Palestine Polytechnic University

College of Information Technology and Computer Engineering

Computer Systems Engineering

Mobile Robot Collecting Tennis Balls

Team Members:

Ahmad I. Taradi

Ali H. Tamimi

Supervisor:

Dr. Zein Salah

May 28, 2018

Acknowledgment

We are very grateful to our advisor Dr. ZainSalah for his support, patience and encouragement. Without him this Project would never be finished.

Furthermore, we would like to express our thanks to Prof. Kareem Tahboob and Dr. Hashem Tamimi for many helpful discussions and the joint cooperation that led to some of the presented results.

We also want to express our special thanks to our teachers in thecollege, they give us their time to help, learn and support us.

Finally, we are also thankful to our family, especially our parents to help and support us at every time and with each step in our life.

And we are thankful to our friends and to our university Palestine Polytechnic University.

Abstract

After tennis training or a match is finished, a huge number of balls remain in the playground, and it takes lot of efforts and time to collect them, especially in case of players that have special needs.

In this project, we develop a Mobile Robot with on board intelligence system to collect tennis balls. A global vision system is constructed to localize tennis balls and the collecting robot. This localization system comprises from four cameras that are mounted on the corners above playground. Utilizing computer vision algorithms, the system detects locations of individual balls and computes a suitable path for collecting them. This informationare continuously delivered to the mobile robot.

Table of Content

List of Figures

Chapter 1 : Introduction

1.1 Overview of the project

Detecting and localizing objects in digital image has become one of the most important applications for industrial use to ease user and save time. These techniques have been developing years ago, but improvement is still required in order to achieve the targeted objective more efficiently and accurately [1].

The goal of this project is to develop a vision-based system that detects, localizes and collects the tennis balls from a playground. Cameras around the playground must automatically detect and count the total number of balls from tennis playground.This information is sent to the Robot in order to move towards the balls and collect them to get them out of the playground.

1.2 Motivations

Since the world is in a currently rapid revolutionizing in technology, and robots are increasingly utilized to ease the life of human, we decide to choose our idea to keep up with human needs of a welfare life, as an application, we want to solve the problem of collecting tennis balls after a match or a training session.

1.3Goals and Objectives

The goal of this project is to build a vision system to let a robot collect tennis balls from the playground.

Project objectives:

1- Install Cameras around a tennis playground.
2- Calibrate the installed cameras.
3- Pre-process the images.
4- Detect the balls and find their locations.
5- Detect the Robot and find its locations.
6- Configure Network on the site to communicate with the robot.
7- Find the path for Robot in order to collect tennis balls.

1.4 Importance

This project will save time and effort of players who play tennis, help people who has special needs.This project could be used in various applications such as collecting trash.

1.5 Short Description of project

Our project consists of two parts:

1. Robot to collect tennis balls, this robot will be designed and implemented by our partners group with a mechanical methodology to collect tennis balls, a myRio Microcontroller will be added to the robot to behave in order to collect Tennis balls according to navigation data which will be received from the vision system.
2. A Vision system which consist of 4 calibrated cameras, which connected to a configured network which will allow the access to them from a PC, The PC will pre-process the images in order to detect the robot, find location the robot, detect the balls and find their locations, finally using the network at the site, the navigation data will be sent to the robot from the PC.

1.6 Limitations

We have two major limitations in or project:

The first one is the tradeoff between camera quality and the network bandwidth, the highest quality of cameras lead to high size images, so network technology must be upgraded, but the technology must not exceed the estimated budget of the project.

The Second which is the change of the weather may and daytime, will make various Intensity of illumination in the images, which will affect slightly the detection of the robot and the ball.

1.7 Problem Analysis

The first issue is to detect all tennis balls in playground using the cameras, and then determine the locations of (X, Y) in pixels term.

The second one is to calibrate cameras in the sites balls and distance between tennis balls and Robot.

The vision system would capture and image of the siteof playground to be analyzed using C++ andOpenCV Library and revealed that the tennis balls is present or not, and if the tennis balls is detected, the system sends a message contains the location of the ball to the Robot in order to start collecting the tennis balls. And to provide adequate way for Robot we needed sensors like as ultrasonic sensor to avoid obstacles.

In order to collect tennis balls we need to make Robot with on board intelligence and Global System to detect, recognize and find the real location of balls and Robot.

Chapter 2 :Background

This chapter will mention a theoretical background of the most important issues of the project.

First it will discuss the fundamental of Image, the Color Model and Thersolding, these topics are related to the detection part of the project.

Then the camera calibration, Distortion, estimating the camera pose and finding real world location, these topics explain the mathematical model of camera and shows the equations which is needed to calibrate and obtain real world coordinates.

2.1 The fundamental of Image

An image can be represented in digital form as a 2-Dimnatinal vector $f(x, y)$,

$$f\ x,y\ = \begin{matrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(N-1,0) & \cdots & f(n-1,N-1) \end{matrix}$$

Where

f: Vector function

x: Coordinates in X-Axis

y: Coordinates in Y-Axis

N: Matrix Size as Integer [2]

2.2 Color Model

The image taken from the camera will be displayed as f as shown previously, each item has a 3-byte size in the RGB color model, R is referred to in Red, G is indicated by Green and B referred to Blue, each byte represents a value in the range (0-255) , Since these values combine to show the color of the item.

But the main problem of this color model is that it's affected by light and object may be changed its color in the various Light situations, a solution to that problem is to convert to HSV color model.

The H referred to HUE, S referred to saturation and V referred to Values, its size also 3-Byte for each element.

As we done converting to HSV, we would have H in range of (0 – 360o), S in range (0 - 1) and V in range (0 - 1). [3]

Figure 2.1: Tennis Balls Image [4]

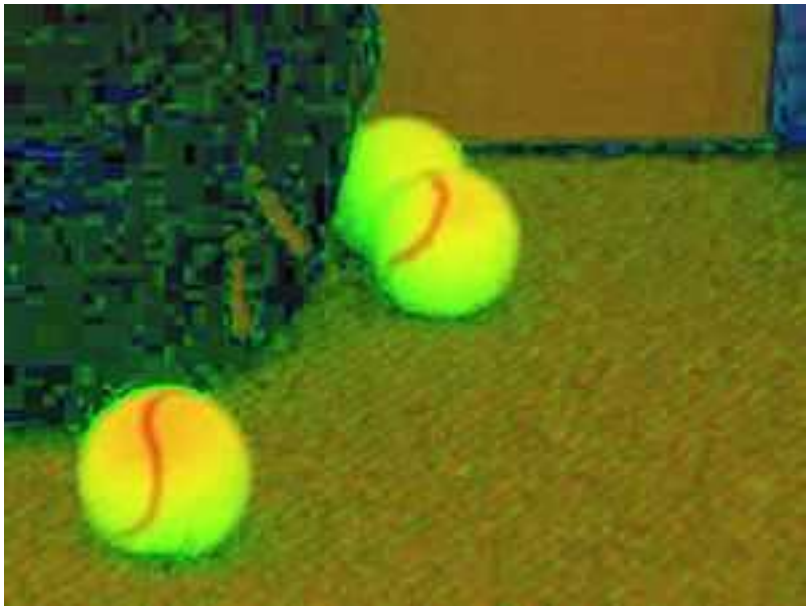

Figure 2.2: Tennis Balls Image in HSV [4]

Figure 3 shows the original image of the tennis balls. Figure 4 shows how the image will appear in the HSV color model we will use to detect the balls.

2.3The Threshold

It’s necessary to make a threshold to the image in order to make unwanted object to disappear.

This enable us to detect only tennis balls, based on the threshold each pixel out of the range to 0, all pixels in the range will be assign value of 1.

The range was identified by trial it were:
(H_min=19,H_max=39,S_min=119,S_max,V_min=4,V_max=206).

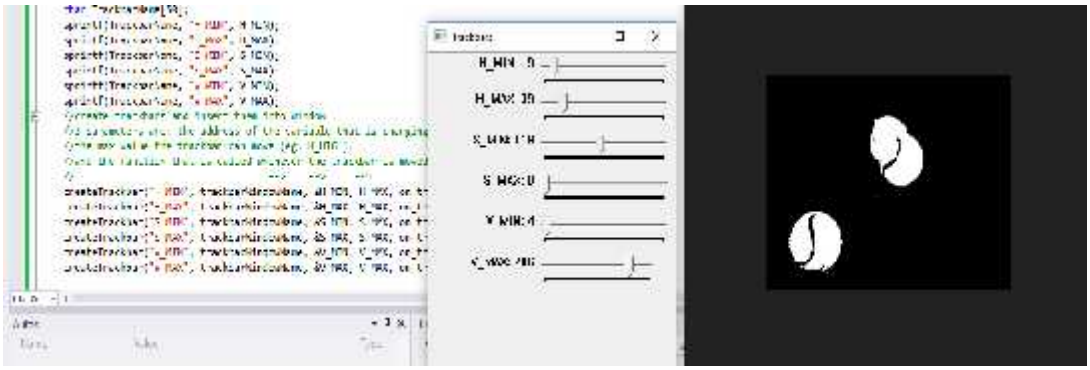

Figure 2.3: Tennis Balls Image Threshold[5]

Figure 5 shows how the photo appears after Threshold.

2.4 Cameras Calibration

Camera calibration is an important step in our project, without camera calibration we are not able to detect the real world coordinates of balls.

Camera calibration goal is to obtain the intrinsic and extrinsic parameters of a camera and lens.

Extrinsic parameters are:

- Translation.
- Rotation.

Intrinsic parameters are:

- Focal length.
- Principle point.
- Lens distortion.
- Color sensitivity.

These parameter are used to

1. Correct images
   - Unipolar rectification.
   - Distortion removal.
   - Color balance correction.
2. Compute distance via triangulation, stereo matching or multi-view reconstruction.

Pinhole Model:

$$sm' = A\ R|t\ M'$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where:

- (X, Y, Z) are the coordinates of a 3D point in the world coordinate space
- (u, v) are the coordinates of the projection point in pixels
- A is a camera matrix, or a matrix of intrinsic parameters, intrinsic parameters does not depend on the scene viewed
- (c_x, c_y) is a principal point that is usually at the image center
- (f_x, f_y) are the focal lengths expressed in pixel units.

- joint rotation-translation matrix $[R|t]$ is called a matrix of extrinsic parameters, it used to describe the camera motion around a static scene[5][6].

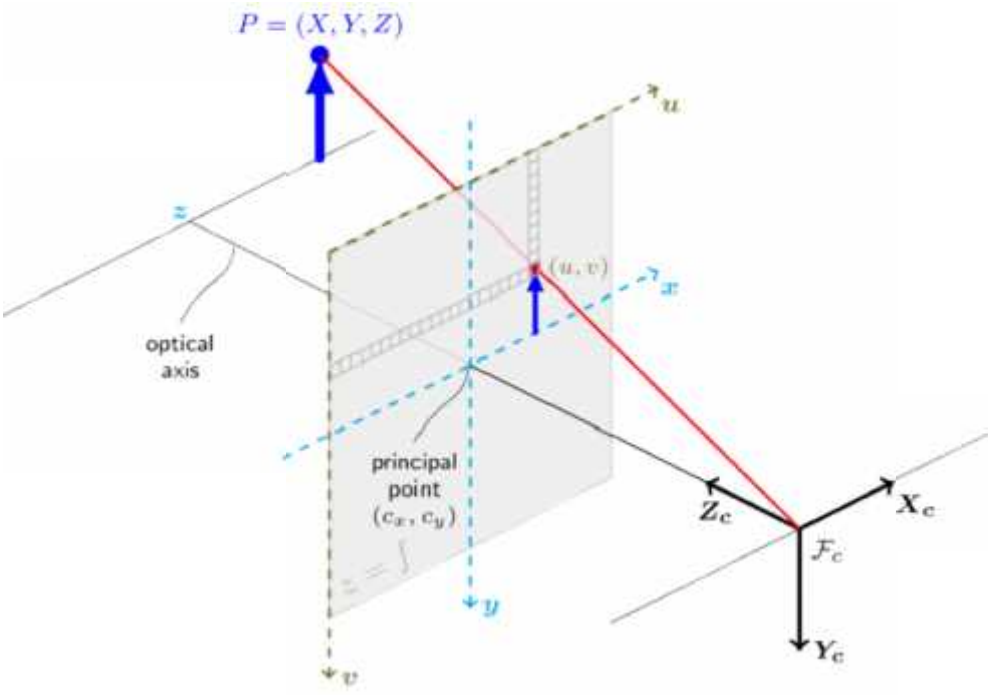

Figure 2.4 Pine Hole Model[5].

2.5 Distortion

Also camera lens will make distortion which effect the image as followed

Figure 2.5 Distortion Effects [5].

And the distortion coefficients are$(k_1, k_2, p_1, p_2[, k_3[, k_4 k_5 k_6]])$. Where $k_1 k_2 k_3 k_4$ $k_5$and k_6 are radial distortion coefficients. $p_1$and p_2 are tangential distortion coefficients. These coefficients are use as 4, 5 or 8 vector. [5][6].

2.6 Estimating Camera Pose:

It is a technique, to match the real world points to the corresponding points in the image in order to find the rotation and translation matrices, which we will use to get the real world coordinates of the point in an image. [5] [6].

2.7 Finding real world location

As the camera intrinsic and extrinsic parameters are found(camera matrix, rotation matrix, translation matrix), a real world coordinate of any image of the image could be obtained using the following equations:

$$s \begin{matrix} u \\ v \\ 1 \end{matrix} = M(R \begin{matrix} X \\ Y \\ Z_{const} \end{matrix} + t)$$

Where M is camera matrix, R – rotation matrix, t – translation matrix, and s is an unknown. Zconst represents the height which is near to zero in our case.

The previous equation helps to finding which we need in the next equation to find real world coordinates.

As S is found we can solve this equation to find (x, y) for any pixel in the image.

$$R^{-1}M^{-1}s \begin{matrix} u \\ v \\ 1 \end{matrix} = \begin{matrix} x \\ y \\ Z_{const} \end{matrix} + R^{-1}t$$

This method helps finding real world coordinate of a plane, which is the solution which is needed at our problem.

Theimportance of the extrinsic parameters that they used to find the relative Rotation and relative translation of each camera to the space that represents our coordinate system. [5][6].

2.8Communication Protocol

The communication method to be used is SOAP.

“SOAP (Simple Object Access Protocol) is a protocol specification for exchanging structured information in the implementation of web services in computer networks. Its purpose is to induce extensibility, neutrality and independence. It uses XML Information Set for its message format, and relies on application layer protocols, most often Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

SOAP allows processes running on disparate operating systems (such as Windows and Linux) to communicate using Extensible Markup Language (XML). Since Web protocols like HTTP are installed and running on all operating systems, SOAP allows clients to invoke web services and receive responses independent of language and platforms.”[7]

This SOAP packet will be structured to have the Path and the robot location and send them to on-board system, which will be able to parse it and behave according to the data transmitted to it.

2.9 Path

The path will be sent to the robot as a sorted array of location of tennis balls which presented as (x, y).

2.10 Robot Localization

A pattern which is easy to recognize will be added to the robot in order to localize the robot and find its location.

Chapter 3 :Requirements

In this chapter we discuss the main requirements of the project and what are the expected results of our system.

3.1 Main Requirements

The main four requirements of the project are finding real world coordinates of tennis balls, find location and direction of the robot, find the path of the robot to collect the tennis balls and send navigation data to the on board intelligent system.

3.1.1 Finding real world coordinates of tennis balls

In this requirement, the vision system will detect all the balls in the site through the images captured by cameras, locate them in the image and find their centers, multiple cameras may find a ball or more of same location as different ball which will cause a redundancy of the number of the balls.

The vision system must be able to locate the real world location of the tennis balls using the centers in images, and reduce the redundant locations in the final result.

3.1.2 Find location and direction of the robot

In this requirement, the vision will detect specific pattern in the image, these patterns are installed at front and back of the robot, so that the vision system could find the robot location in the real world, and then calculate its direction.

3.1.3 Find the path of the robot to collect the tennis balls

In this requirement, the vision system solves the distances and the angles between the robot and the balls, and then uses this information to find the best order to go through as the path of the robot.

The time Factor is important in this requirement; since the project will run in the real-time, so this requirement must not be implemented in high order complexity of time.

3.1.4 Send navigation data to the on board intelligent system

In this requirement, the vision system will send the navigation data to the robot through the network and will update each time it finds an update of the path or the robot location.

3.2 Expected Results

The vision system will be able to detect , locate tennis balls and find the shortest path for the robot. Then send navigation data to the robot to collect tennis balls.

Navigation data consists of the robot site, the tennis venue, and the shortest way to collect it.

Chapter 4 :Design Options

In this chapter, we will discuss our design options and its alternativesand comparison between them, and then we will mention what we choose and what is the justification of our choice.

The first design option is the omnidirectional vision sensor and the second is foursurveillance cameras around the playground.

4.1 Omnidirectional vision sensor

The Omni Directional Sensor consists of a convex mirror place on c-mount lens at the opposite side of a camera which captures the around area of the camera, the next figure shows how omnidirectional Sensor Implemented.

Figure 4.1 Omnidirectional vision sensor [8]

The next Figure shows how omnidirectional vision sensor constructed on a robot, as much higher it constructed the much wider angle it capture.

Figure 4.2 Omnidirectional constructed on robot [9]

The next Figure shows a captured image using omnidirectional vision sensor, the main problem with the omnidirectional that is the captured images has a high distortion, as we undistort the images, an multiple of image details will be lost, which leads us to information lost.

Figure 4.3 image taken by omnidirectional[10]

The previous Omnidirectional cost is EUR 525, and does not include fees, shopping costs and a c-mount camera.

4.2 Four surveillance cameras

In this design option 4 surveillance cameras will be constructed in the top four corners of the tennis playground.

Figure 4.4: Surveillance Camera[11]

The main issue to consider in this design option is to choose the quality of the cameras to fit the network bandwidth and will be installed to allow the vision system to access these four cameras.

The cost of each camera is 25 to 125 dollars depending on the quality and features, in addition to the cost of cameras, the network will cost 40-60 dollars.

4.3 Comparison between the alternatives

The caparison will be done according to cost, speed, accuracy,construction and availability.

In terms of cost, the omnidirectional vision sensor will cost more than the four surveillance cameras since its cost without c-mount camera and fees equal to the four surveillance cameras.

In terms of speed, the omnidirectional will be faster since 1 image need to be pre-process, analysis and extract information from it, on the other hand, instead of 1 image there will be four images.

In terms of accuracy, omnidirectional vision sensor will be less accurate since we need to undistort the image and that will cause loss of details and information.

In terms of construction, the omnidirectional will be easier to construct since it's going to be constructed on the robot and connected directly to the onboard intelligent system, as the opposite of the four surveillance cameras which will need a network in addition.

In terms of availability, the Japanese omnidirectional vision sensor has stopped production, and we have only found one at a distributor in Italy. On the one hand, surveillance cameras are widely available in local stores, and there is no need to wait for delivery.

4.4 The decision

We decided to choose four surveillance cameras; since it satisfies the following reasons:

- Cost is within the scope of cost of the project.
- Speed is acceptable.
- Accuracy satisfies the project needs.
- And widely available in local market.

Chapter 5 :Hardware Design

In this chapter, hardware component of the system will be mentioned along with schematic diagram shows how they will be connected, with explanation.

1. Four surveillance cameras

   There are two types of surveillance cameras, CCTV cameras, IP cameras, CCTV using a coaxial cable to transmit data as analog signals and storing them in a digital video recorder (DVR), CCTV is difficult to deal with; because we could not find out how the recorded video clips Video encoded in a DVR, the IP camera is easy to access and does not require a network video recorder; since we can access it using CAT5 Ethernet cable using its IP, our choice is IP camerasCAT5 cables to connect IP cameras to switch.
2. Switch, there are two choices of the switches, standard switches and POE switch which can provide IP cameras with power; Since the POE switch is much higher cost than standard switch, we will use the AC-DC adapter to provide power to IP cameras.
3. PC; Since the vision system will be implemented on PC.
4. Robot which will collect the tennis ball, with a NI myRio microcontroller at the on board intelligent system, the robot will communicate with the vision using wireless connection; since myRio has wireless adapter.

The next Figure shows how hardware component will be connected.

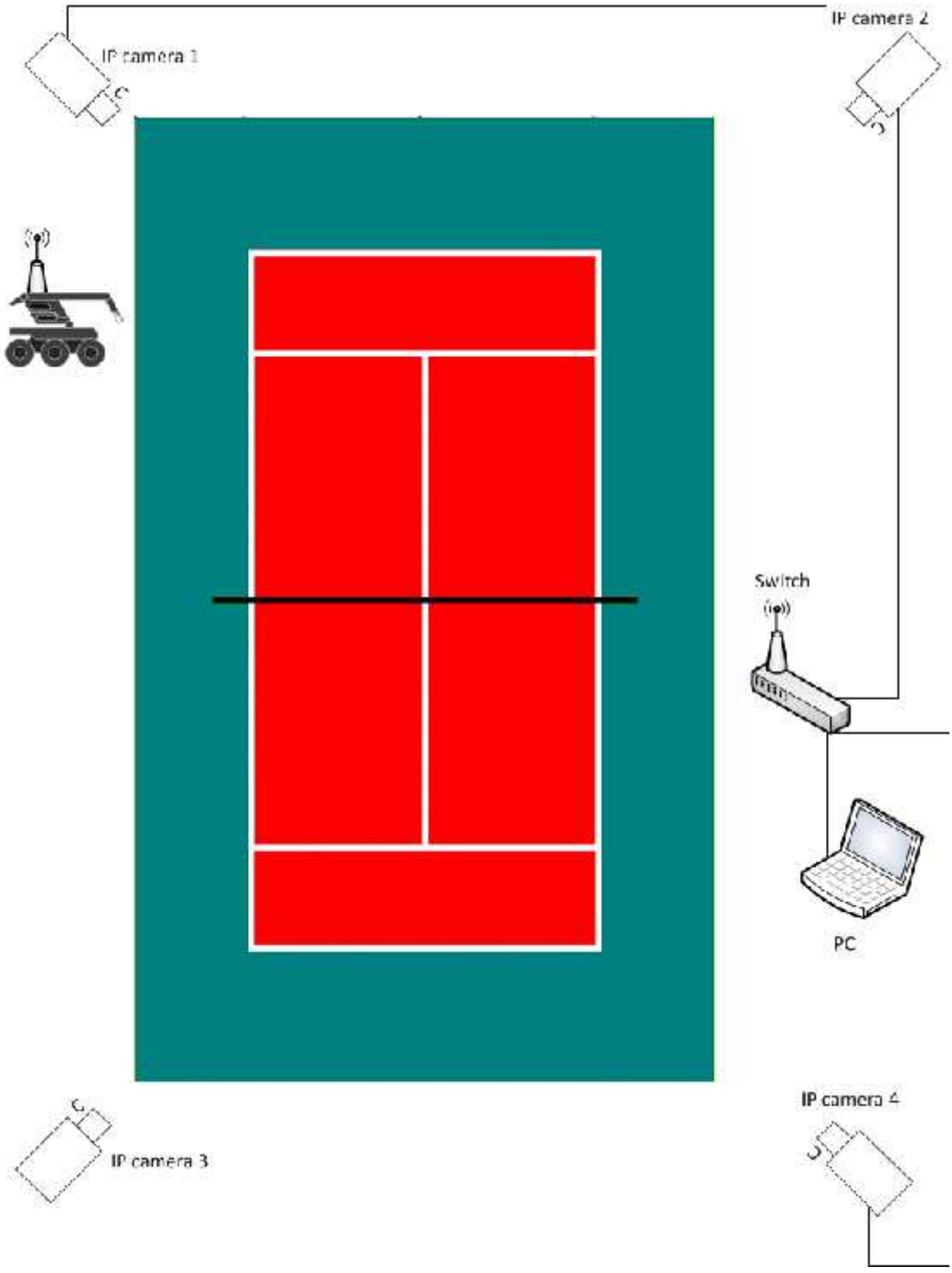

Figure 5.1 Connection of hardware components

Chapter 6 Software Design

6.1 Software Environment

The Vision System will be implemented under Windows operating System, using C++ as programming language along with openCV library, the tools which will be used are Microsoft visual studio 2013 and Matlab 2017a.

6.2 Configuration Tasks

These tasks must be done after installing the cameras to enable the vision system to run and must be done to all the cameras, the output of these task is the input of requirements tasks.

The configurations tasks are:

1. Calibrate the camera using Matlab, the output of this task is the camera matrix and distortion coefficients.
2. Estimating camera pose using camera matrix, distortion coefficients and reference points in image and their real world location, the output of this task are the rotation and translation of the camera, this task must done each time camera change its pose.

The next figure shows flow diagram of the configuration tasks,

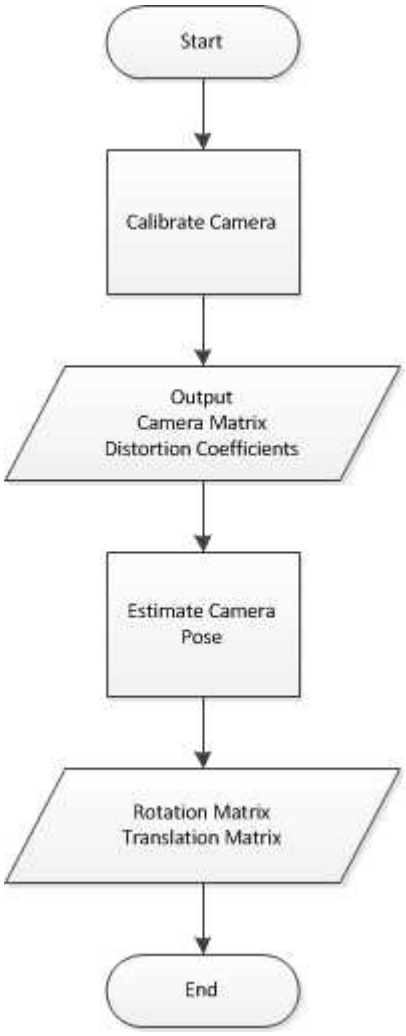

Figure 6.1 Flow diagram of configuration tasks

6.3 Requirement Tasks

These task will be executing repeatedly in the run time, these task are:

1. Capture Image from camera, this task will be execute in parallel four times in order to capture four images from each camera.
2. Detect tennis balls in the image and find their centers, this task will be execute in parallel four times in order to detect all tennis balls in the playground,
3. Detect the Robot in image and find its front and back.
4. Find Real world locations of the tennis balls.
5. Find the real world location of the robot.
6. Reduce repetition in tennis balls.
7. Find a shorter path.

8. Send the navigation data to the robot

The Next Figure shows how the flow diagram of the requirements tasks,

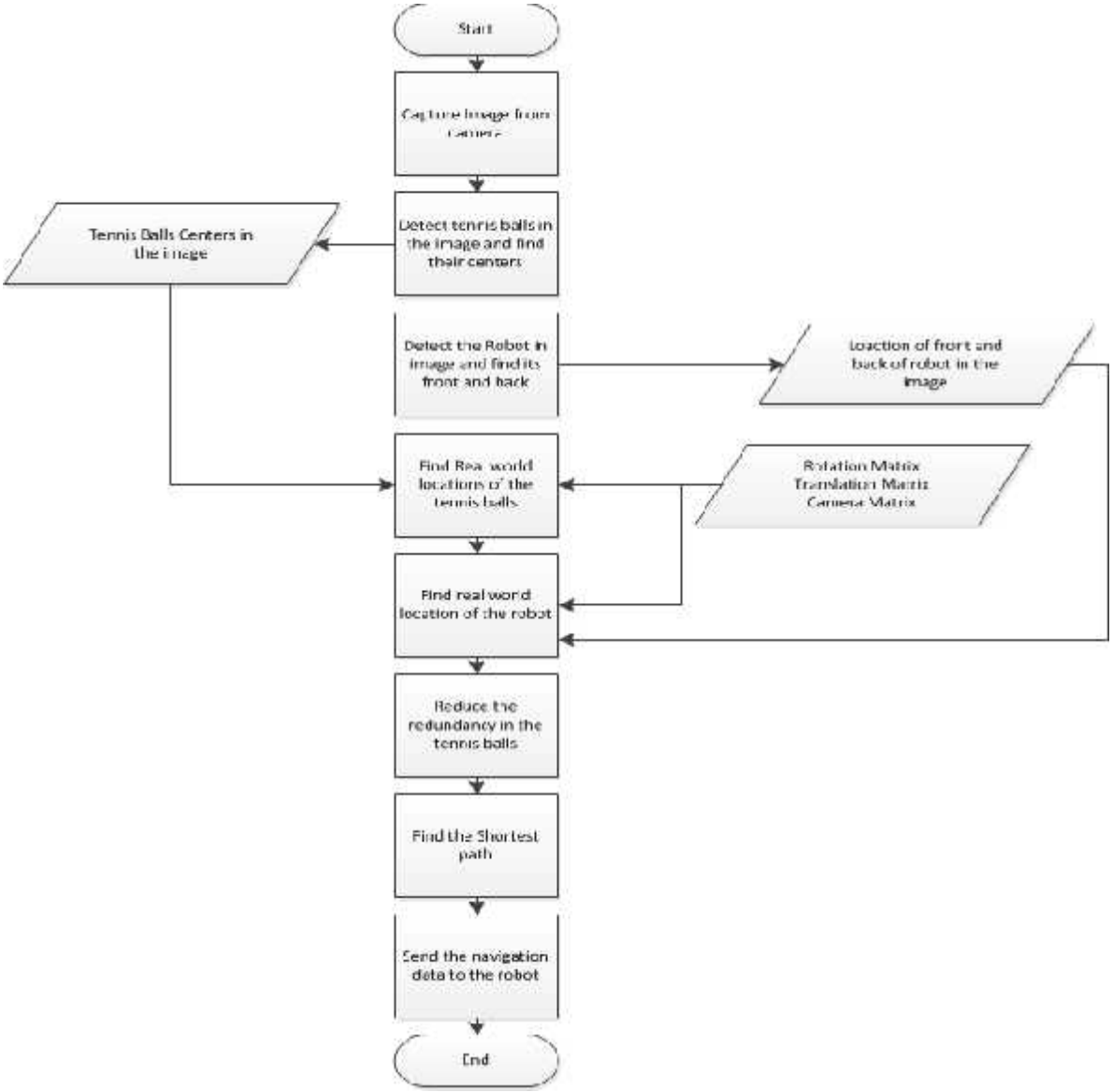

Figure 6.2 Flow diagram of requirements tasks

Chapter 7 :Implementation

In this chapter, we discuss the issues of implementation of each task in our project

7.1 Ball Detection

The objective of detecting the ball to find the vector of the center of the mass of each ball in the image, in order to use the vector to obtain the real world format of the balls and finished as follows :

- Color base detection for the tennis ball
- Median filter in order with kernel 5x5 in order to remove noise and the white line at the middle of the ball
- Edge detection to find the contours of the balls, the edge detection helps detect multiply balls
- Finding center of mass for each contour which represent the center of ball in the image

7.2 Camera Calibration

In this step we found the camera matrix and Distortion coefficients using Matlab Single camera calibration toolbox, this step must be done for each camera, in order to do it, an images of chessboard must be taken, this chess board should contain an even number of squares on the first side and the other contains odd number of squares.

Matlab Camera Calibration Tool need from 10 to 20 images so it can find the camera inartistic and distortion confidents of the camera.

This step doesn't need a camera to be installed to fixed place, but it must not change its location at the duration of camera capturing the images.

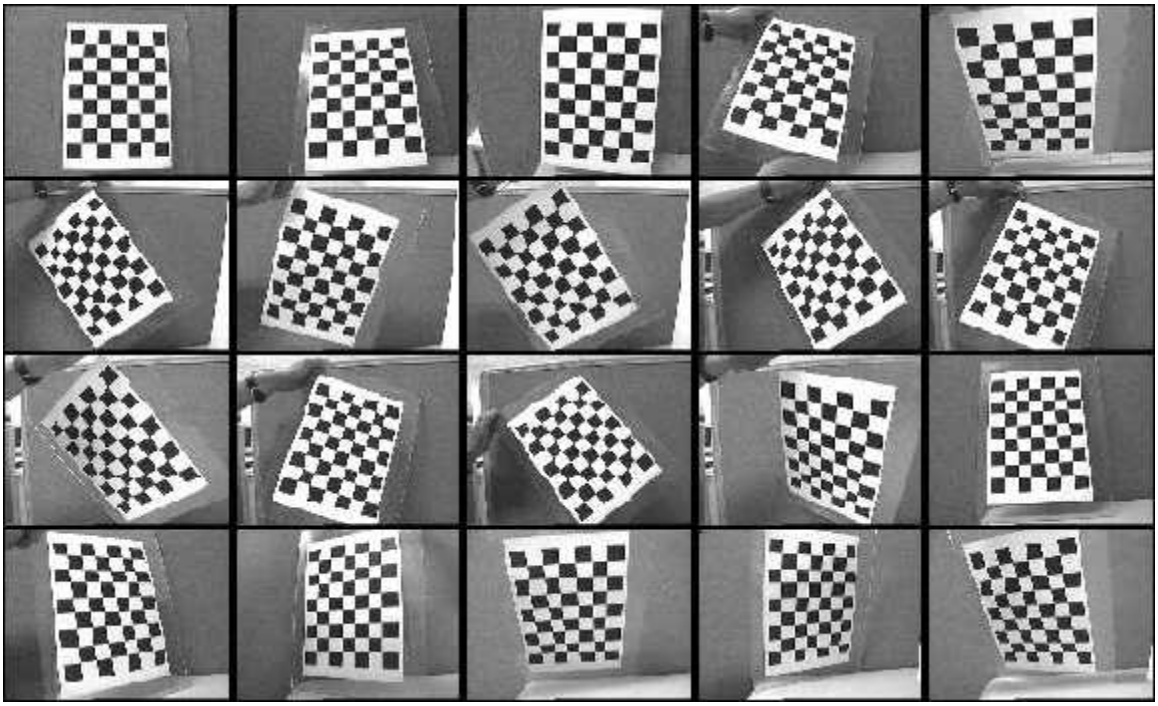

Figure 7.1 sample of calibration images [12]

The Size of the square of the chessboard is needed in the calibration algorithm which used in Matlab Camera Calibration Tool, and for an accurate measure, the square size is calculated as the following formula

$$square\ size = \frac{length\ of\ all\ squares\ of\ side}{number\ of\ squres}$$

After inserting images into the tool, the inner corners of the chessboard are wrapped as shown in the following figure.

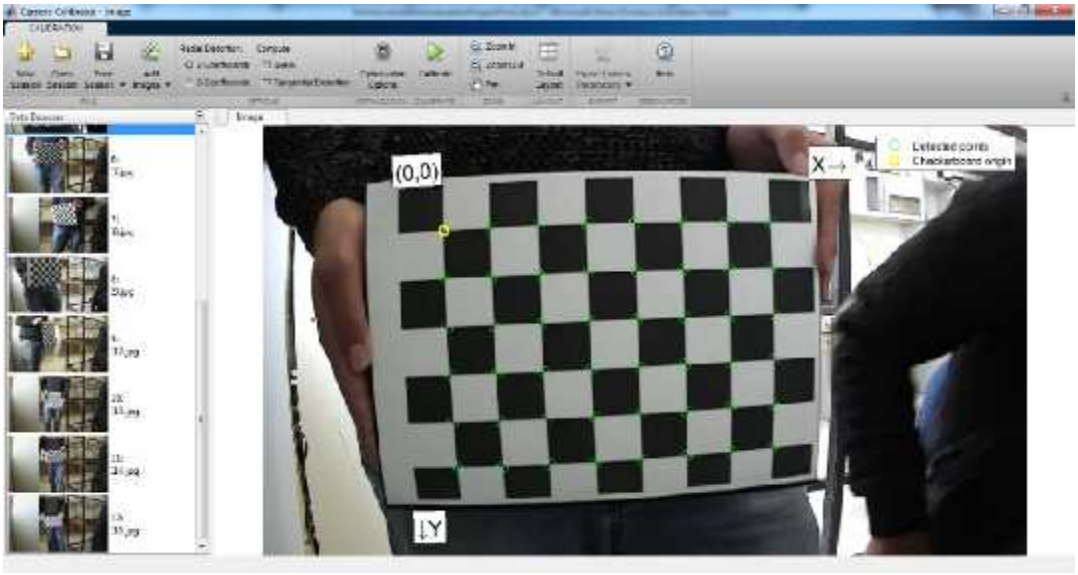

Figure 7.2 Chess board inner corner detected in Matlab[13]

The following figure shows the distortion factors obtained after calibration.

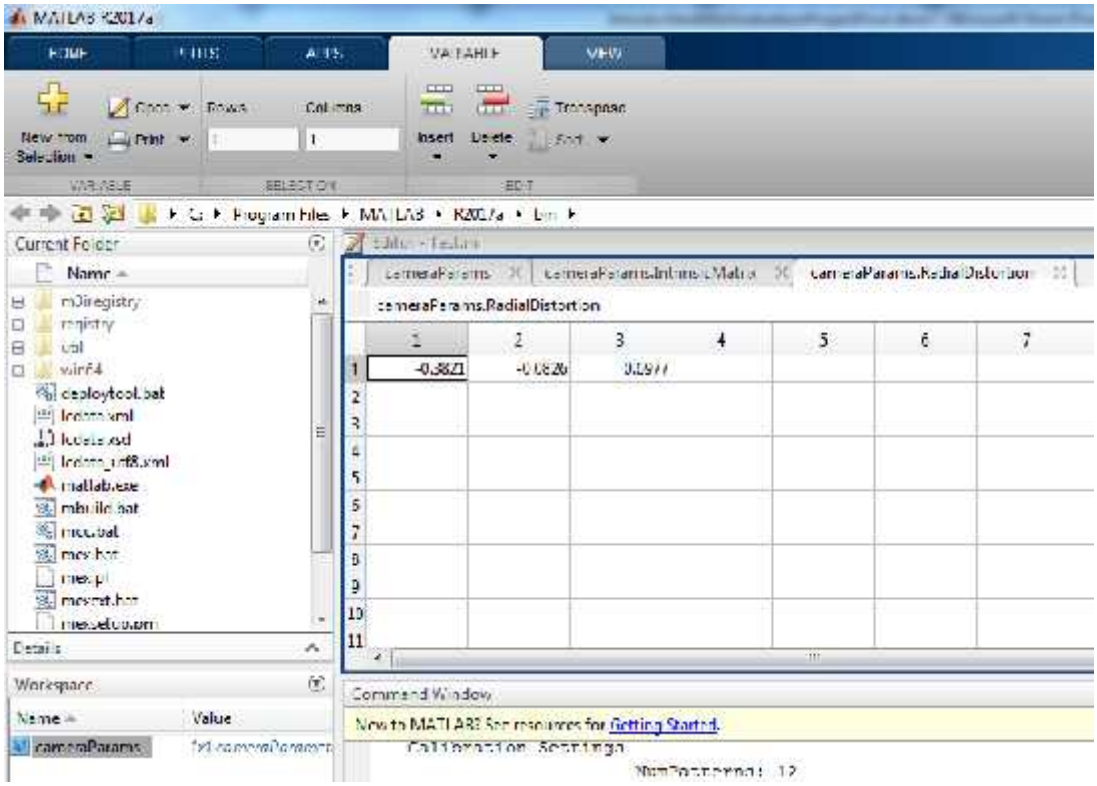

Figure 7.3 Distortion Coefficients[14]

The next figure shows the camera matrix which we obtain after the calibration is done.

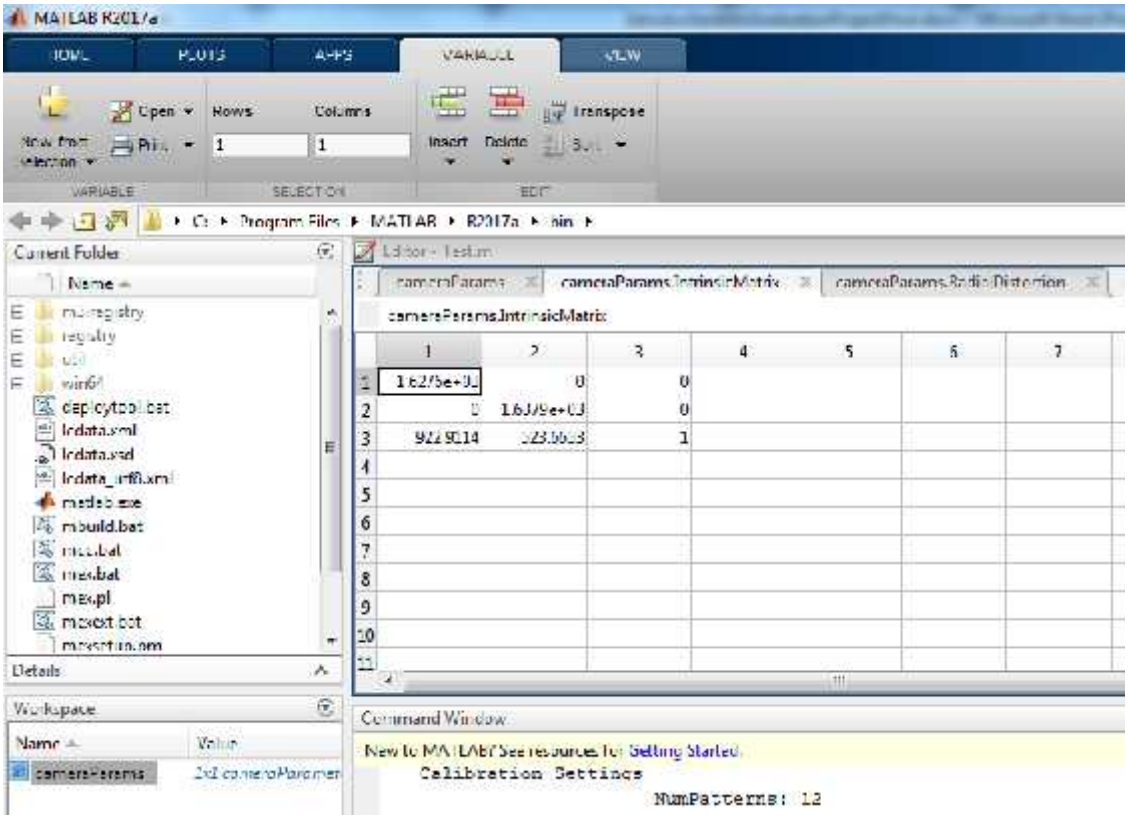

Figure 7.4 Camera Matrix[15]

If number of accepted images is between 10 -20 image, the tool can then estimate the intrinsic and extrinsic camera parameters, the next two figure shows the result of calibration which has been exported to Matlab work space

The result of calibration that we obtain the instratic and extraticparameter after the calibration is done, these values are needed in order to solve the camera pose.

7.3 Estimating Camera Pose

In this section, the result of estimating camera pose are the rotation and translation matrix which will be used to obtain the real world location from pixel coordinates, we use the camera matrix and distortion coefficients along with reference points image along with their corresponding real world coordinates to find rotation and translation matrix, this method is done by using function SolvePnP in opencv, which estimates the camera pose.

Figure 7.5 Estimating camera pose for 1 camera[16]

The previous Figure shows the axis of the real world location after estimating the camera pose, the blue is the z axis, red is the x axis and the green is the y axis.

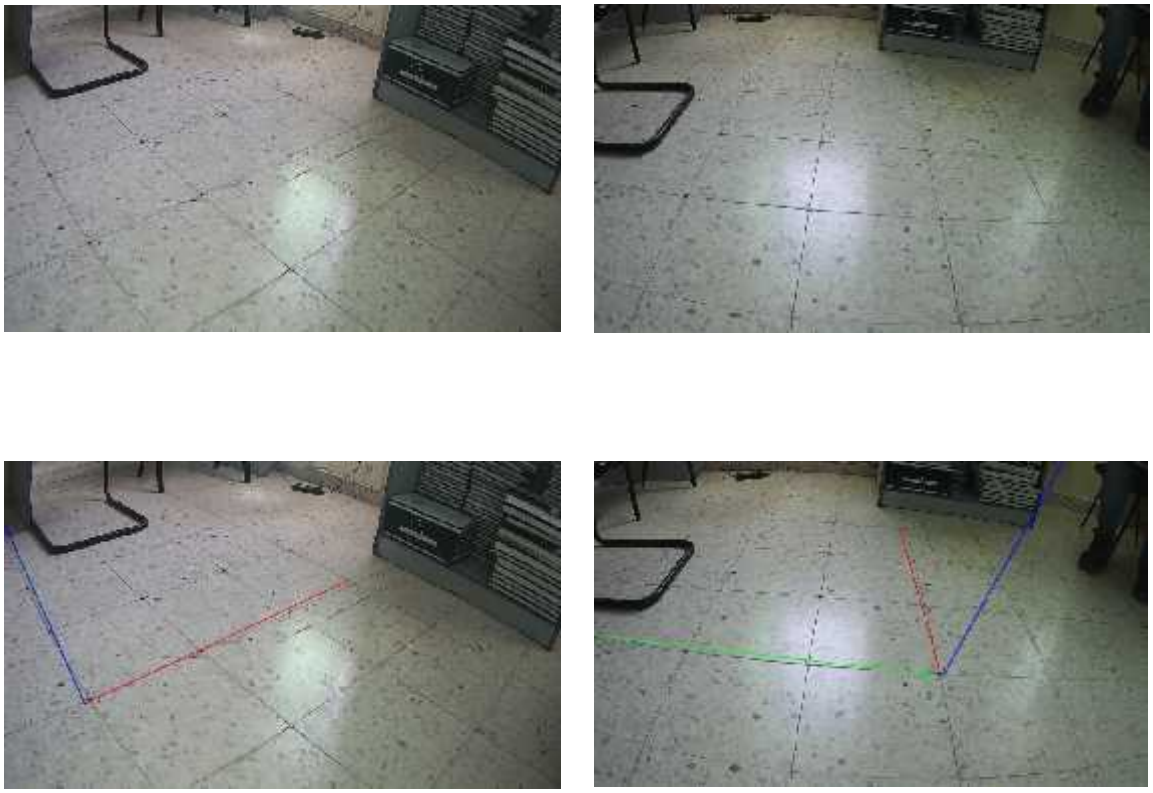

Figure 7.6 Estimating camera pose for 2 cameras[17]

The previous two figures shows the result of estimating camera pose for two cameras on the same site, with the same reference points and their corresponding real world locations, the blue is the z axis, red is the x axis and the green is the y axis, the point which we proofed that we can use multiply cameras on the same site if only the reference points are common between the two sites.

7.4 Coordinate System

In this section, the guide line of how to define a real world coordinates for the site will be explained

Coordinate system which must be in millimeter units; since the Matlab Camera Calibration Tool calibrate base on the millimeter.

In vision system which we implemented the coordinate system could be defined easily by using for reference Ping-Pong balls, the vision system will detect them and find their center in image, the only think the user enters is their real world coordinates.

7.5 Obtain Real World Location

As we done all the steps above, we had all the parameters needed to obtain real world coordinates, we have applied the reverse pinhole model equation to obtain real world coordinates for specific point of the image as shown in the next two figures.

Figure 7.7 specific point in image[18]

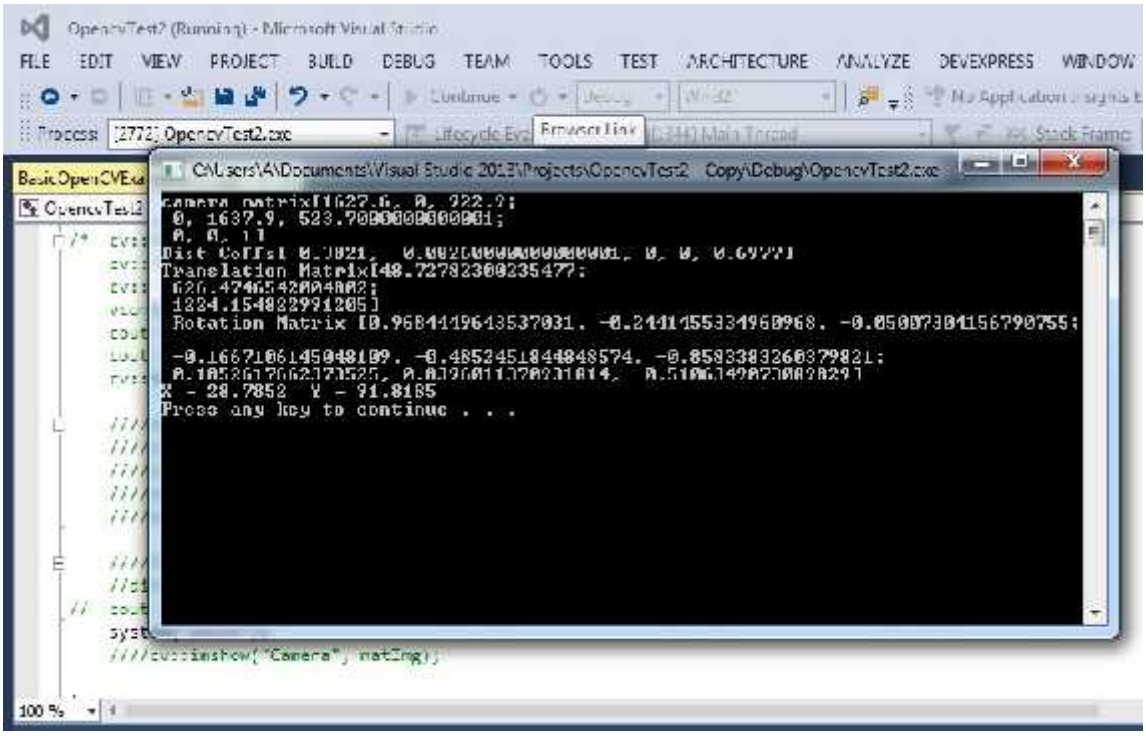

Figure 7.8 Real world location of specific point[19]

7.6 Robot Localization

It's an important task to localize the robot while the system is running in the real Time; the result of the robot location will be used in the path planning step.

In this Step we supposed that the robot is a box, then we added two different color balls at the top of the box as markers, one of them marks the front of the robot, and the other marks the back of the robot, by detect these markers and their centers we was able to find the real world location of them.

Figure 7.9 Robot simulation as box[20]

The robot can be simulated as a vector in the space, which has a position and angle; the position is the front of the robot and the angle will define the direction.

The angle is calculated as the Tan inverse of the vector and has a value ranging from -180 to 180; so the line is at the 0 ° angle.

7.7 Enhance the real time processing

Since the vision system must handle 4 images for the navigation system, we need parallel parallel processing, so we use the std :: future <T> and std :: Async (F) object in C ++, these features in C ++ Enable us to run methods at one time, so we can reduce the processing in real time.

7.8 Ball Aggregation

The advantage of using multiple cameras is to get a wider angle of the stadium, but many balls may be discovered more than once, and to solve this problem, we implement a medium between the sites near the balls, this site may point to one ball or two balls next to each other; Because the robots collection mechanism is 70 cm, it will be an advantage to merge the nearby balls to each other.

7.9 Communication

Web service was implemented in order to establish communication between the vision system and the robot, web service contain four web methods and Flag.

These methods are:

- The first web method which will be used from vision system to set Robot Location Values
- The second web method will also be used from vision system to set Balls Location Values
- The Third web method which will be used from Robot to get Robot Location Values
- The Fourth web method which will be used from Robot to get Balls Location Values
- The Flag is used to state if the robot had read the current values from web service.

7.10 Synchronization

In this section, we will mention how data will be transferred between the vision system and the robot through the web service.

As Vision system started, it will get into iteration, each iteration ends with updating values of web service, the Robot Will request a new values if the flag state that the Values aren't read by robot yet, and direct to the next location according to the values.

As soon as the robot reaches its new location, it will request a new data from web service, if the flag state a new data, the robot will get it, if not it will direct for the next location.

7.11 Path Planning

Since the vision system will run in the real-time, path planning must be implemented as a fast as possible, for the path we use weighted sum of the distance and the angle, we assume the least result represent the highest priority as the following:

$$Priority = W_1 * Distance + W_2 * abs(angle)$$

Where:

- W_1: the weight of the Distance
- W_2: the weight of the angle
- Distance: the Distance between the front of robot and the ball
- Angle: the Angle between the robot and the ball which is between -180 and 180.

Chapter 8 : Testing

In this chapter we will mention how did we made the test and explain the our result.

First of all we as mentioned in section 7.6; the robot was simulated as box which marked by two colored balls, the blue represent the back of the robot and the pink present the front of the robot.

In the first step in the test a set of tennis balls was thrown on the ground and the box was placed on the ground in random placed. We marked each ball with its Indexed number which is generated by the system automatically.

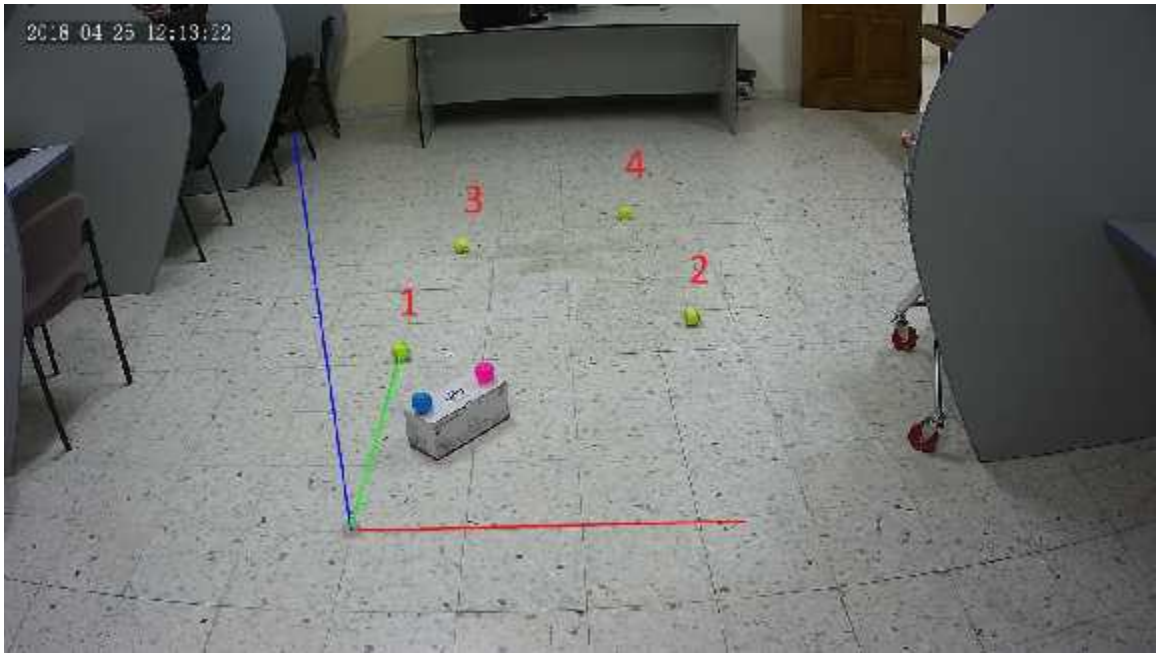

Figure 8.1: Test Simulation image 1

As vision system started it detects, recognizes and localizes the tennis balls and box (front, back). In addition to that, the system calculates the distances between all the balls and the system, and in which angle its place in relative to the box, as shown in the next figure.

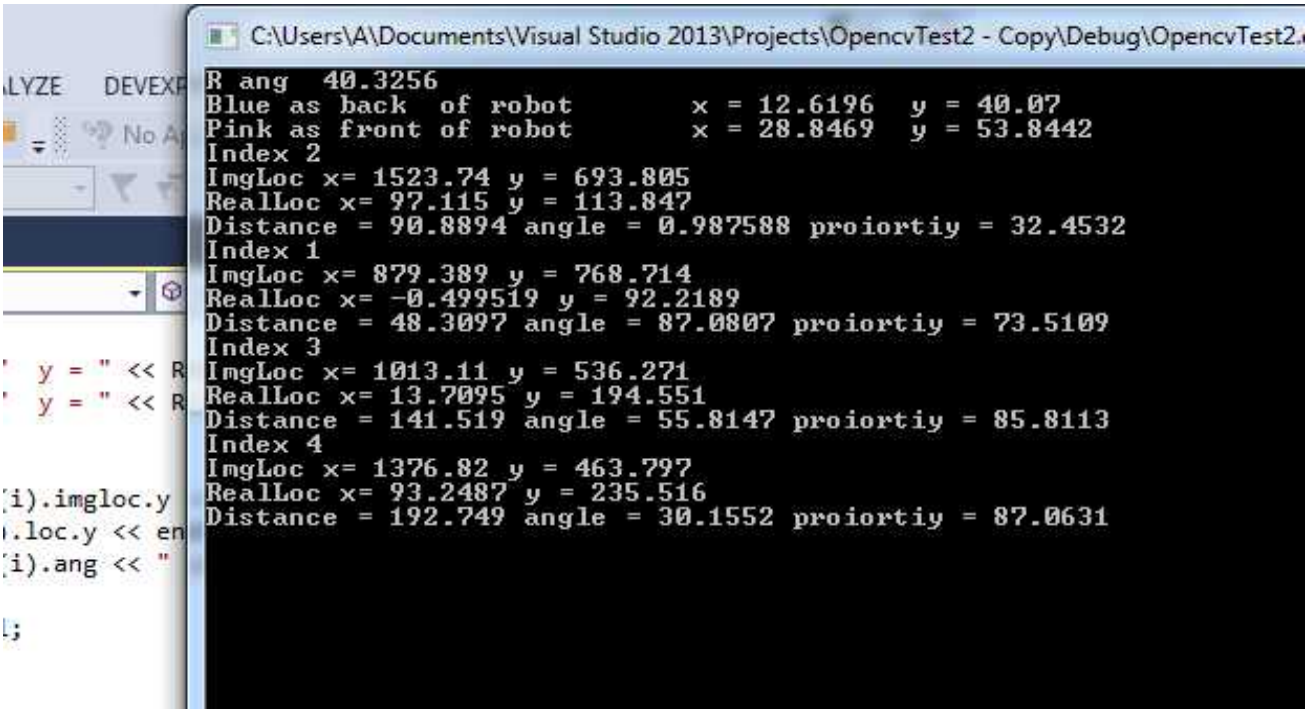

Figure 8.2: Result of Test Simulation image 1

The previous figure shows the number of the detected ball, their locations, their distances and the angle relative to the robot, the highest priority balls has the lowest number.

In this result, we can clearly notice that the ball index 1 in the least distance but the ball index 2 is the highest priority; since the path was implemented as weighted sum between distance and the angle, the ball index 2 take higher priorities than ball index 1, and the explanation that we design the system to avoid sharp angle if possible.

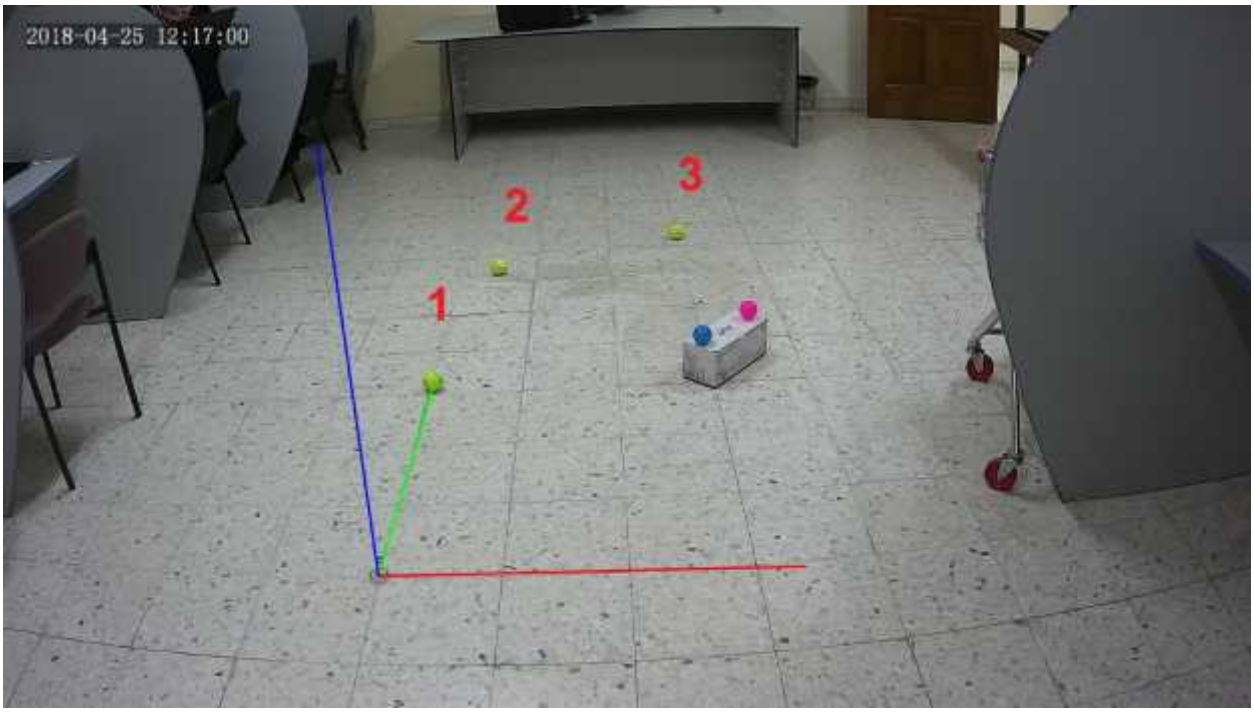

Figure 8.3: Test Simulation image 2

The previous figure shows the image after removing the ball number 2 and place the box on its place manually, since the index numbering is auto generated, we marked the balls again manually, the vision system updated the result of the distances and angles based on box location as shown in the next figure.

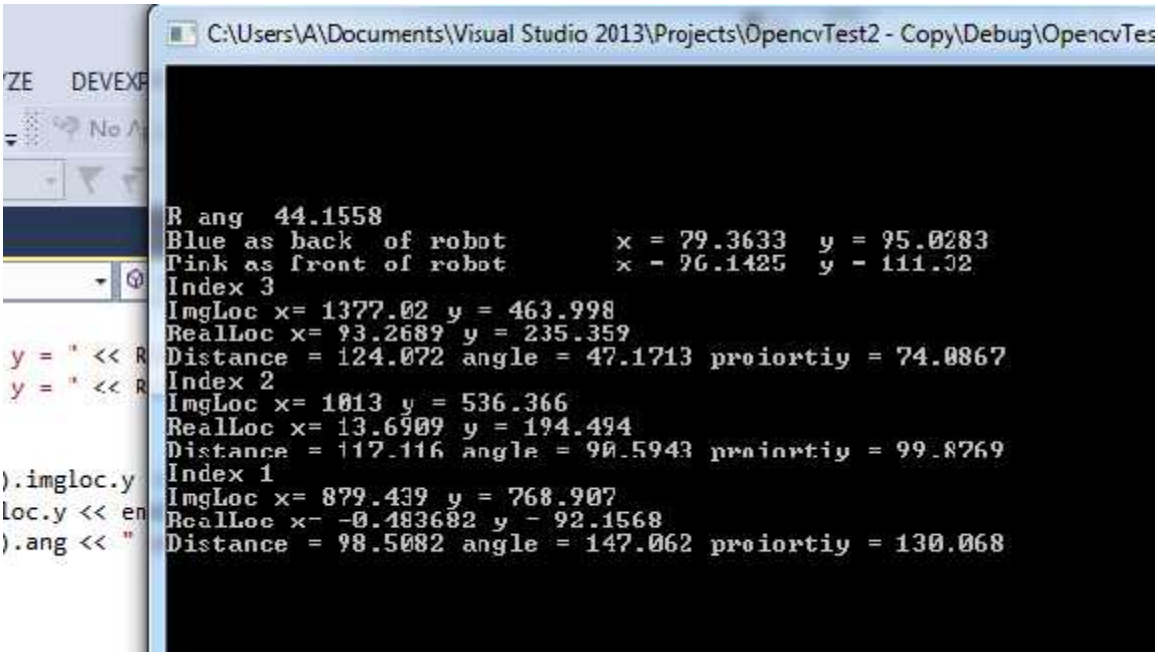

Figure8.4: Result of Test Simulation image 2

The highest priority ball after the update was ball index 3 according to the output of the vision system in the previous Figure.

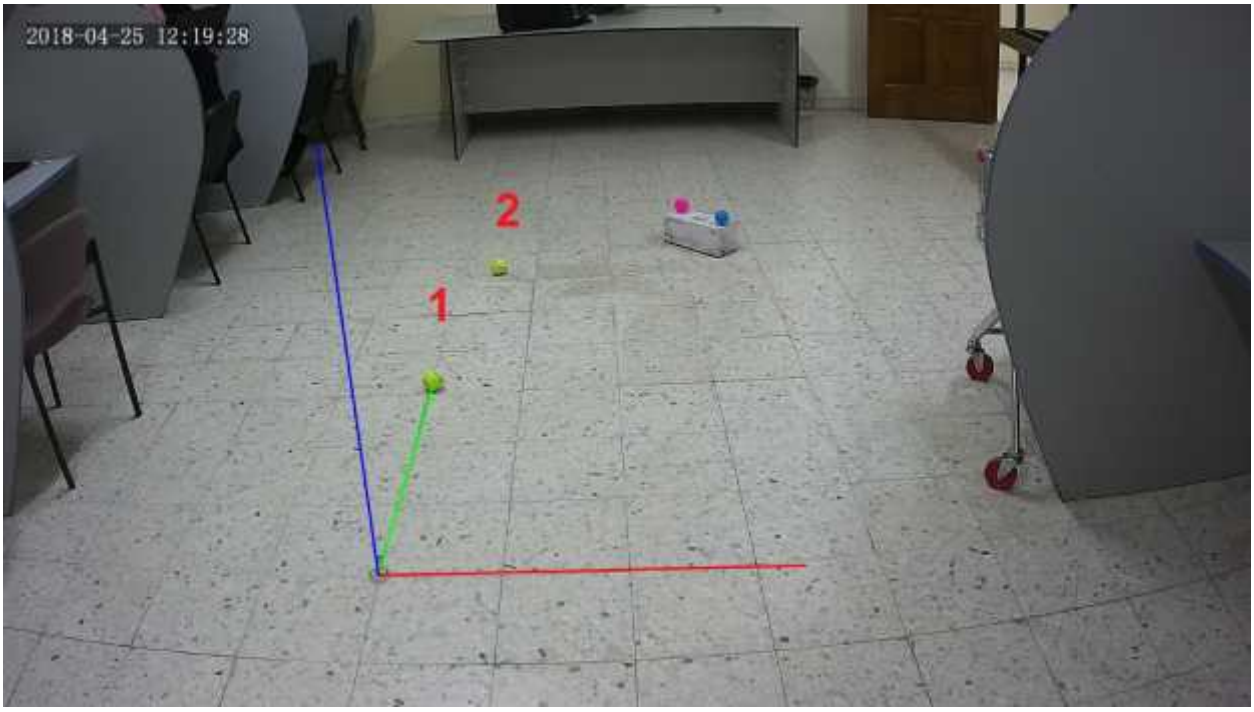

Figure 8.5: Test Simulation image 3

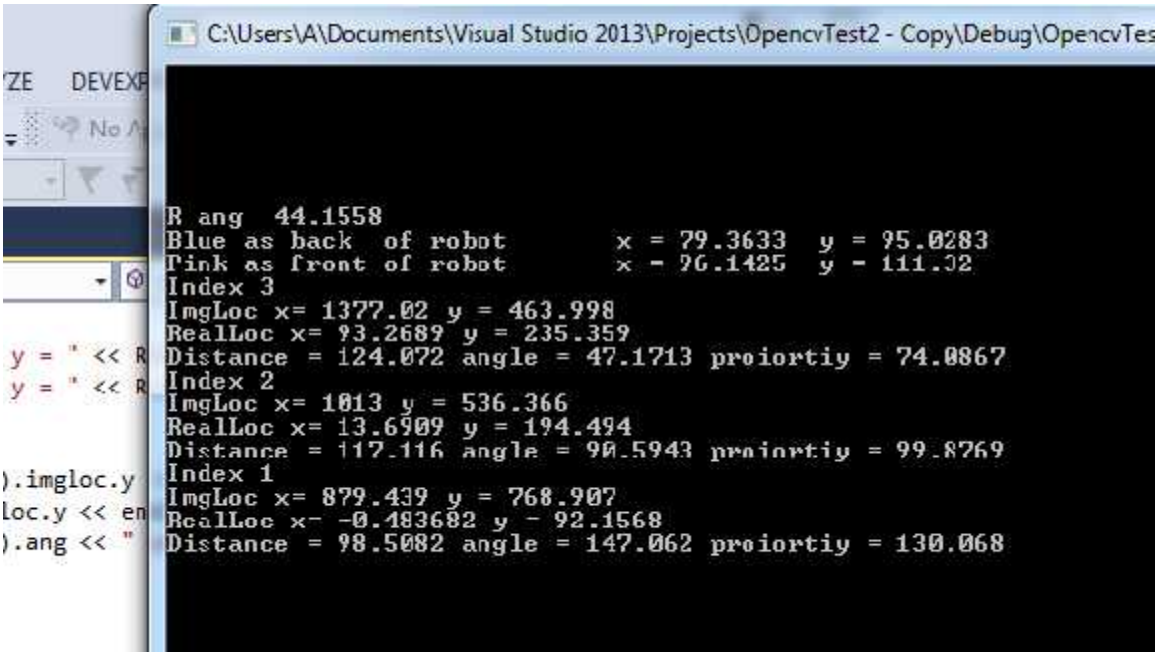

Figure8.6: Result of Test Simulation image 3

In the previous two figure, the ball index 3 was removed and the box placed on its place, the system updates the distances and angles again based on the box location, the result of previous result shows that the next ball will be ball index 2.

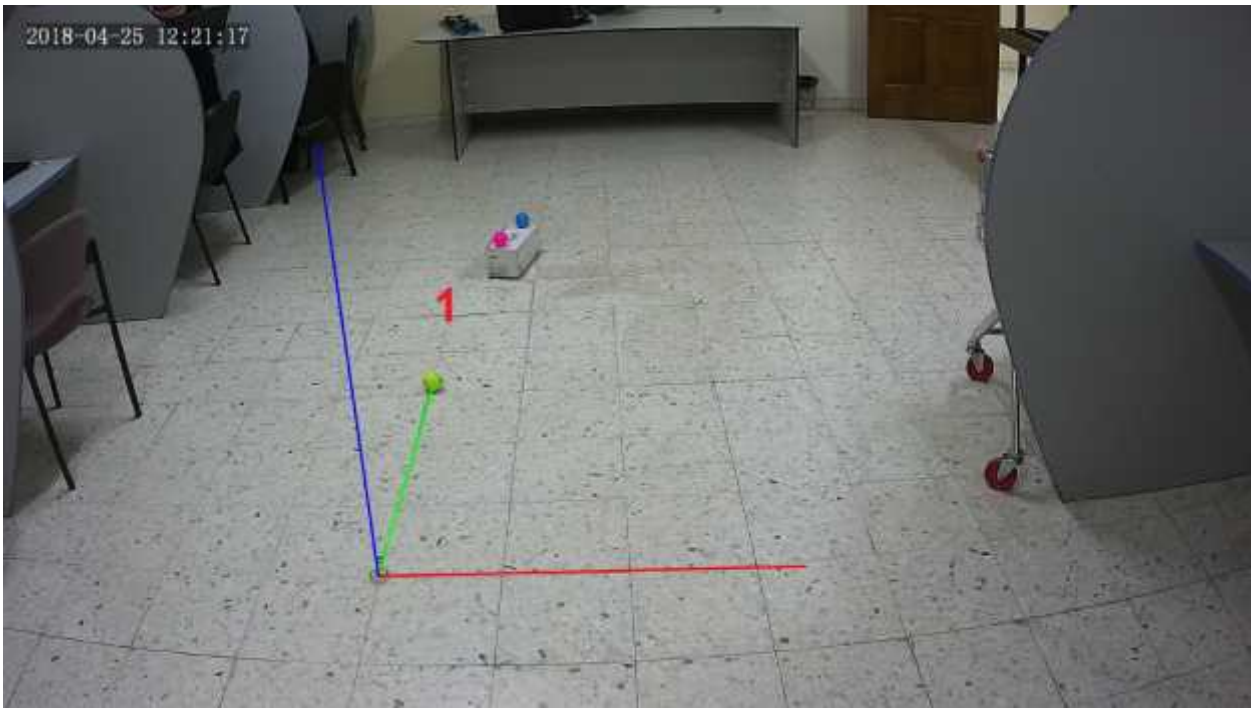

Figure 8.7: Test Simulation image 4

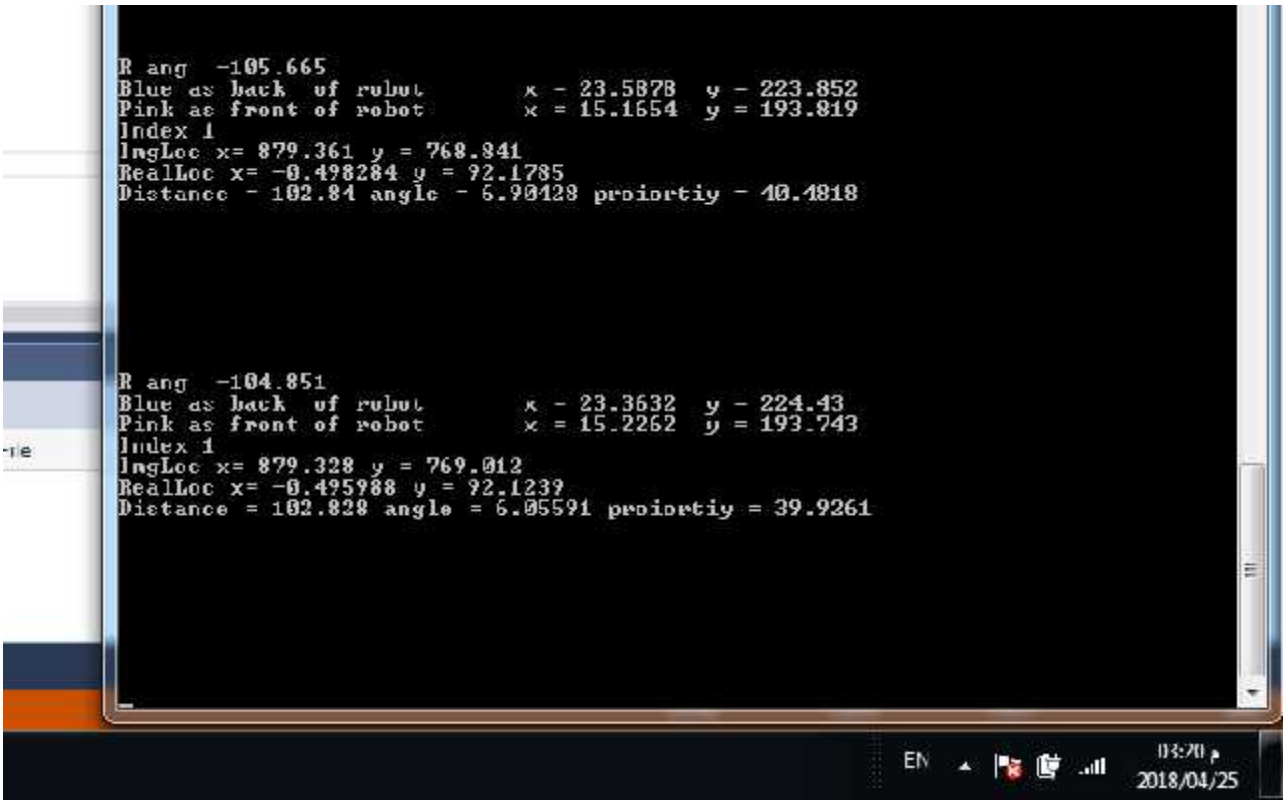

Figure8.8: Result of Test Simulation image 4

The previous two figures show how the results after removing ball index 2 and place the box in its location, the previous figure shows how results after the vision system update distances and angles based on the box location.

Conclusions

The proposed system, which was designed with highly efficient and effective algorithms, has demonstrated its ability to track a tennis balls accurately and rapidly. The proposed algorithms can overcome these detection challenges. This shows the proposed system can be contributed to an automatic umpiring system. Furthermore, the techniques are applicable to the more general problem of tracking fast-moving small round objects and their low cost opens up the prospect of improved access by amateur users.

References

[1] “Digital Image Processing Techniques for Object Detection From Complex Background Image R. Hussin*, M. RizonJuhari, Ng Wei Kang, R.C.Ismail, A.Kamarudin” https://ac.els-cdn.com/S1877705812025684/1-s2.0-S1877705812025684-main.pdf?_tid=efbb9268-e251-11e7-b43a-00000aab0f6c&acdnat=1513422900_1d2b7ada468b6bd83b19b808098d552f [Accessed: Dec. 10,2017].

[2]

[3] https://www.kirupa.com/design/little_about_color_hsv_rgb.htm [Accessed: Dec. 10, 2017].

[4]https://www.elphel.com/wiki/OpenCV_Tennis_balls_recognizing_tutorial [Accessed: Dec. 10, 2017].

[5]https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[6] https://stackoverflow.com/questions/12299870/computing-x-y-coordinate-3d-from-image-point

[7]https://www.elphel.com/wiki/OpenCV_Tennis_balls_recognizing_tutorial [Accessed: Dec. 10, 2017].

[8]http://www.vstone.co.jp/english/products/sensor_camera/img/img_robot01_big.jpg

[9]https://www.ti.uni-bielefeld.de/downloads/equipment/PioneerPanoramaFront_small.jpg

[10]http://home.deec.uc.pt/~jpbar/MyImage/para2.jpg

[11] “https://www.lorextechnology.com/hd-ip-camera/3mp-hd-ip-camera-with-long-range-night-vision/LNB3163-Series-1-p” [Accessed: Dec. 10, 2017].

[12] http://yang.ming.free.fr/research/vision/calibration/in.jpg