing the OpenCV library version 2.4.11. The configuration for the laptop on the Arduino Mega 2560 board can be seen in figure 7.

http://dx.doi.org/10.36596/jcse.v5i1.727

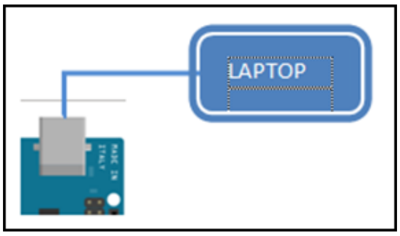


Figure 7. Serial communication configuration

The laptop is connected to the Arduino board through the USB port on Arduino. This allows the laptop to connect via serial communication. To establish the connection, the laptop needs initialization with Arduino. The laptop only sends data to Arduino, and Arduino only reads the data sent by the laptop, making the serial communication unidirectional.

Webcam initialization (kamera)

The camera serves as a sensor to capture images of objects. The camera used is the built-in webcam on the laptop. However, an external camera can also be used by connecting the camera's USB cable to the laptop's USB port. To connect the camera to image processing, initialization is required. If using the laptop's internal camera, use the camera address with the number zero (0). If using an external camera, it is advisable to check first in the laptop's Device Manager[12].

Flowchart Image Processing Workflow

To facilitate program design in the trolley robot, creating a flowchart for image processing as seen in Figure 8 is crucial. This flowchart explains the working system when the trolley robot performs image processing from start to finish, which will later be sent to Arduino. The process begins with calibrating the red color that the robot will follow, obtaining the appropriate HSV values. Next, the HSV image is converted into a binary image to facilitate the separation between red and other colors. After successfully detecting the red color, the object is marked with a box, and the box's area is calculated to determine the center point of the object.

The center point of the red object is measured with the centerline visible in the screen capture. From this distance, the PWM value is calculated to move the DC motor, allowing the robot to follow the object. If the object is to the right of the robot's front, the PWM value for the left-side DC motor will be greater than the PWM value for the right-side DC motor, causing the robot to turn right. Conversely, if the object is to the left of the robot's front, the PWM value for the right-side DC motor will be greater than the PWM value for the left-side DC motor, causing the robot to turn left..

http://dx.doi.org/10.36596/jcse.v5i1.727

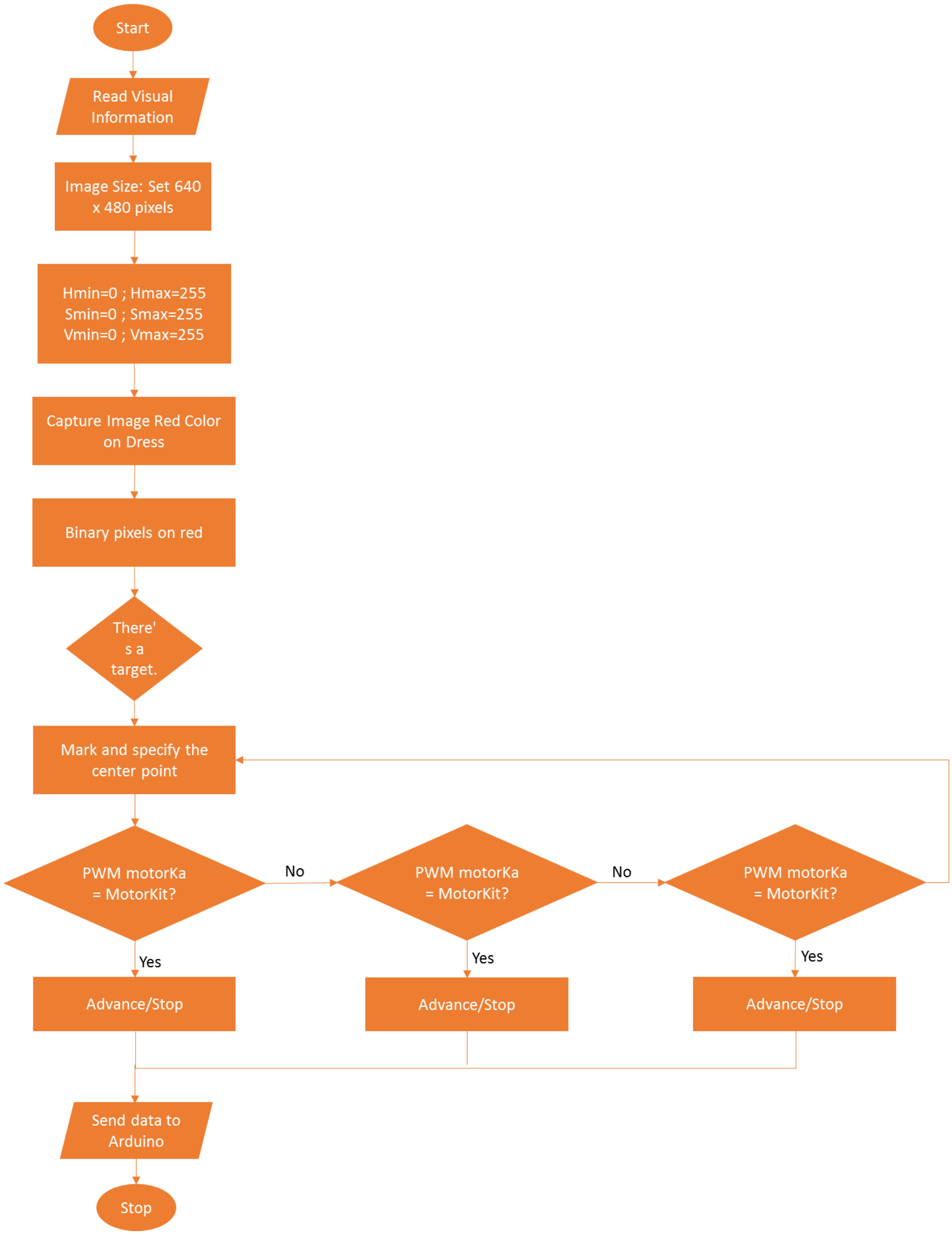


Figure 8. Image Processing Flowchart

Flowchat Working Process Robot Trolley

To facilitate the overall program design of the human-following trolley robot, the creation of a flowchart is essential. In Figure 9, there is a flowchart illustrating the entire workflow of the trolley robot system, combining image processing with ultrasonic sensor readings.

The explanation of the flowchart for the trolley robot's operation is augmented by ultrasonic sensors to avoid collisions with objects or obstacles around the robot that may obstruct its path. When the robot is first powered on, the initial step involves initializing the ultrasonic sensor and establishing serial communication with the laptop. Subsequently, the robot receives image processing data in the form of PWM values used to drive the DC motors.

http://dx.doi.org/10.36596/jcse.v5i1.727

 e-ISSN 2721-0251

Upon receiving this data, Arduino combines it with distance measurements obtained from the ultrasonic sensor. If the central ultrasonic sensor detects a distance between the robot and an object or obstacle in front of it less than 30 cm, the trolley robot will stop, even if image processing provides PWM data. When the left ultrasonic sensor reads a distance of less than 15 cm, the right DC motor stops, and the left DC motor rotates, causing the robot to turn right. Similarly, when the right ultrasonic sensor reads a distance of less than 15 cm, the left DC motor stops, and the right DC motor rotates, causing the robot to turn left. When all three sensors read a distance greater than 30 cm, the robot moves according to the PWM provided by the laptop through image processing..

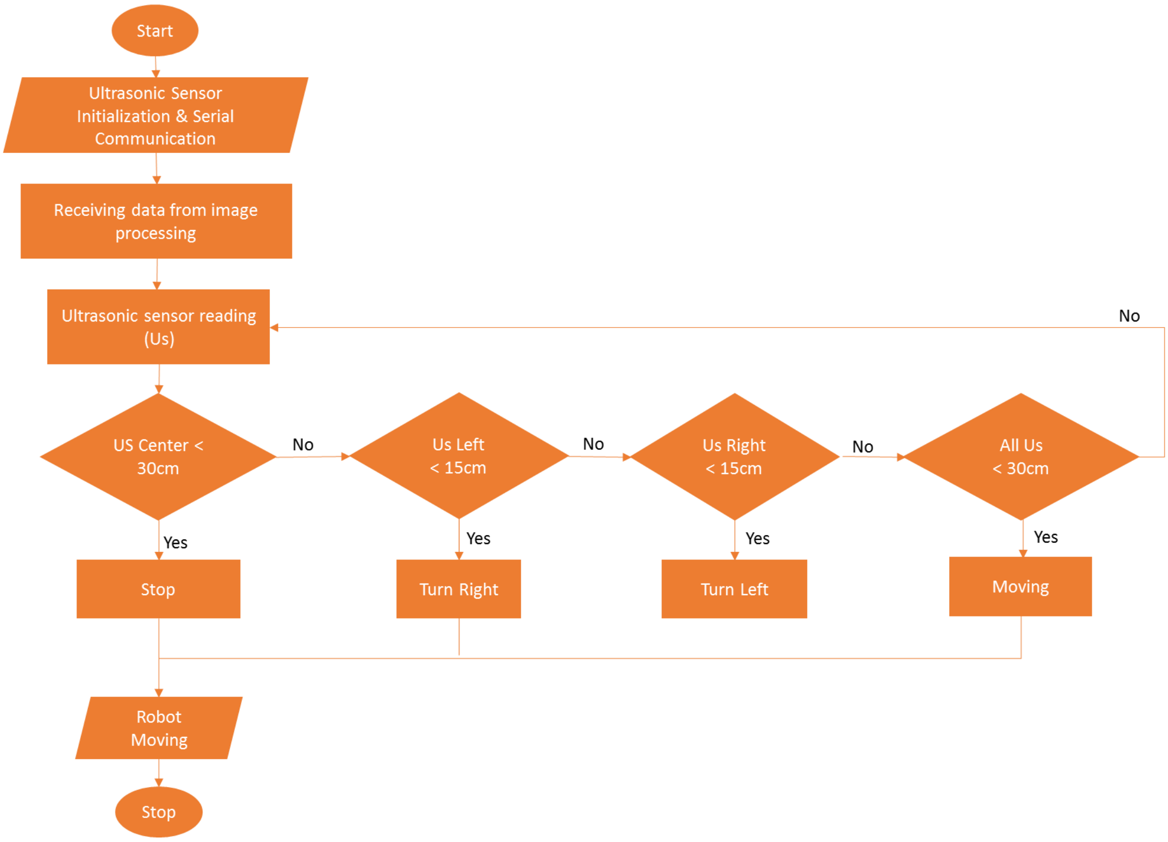


Figure 9. Flowchart of the Trolley Robot System

3. Results and Discussion

After the planning and construction of the device are completed, the next step is to conduct device testing with the following objectives: (i) To assess the results of the device design and construction. (ii) To analyze errors and weaknesses in the device and make comparisons. (iii) To identify shortcomings and weaknesses in the circuit or device, allowing for corrections in case of errors or malfunctions.

Testing

At this stage, tests and measurements are carried out on each block of the network. The results of these tests or measures will be used for comparison between the design and the set of instruments that have been made. Some parts will be tested or measured among others:

Image Processing Test Capture Image

Image processing testing during image capture aims to ensure that the program can capture images effectively. To verify that the image processing program functions properly, the initial step is to capture an image with the camera to check if the program is capable of retrieving images from the connected camera. The program list creation is performed using Microsoft Visual Studio 2012. One of the crucial commands for running the created program list is the Local Windows Debugger.

http://dx.doi.org/10.36596/jcse.v5i1.727

When the program list is about to be executed, the application will build the program to check for any errors. If errors are detected, the program list cannot be executed (debugged).

After creating the program list for capturing images from the camera, the program captures images from the internal camera on the laptop and displays them on the laptop screen with a resolution of 640 x 480 pixels. The testing results can be seen in Figure 10.

Figure 10. Capture Image View

Binary Image Testing and Color Calibration

Biner image testing and color calibration aim to ensure that the number of colors in the image can be converted to just two colors that will be used. To determine if the robot can only follow one color, the first step is to calibrate the color in the image captured by the camera by observing the image that has been converted into a binary image.

When the camera captures an image, the image is still in RGB color. Then, to adjust the RGB color to be close to the original, it is converted to HSV. Next, to obtain a suitable binary image, Thresholding is performed, which separates pixels based on the degree of grayscale. If the grayscale level of a pixel is smaller than the threshold, it will be converted to black, and if the grayscale level is greater than the threshold, it will be converted to white. This is where color calibration is useful, to obtain a binary image that corresponds to the desired color, turning it into white. On figure 11 and 12, it can be observed that they have been converted to binary images but have not been calibrated yet:

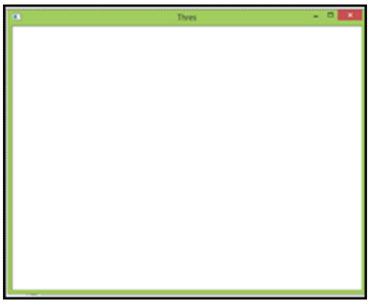


Figure 11. Binary Image View

Figure 12. Color calibration view

The binary image only shows white color over its whole surface when it is not yet calibrated. It is specifically calibrated for the color red because the reference object to be used is red. The outcomes are shown in Table 3 following the creation of a color calibration dialogue box and the development of the program to convert the image in the camera to a binary image.

http://dx.doi.org/10.36596/jcse.v5i1.727

Table 3. Calibration value

Characteristics	Value 0 – 255
Hue	162
Saturation	255
Value	119

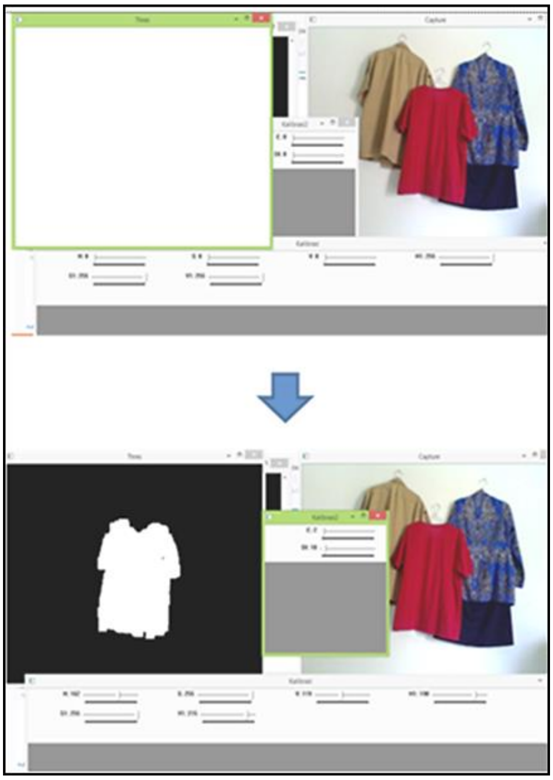


Figure 13. View color calibration results

In image 13, it can be observed that when the image from the camera is converted into a binary image, only the part of the red-colored shirt turns white. The other parts, aside from the red color, turn black.

Center Point Markers Testing Object

The object center point marker test aims to determine that the image processing system is already able to determine the size of the object and the center point of an object. This center point is used to determine how much PWM should be given so that the robot can move following the object.

To ensure that the object center-point marking program functions properly, the first step is to locate the object with the largest size. Once the camera identifies the largest object, it proceeds to mark that object. This marking is essential for determining the center point of the detected object. The program also includes x and y-axis lines in the camera image, facilitating the determination of the image center point from the camera.

The test results can be seen in figure 14, where the green box serves as the indication of the detected object. After the green box appears, the area and center point of the object can be determined. The area of the object can be calculated using the formula for the area of a rectangle.

$$L = Width \; x \; Height \quad (1)$$

Then, to find the center point of the object using the formula below:

$$Center\ Point: \left(\frac{Height}{2} + \frac{Width}{2} \right) \quad (2)$$

http://dx.doi.org/10.36596/jcse.v5i1.727

The red lines represent the x-axis and y-axis of the image captured by the camera. The intersection of the x-axis and y-axis produces the center point of the camera image.

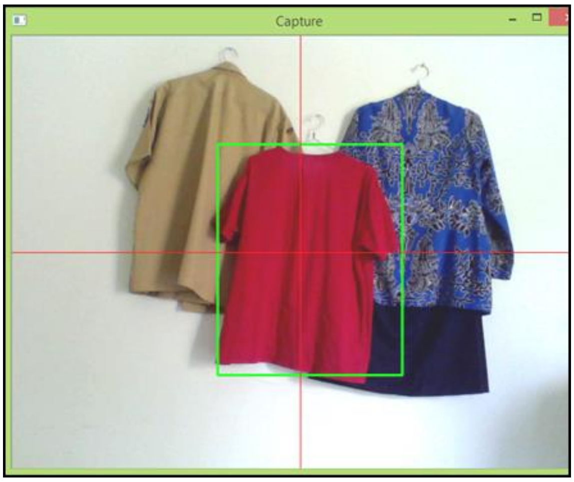


Figure 14. Object marker result

Image Processing Test

Image processing testing is conducted to ensure that the overall image processing system functions properly, and it involves the following stages: (i) Changing the position of objects with different colors: There are three shirts with different colors, one of which is the object to be detected, namely the red-colored shirt. Then, the positions of these three shirts are exchanged three times to determine if the program correctly follows the red color. (ii) Interference by other objects with similar colors: Several shirts and objects with colors almost identical to the object's color are placed close to each other to test if the program can correctly select the color of the object. (iii) Influence of object position on PWM: The object's position will be changed from close to the camera to far from the camera, and then it will be placed on the right and left sides of the camera. The corresponding PWM values will be observed based on these positions. Here are the results of the image processing test:

1. The results of the image processing test.

Figure 15. Object in the middle

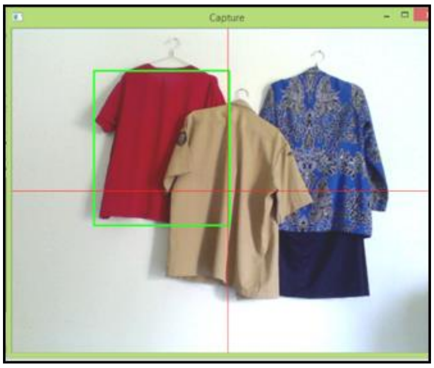


Figure 16. Object in the left

http://dx.doi.org/10.36596/jcse.v5i1.727

 e-ISSN 2721-0251

Figure 17. Object in the right

Figures 15, 16, and 17 demonstrate that despite shuffling the object's position multiple times, the robot consistently marks the red-colored shirt (object).

2. Testing with interference by other objects with the same color

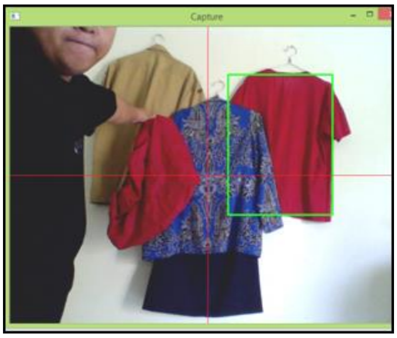


Figure 18. Clothing with other objects

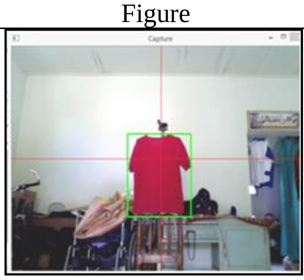
Figure 19. Clothing with clothing of different sizes

Figure 20. Plain shirt with patterned shirt

From Figures 18, 19, and 20 above, it can be seen that when the object is placed close to another object with nearly the same color, there is no error in marking the specified object.

3. Testing with effect of object position on PWM

Table 4. Object position on the x-axis

Object Position	Figure	Right Motor	Left Motor
Center		140	140

http://dx.doi.org/10.36596/jcse.v5i1.727

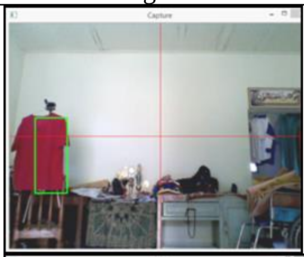
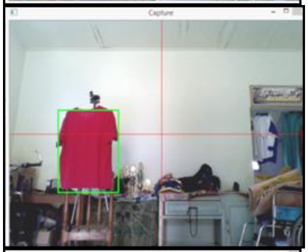
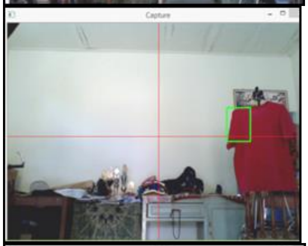
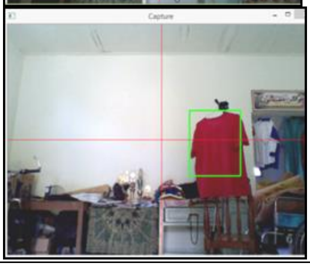
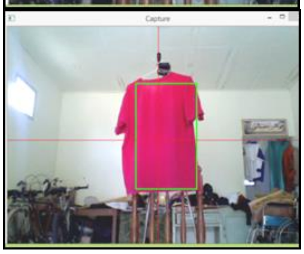
Object Position	Figure	Right Motor	Left Motor
Left		163	117
Left		155	125
Right		123	157
Right		129	151

Table 4 shows the movement of the object on the x-axis. When the object is aligned with the camera, the PWM of the right DC motor is equal to the PWM of the left DC motor. When the object is to the right of the robot, the PWM of the right DC motor will be smaller than the PWM of the left DC motor, causing the robot to turn to the right. Conversely, when the object is to the left of the robot, the PWM of the right DC motor will be larger than the PWM of the left DC motor, causing the robot to turn to the left.

Table 5. Object Position on the z-axis

Object Position	Figure	Right Motor	Left Motor
1 meter		10	11
1,5 meter		140	140

http://dx.doi.org/10.36596/jcse.v5i1.727

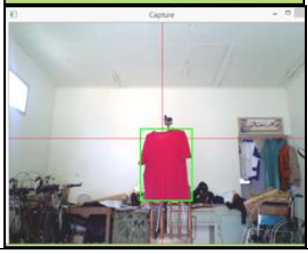
Object Position	Figure	Right Motor	Left Motor
2 meter		142	143
3 meter		151	150

Table 5 shows the forward movement of the object. From the four experiments above, we can identify two situations: when the object is close to the robot and when the object is far from the robot. When the object is at a close distance to the robot (0 – 1 meter), the generated PWM is very small, around 10 or nearly the same as the robot not moving. When the object is at a far (as long as it is still detected) or moderate distance (1.5 meters or more), the generated PWM is large, around 140, allowing the robot to move.

Ultrasonic Sensor Testing

Ultrasonic sensor testing is conducted to ensure the proper functioning of the ultrasonic sensor program. This is achieved by directly displaying the results of the ultrasonic sensor measurements through serial communication in the Arduino IDE. The test results from the ultrasonic sensor readings by Azis et.al [13], can be seen in Table 6 the randomly measured results on the ultrasonic sensor show that all three sensors are functioning properly.

Table 6. Ultrasonic Sensor Testing

Sensor Right	**Sensor Center**	**Sensor Left**
Right=51	Center=270	Left=106
Right=47	Center=106	Left=105
Right=43	Center=105	Left=0
Right=43	Center=106	Left=0
Right=45	Center=105	Left=0
Right=45	Center=270	Left=106
Right=45	Center=103	Left=106
Right=43	Center=106	Left=107
Right=46	Center=108	Left=106
Right=46	Center=106	Left=106
Right=46	Center=109	Left=106
Right=46	Center=104	Left=105
Right=47	Center=106	Left=106
Right=48	Center=105	Left=106
Right=48	Center=105	Left=106

In Table 6, the ultrasonic sensor at the center exhibits significant variation in values, along with some noticeable outliers, potentially impacting the overall performance of the robot. To address this issue, a comprehensive approach is recommended, including operational environmental analysis to identify influencing factors, measurement of environmental variabilities like light, temperature, and humidity, separate testing of the ultrasonic sensor for damage or interference

http://dx.doi.org/10.36596/jcse.v5i1.727

detection, real-time system monitoring to record specific conditions during normal operations, analysis of log data to identify patterns causing sensor variations, correlation with external resources such as sunlight, development of correction algorithms for identifiable factors, and recalibration of the ultrasonic sensor to ensure reading accuracy [14],[15],[16].

Robot trolley testing

The testing of this trolley robot aims to ensure that the entire system functions properly. The testing involves the evaluation of both the software and hardware components of the robot. To evaluate whether the trolley robot operates according to its design, several tests were conducted: (i) **Robot Response Test:** In this test, the PID values of the robot were systematically adjusted to modify the robot's response, aiming to find the optimal PID configuration that yields the best response. (ii) Obstruction Test with a Person: While the robot was following a moving object, a test was performed by introducing an obstruction— a person walking in front of the trolley robot, wearing a shirt with a color different from the object being followed. This test aimed to determine whether the robot would be affected by the interference of a different colored object passing in front of it. (iii) Two Similar Colors Test: This test involved changing the color of the wall to red, but the shade of red did not match the color of the object being tracked. The object was then relocated to assess whether the trolley robot could still successfully follow the object despite the change in the surrounding color. The results of several trials of the robot trolley, can be seen in figure 25, 26, 27 and 28.

1. Testing the robot's response

Figure 25. The trolley robot responds slowly

Figure 26. Robot trolley is not following the object

Figure 27. Robot trolley is following an object

Figure 28. The trolley robot responds slowly

From the test results, it is evident in Figures 25 and 26 that the trolley robot exhibits slow movements and frequently experiences errors or struggles to follow the object effectively when the PID is not yet configured. However, in Figures 26 and 27, after PID tuning, it is observed that the robot's movements have improved with a rapid response, allowing the trolley robot to successfully follow the object.

2. Testing with interference from a person walking in front of the robot trolley

http://dx.doi.org/10.36596/jcse.v5i1.727

Figure 29. People walk closer to robots

Figure 30. People are between objects and robots.

Figure 31. People stay away from robots

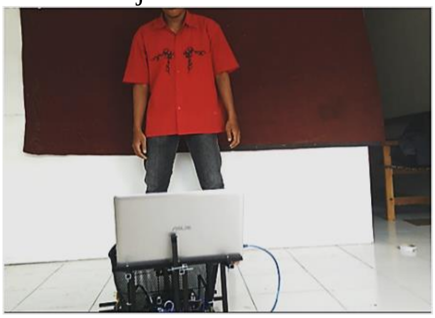


Figure 32. Object is in the center

Figure 33. The object is on the left of the robot trolley

Figure 34. The object is on the Right of the robot trolley

The results of the test in the figure above show that the robot is not affected or disturbed by the presence of other people walking between the object and the trolley robot. Testing with two nearly identical colors, as shown in Figure 32 object in the center, Figure 33 Object to the left of the robot trolley, and Figure 34 Object to the right of the robot trolley, indicates that the robot can still track the object even though the wall has a red color. It is important to note that the red color on the wall is not identical to the red color on the shirt used as the object. Therefore, the results of this test prove that the robot will not be disturbed by a red-colored banner or wall that is different from the red color of the object.

Analysis Results

From the results of testing all the components listed in Table 4, it is evident that the trolley robot can operate according to the intended design, even though changes in light intensity that may occur at any time can affect the camera's ability to recognize the detected colors. Additionally, the robot's speed is only adjusted to the walking speed of the person being followed. When the robot moves too fast, the ultrasonic sensor will stop it if the distance between the robot and the person being followed is less than 30 cm. This is intended to prevent collisions between the robot and the person being followed.

http://dx.doi.org/10.36596/jcse.v5i1.727

Table 4. Testing for trolley robot system

Test Number	Testing	Result
1	Capture Image testing with camera	The camera can take pictures perfectly with a 640 x 480 pixel size.
2	Color and binary image calibration testing	Calibration manages to distinguish between objects and objects around them by giving only the white color to the object and the black color to other objects in the binary image.
3	Testing the center point marker of the object	Appears a green box on the side of the object as a marker
4	Image processing test:	
	a. Change the location of the object by another object	a. The program continues to follow the object used as a benchmark.
	b. Interference by other objects of the same color	b. The program can still follow the object, because the object has a larger size
	c. Effect of object position on PWM	c. When the object is in the center of the right PWM = left PWM When the objects are in the right pWM < left pWm When the Objects are on the left right pwm > left Pwm
5	Ultrasonic sensor testing	Serial communication successfully displays the value of the data detected by the sensor
6	Robot trolley testing:	
	a. Robot response testing	a. The robot responds very well when given a PID value
	b. Testing with interference from a person walking in front of the robot	b. The robot is not disturbed at all, returning to follow the person it should follow
	c. Pengujian dengan dua warna yang hampir sama	c. Even with a red-colored wall, the robot still follows a person in a red shirt, because the calibration value between the wall and the person's shirt being followed is different.

4. Conclusion

Based on the design and analysis results conducted on the trolley robot, it can be concluded that this robot is capable of recognizing objects wearing red-colored clothing after color calibration when the robot is turned on. The calibration process not only aids in recognizing the color of the object but also plays a role in distinguishing objects with different colors. After recognizing the object, the robot can follow it by utilizing image processing to differentiate between the target object and other colors, producing PWM values to drive the DC motors. The robot is also equipped with ultrasonic sensors that allow it to monitor the distance between itself and the followed object. The minimum distance that can be maintained is 30 cm, while the robot can still follow an object with a maximum distance of over 3 meters, as long as the object is still detected. To prevent collisions, ultrasonic sensors are also placed on the right and left sides of the robot.

However, there are some limitations that can be improved in future research. Image processing in object recognition can be optimized by adding recognition of upper body parts' shapes (human shoulders), resulting in more optimal robot responses. Additionally, light intensity can affect image processing results, so it needs to be considered for further research. The HC-SR04 ultrasonic sensor can also be enhanced to be less affected by light. The use of a laptop with better specifications is also recommended to prevent potential hangs or freezes..

http://dx.doi.org/10.36596/jcse.v5i1.727

References

[1] Shahria MT, Sunny MSH, Zarif MII, Ghommam J, Ahamed SI, Rahman MH. A Comprehensive Review of Vision-Based Robotic Applications: Current State, Components, Approaches, Barriers, and Potential Solutions. Robotics. 2022; 11(6):139. https://doi.org/10.3390/robotics11060139.

[2] M. Lovita and R. Mukhaiyar, “Rancang Bangun Troli Otomatis Berbasis Computer Vision”, JTEIN, vol. 3, no. 2, pp. 399-406, Aug. 2022. https://doi.org/10.24036/jtein.v3i2.265

[3] M. . Akbar Suprobo, S. . Hariyadi, and W. . Suryono, “Smart Electric Trolley Menggunakan Sistem Pengereman Otomatis Berbasis Wemos D1R1”, SNITP, vol. 5, no. 1, Jul. 2022.

[4] G. Ananda, C. Wahyuni, E. Sania, and S. Sukardi, “Design of An Automatic Follower Shopping Trolley Based on Image Processing”, Teknomekanik, vol. 2, no. 2, pp. 48-54, Dec. 2019. https://doi.org/10.24036/tm.v2i1.3472

[5] Ambikapathy, A., et al. "Analysis of Object Following Robot Module Using Android, Arduino and Open CV, Raspberry Pi with OpenCV and Color Based Vision Recognition." Advances in Power Systems and Energy Management: Select Proceedings of ETAEERE 2020. Springer Singapore, 2021. https://doi.org/10.1007/978-981-15-7504-4_35

[6] G. R. Poornima, J. L. Avinash, S. Palle, S. Santosh Kumar, K. N. Sunil Kumar and P. Rajendra Prasad, "Image Processing Based Human Pursuing Robot," 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2020, pp. 408-412, doi: 10.1109/RTEICT49044.2020.9315662

[7] Hugeng Hugeng, Ivan Darrius, Meirista Wulandari; Design of automatic trolley based on microcontroller. AIP Conf. Proc. 7 December 2023; 2680 (1): 020042. https://doi.org/10.1063/5.0126127

[8] X. Zhang, Q. Xu, B. Li and L. Liu, "Design and Implementation of Supermarket Shopping Robot," 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), Manchester, United Kingdom, 2021, pp. 233-236, doi: 10.1109/AIAM54119.2021.00055.

[9] N. N. Azimi, A. S. Arman Sabri, N. H. Shamsul, And N. A. Ahmad, “Auto Following Trolley”, MARI, vol. 4, no. 2, pp. 303–310, Jan. 2023.

[10] A. Imron and I. Anshory, “Design And Build Electric Trolly To Accelerate The Distribution Of Goods In The Warehouse”, PELS, vol. 3, Dec. 2022. https://doi.org/10.21070/pels.v3i0.1366

[11] M. Sanap, P. Chimurkar and N. Bhagat, "SMART-Smart Mobile Autonomous Robotic Trolley," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 430-437, doi: 10.1109/ICICCS48265.2020.9120972

[12] Azis, Abdul, Emidiana Emidiana, and Fathir Azukruf. "Perancangan Prototipe Robot Pembawa Barang Otomatis Berbasis Arduino Uno DIP dengan Sensor Ultrasonic." Elektrika 15.1 (2023): 29-38. http://dx.doi.org/10.26623/elektrika.v15i1.5924

[13] Xiao, Anxing, et al. "Robotic autonomous trolley collection with progressive perception and nonlinear model predictive control." 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022. https://doi.org/10.1016/j.procir.2020.03.097

[14] Keshmiry A, Hassani S, Mousavi M, Dackermann U. Effects of Environmental and Operational Conditions on Structural Health Monitoring and Non-Destructive Testing: A Systematic Review. Buildings. 2023; 13(4):918. https://doi.org/10.3390/buildings13040918

[15] Hassani S, Dackermann U. A Systematic Review of Advanced Sensor Technologies for Non-Destructive Testing and Structural Health Monitoring. Sensors. 2023; 23(4):2204. https://doi.org/10.3390/s23042204

[16] P. Rajak, A. Ganguly, S. Adhikary, and S. Bhattacharya, “Internet of Things and smart sensors in agriculture: Scopes and challenges,” Journal of Agriculture and Food Research, vol. 14. Elsevier BV, p. 100776, Dec. 2023. doi: 10.1016/j.jafr.2023.100776

http://dx.doi.org/10.36596/jcse.v5i1.727