# New York University

Tandon School of Engineering
Department of Aerospace and Mechanical Engineering

ME-GY 6933

Advanced Mechatronics

Final Project

# Collect and Deliver Robot

**Submitted to**

**Prof. Vikram Kapila**

**Sonia Mary C**

**Skender Alickolli**

**Abhidipta Mallik**

## Table of Contents

# 1. Introduction

There are various ways to collect objects and some of them can be very expensive based on the accuracy one is trying to achieve. Our Robot will be using Color detection to locate the object and the base and will be using a differential drive car to do all the motion.

This Robot can be used in various applications like tennis ball collecting, cricket ball collecting etc. Agriculture would also be another good application, where the robot can be used to collect the apples or oranges on the field and deliver them to warehouse.

The Robot detected a colored image, a green ball in this case, it goes to the object, collects it and then delivers it to a base station. All the image processing is done live, meaning that the robot can do dynamic tracking, so the object can be moving and the robot will still go after it. The main components of this project are the Raspberry pi3 and Arduino Uno. All the image processing is done on the Raspberry pi3 and the robot actuation is done using the Arduino Uno. Pi Cam was used as the "eyes" of the Robot and its information was processed on OpenCV using Python 3.5.

# 2. Design and Implementation

## 2.1 Differential drive robot:

We have built a differential drive robot using parallax servo motors. The Arduino UNO is used for controlling the robot. At the front of the robot, two horn like structure are mounted. This helps to collect and drag the objects on the ground.

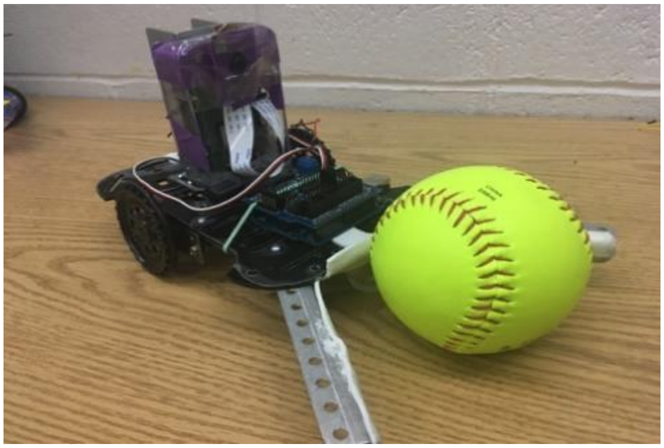


Figure 1: Differential drive robot with ball

## 2.2 Image processing using OpenCV

We used the OpenCV to process the image frames captured by the pi camera. Each frame is then processing at real-time to extract the colored object. The different steps involved in image processing. The first step is to define the lower and upper boundaries of the green color in the HSV color space. These color boundaries will allow to detect the green ball in the frame. The frame is then blurred to reduce high frequency noise to focus on the structural objects inside the frame. The frame is then convert to the HSV color space. The green ball in the frame is located by using *inRange* function in OpenCV. The we performed a series of dilation and erode operation to remove the small blobs in the mask. Next thing is to find the biggest contour in the image and the centroid of the object is being calculated. We are basically using this centroid information to locate the ball position in the image.

## 2.3 Robot Control based on ball position

The centroid of the ball is calculated and this information is used for controlling the movement of the robot. Robot decides to move to left if the ball position is on the left side of the frame. It moves to the right side if the ball is on the right side of the frame. We have defined the threshold values for left, right and forward based on the centroid value of the ball position.

### Distance measurement:

The robot automatically detects the ball distance based on its y value of the centroid. If the ball is far from the object, the y value of the ball centroid is small and this increases as the robot approaches the ball. We have calibrated this value by doing many tests.

### Base station detection:

Once the robot grabbed the ball, it searches for its base station. We used blue color for representing the base station. The robot will look for base station only after it grabbed its ball. We used the same set of image processing techniques for detecting the base station. The distance and direction to the base station is calculated using its centroid position on the screen. We placed the base station above the ground level and therefore the centroid y value reduces when the robot approaches the station and the value would more when the station is away from the robot.

### 2.4 Arduino- Raspberry pi communication

In this project, we used serial communication between Raspberry pi and Arduino to send data with a baud rate of 9600.

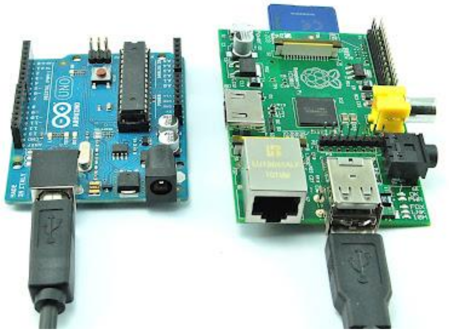
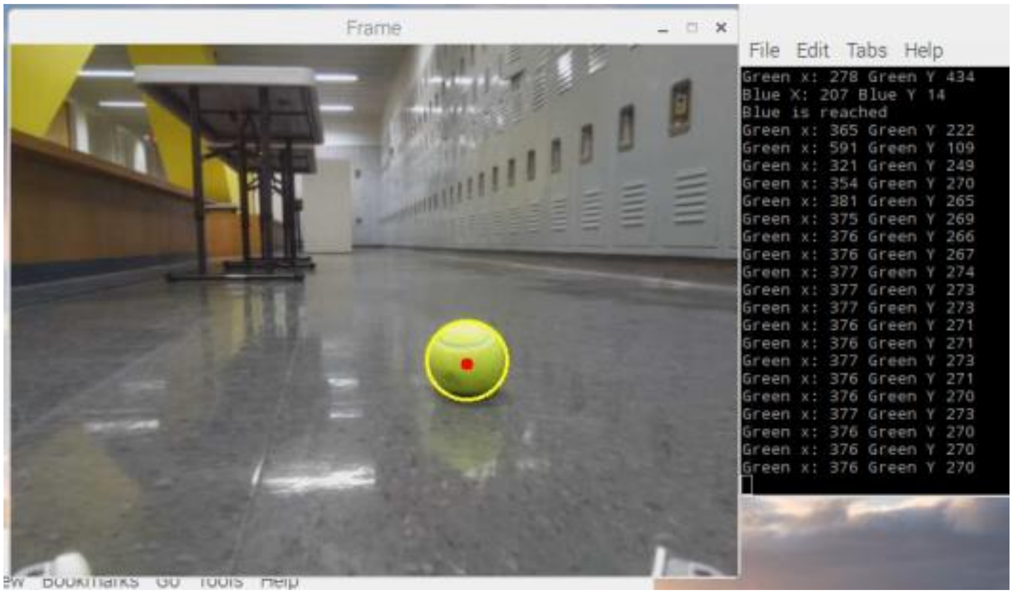


Figure 2: Raspberry pi Arduino connection

## 3. Test and Results

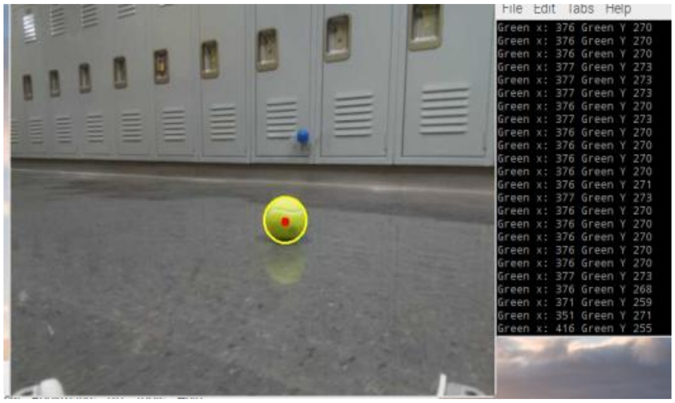
The robot spins and search until it detects a green ball. Figure 3. shows the robots detects the green ball.  The X co- ordinate of the center shows value 376. This indicates the ball is in the right side of the robot. If the X co-ordinate of the centroid is within 230-300, the robot goes forward. If X co-ordinate is less than 230, the robot goes to the left and if it is more than 300 it goes to the right. In this case since the X co-ordinate is 376 the robot moves to right. In this way, the robot tries to go close to the ball.



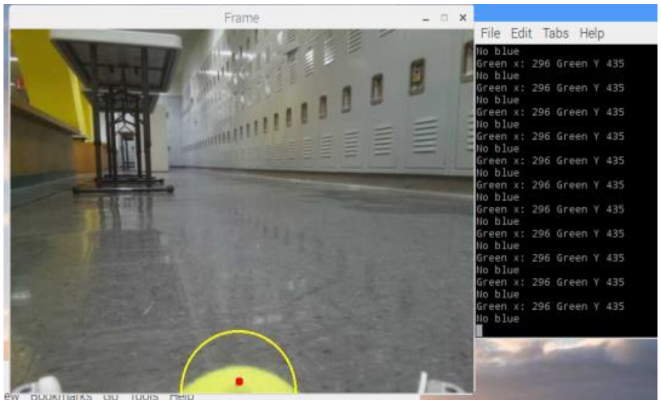
```
Green x: 278 Green Y 434
Blue X: 207 Blue Y 14
Blue is reached
Green x: 365 Green Y 222
Green x: 591 Green Y 109
Green x: 321 Green Y 249
Green x: 354 Green Y 270
Green x: 381 Green Y 265
Green x: 375 Green Y 269
Green x: 376 Green Y 266
Green x: 376 Green Y 267
Green x: 377 Green Y 274
Green x: 377 Green Y 273
Green x: 377 Green Y 273
Green x: 376 Green Y 271
Green x: 376 Green Y 271
Green x: 377 Green Y 273
Green x: 376 Green Y 271
Green x: 376 Green Y 270
Green x: 377 Green Y 273
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
```

Figure 3: Robot detects the ball

In figure 4, the object and base station are in same frame. But here the blue base station is not detected. We want the robot to reach the ball first and after that the robot should look for the base station.  The Y co-ordinate value of the green ball is 270. When the robot encounters the ball, the Y coordinate value becomes more than 400. For this reason, we set the threshold at 400. In this case since the X co-ordinate is 376 the robot moves to right.



```
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 377 Green Y 273
Green x: 377 Green Y 273
Green x: 377 Green Y 273
Green x: 376 Green Y 270
Green x: 377 Green Y 273
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 271
Green x: 377 Green Y 273
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 376 Green Y 270
Green x: 377 Green Y 273
Green x: 376 Green Y 268
Green x: 371 Green Y 259
Green x: 351 Green Y 271
Green x: 416 Green Y 255
```

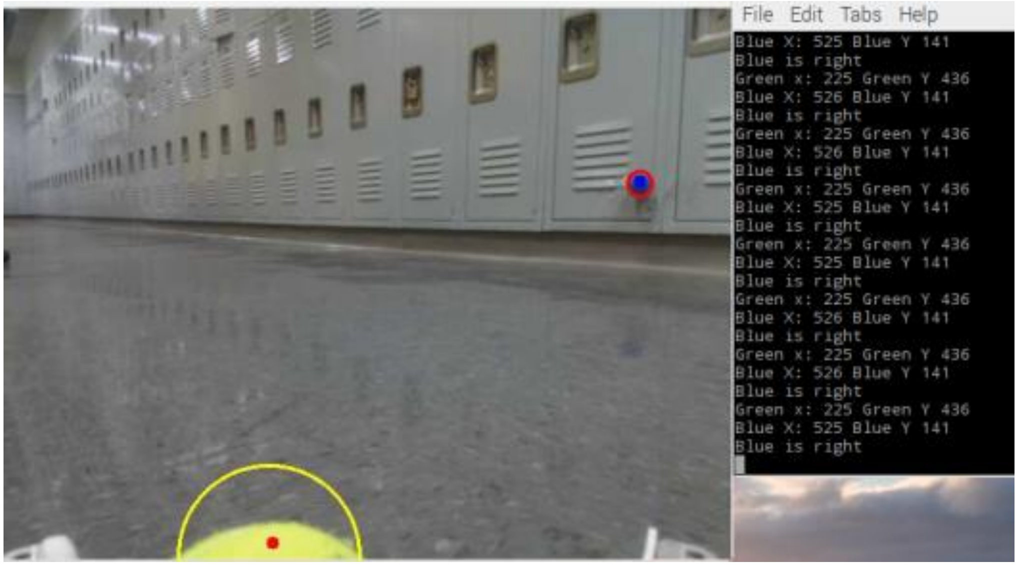


```
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
Green x: 296 Green Y 435
No blue
```

Figure 4: Object and base station in the frame

Figure 5:  Reached the object and searching station

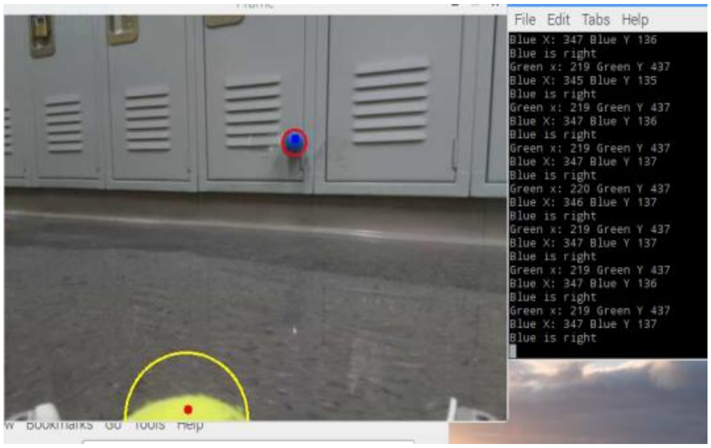
When the robot reaches the green ball, the Y coordinate value of green ball is 435. So, the robot starts looking for the blue ball. But in this position the blue ball is out of sight. So, robot starts to look for the ball with the green ball grabbed in between the robot horns.  The robot spins till the base station is detected. The base station X coordinate is 525. So, the robot moves right.



```
Blue X: 525 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 526 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 526 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 525 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 525 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 526 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 526 Blue Y 141
Blue is right
Green x: 225 Green Y 436
Blue X: 525 Blue Y 141
Blue is right
```

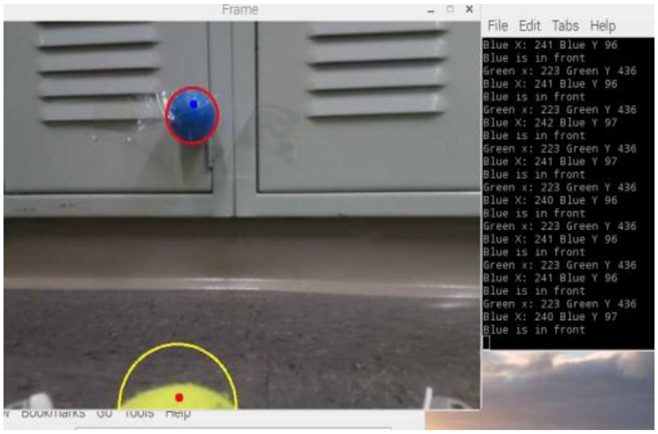
Figure 6: Detects base station

The blue X coordinate is 347. So, the robot moves right. The blue Y co-ordinate is 137. The robot is supposed to move till the blue Y co-ordinate is less than 50. That is because we calibrated and found that the robot reach the base station when the blue Y co-ordinate is less than 50.



```
Blue X: 347 Blue Y 136
Blue is right
Green x: 219 Green Y 437
Blue X: 345 Blue Y 135
Blue is right
Green x: 219 Green Y 437
Blue X: 347 Blue Y 136
Blue is right
Green x: 219 Green Y 437
Blue X: 347 Blue Y 137
Blue is right
Green x: 220 Green Y 437
Blue X: 346 Blue Y 137
Blue is right
Green x: 219 Green Y 437
Blue X: 347 Blue Y 137
Blue is right
Green x: 219 Green Y 437
Blue X: 347 Blue Y 136
Blue is right
Green x: 219 Green Y 437
Blue X: 347 Blue Y 137
Blue is right
```

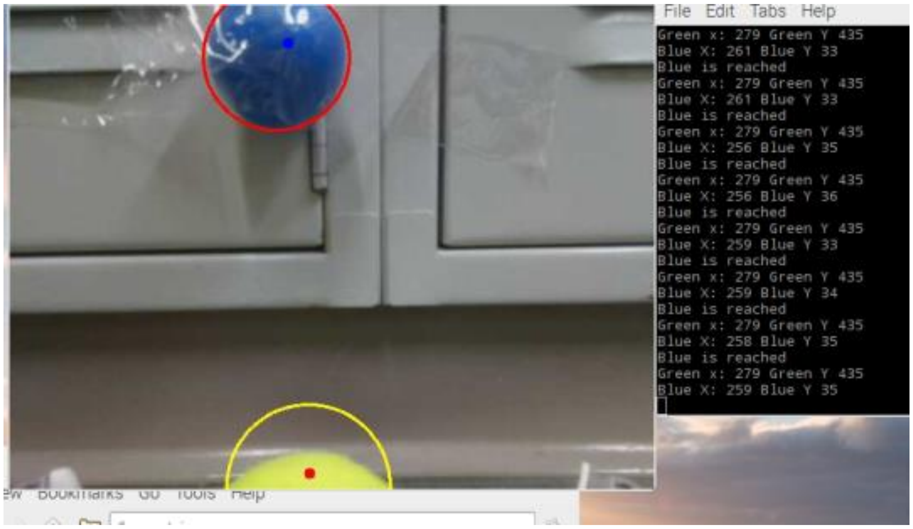
Figure 7: Orienting towards the base station



```
Blue X: 241 Blue Y 96
Blue is in front
Green x: 223 Green Y 436
Blue X: 241 Blue Y 96
Blue is in front
Green x: 223 Green Y 436
Blue X: 242 Blue Y 97
Blue is in front
Green x: 223 Green Y 436
Blue X: 241 Blue Y 97
Blue is in front
Green x: 223 Green Y 436
Blue X: 240 Blue Y 96
Blue is in front
Green x: 223 Green Y 436
Blue X: 241 Blue Y 96
Blue is in front
Green x: 223 Green Y 436
Blue X: 241 Blue Y 96
Blue is in front
Green x: 223 Green Y 436
Blue X: 240 Blue Y 97
Blue is in front
```

Figure 8: Heading towards the base station

The blue X coordinate is 240. So, the robot moves front. The blue Y co-ordinate is 97. The robot does not stop until the y value is less than 50. The robot reached the base station and its Y coordinate is 35. So, the robot stops. In all these cases the value of green X coordinate and Y coordinate did not change because it moved along with the robot. Thus, the robot collects the green ball and delivers the it to the base station.



```
Green x: 279 Green Y 435
Blue X: 261 Blue Y 33
Blue is reached
Green x: 279 Green Y 435
Blue X: 261 Blue Y 33
Blue is reached
Green x: 279 Green Y 435
Blue X: 256 Blue Y 35
Blue is reached
Green x: 279 Green Y 435
Blue X: 256 Blue Y 36
Blue is reached
Green x: 279 Green Y 435
Blue X: 259 Blue Y 33
Blue is reached
Green x: 279 Green Y 435
Blue X: 259 Blue Y 34
Blue is reached
Green x: 279 Green Y 435
Blue X: 258 Blue Y 35
Blue is reached
Green x: 279 Green Y 435
Blue X: 259 Blue Y 35
```

Figure 9: Reached the base station

## 4. Conclusion

Based on the experiments that we conducted the robot responds well, however, it appears to be a little slow and would sometimes respond after the command is sent from the Raspberry Pi. This happens because while the servos are set to move for 100 ms in one direction (left), the object will now be in front of the camera and the Raspberry Pi will be telling the Arduino to be moving forward instead of continuing left, but the Arduino would not be listening to Pi since its actuating the Robot in the left direction while the Pi is talking. This can be eliminated by using a faster and multicore microcontroller.

The Collect and Deliver Robot successfully completes the required task even though not at a very fast speed. Having bigger wheels and a holding basket would be a future development to make the robot work in the real-life environment.

## 5. References

1. http://www.pyimagesearch.com

2. http://docs.opencv.org/2.4/doc/tutorials/tutorials.html