MARKER-BASED TRACKING UNDER EXTREME ILLUMINATION:

FEASIBILITY STUDY AND PILOT TESTING

MUHAMAD AFIQ AFIFI BIN FAUZI

BACHELOR OF MECHATRONIC ENGINEERING

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SUPERVISOR ENDORSEMENT

"I hereby declare that I have read through this report entitled 'Marker-based tracking under extreme illumination: feasibility study and pilot testing' and found that it complies the partial fulfilment for awarding the Degree of Bachelor of Mechatronic Engineering."

Signature	:	……………………………….
Supervisor's Name	:	Pn. Nurdiana binti Nordin
Date	:	21 MEI 2018

MARKER-BASED TRACKING UNDER EXTREME ILLUMINATION: FEASIBILITY STUDY AND PILOT TESTING

MUHAMAD AFIQ AFIFI BIN FAUZI

A report submitted in partial fulfilment of the requirements for the Degree of Mechatronics Engineering

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2018

STUDENT DECLARATION

I declare that this report entitles "Marker-based tracking under extreme illumination: feasibility study and pilot testing" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……………………………………

Name : MUHAMAD AFIQ AFIFI BIN FAUZI

Date : 21 MEI 2018

DEDICATION

To my beloved father and mother

Acknowledgement

Praise to Allah because give me strength, health, and patience during my Final Year Project. During this semester. I would like to take this opportunity to express my appreciation and thanks to those who are helping and contributed while conducting this project.

I wish to acknowledge certain individuals for providing me an opportunity and guiding me to do my Final Year Project for this semester. I would like to thank my parents with sincere gratitude for their never-ending support. Other than that, my sincere thanks to Pn Diana Nordin, my supervisor for helping and supporting me unconditionally to complete this project. Besides, I sincerely thank her for all the valuable experiences, knowledge and skills those shared with me.

In addition, special thanks to my both panel which are, Mr. Syed Mohamad Shazali Bin Syed Abdul Hamid which is my panel 1 and Dr. Mariam Md Ghazaly, my panel 2 who rendered their help and advice in improvising my overall project during this semester. My project going smoothly and I really enjoy in gaining experience during this project.

Last but not least, I also would like to thank my friends for the moral support, encouragement and all the help directly or indirectly to complete this project successfully.

ABSTRACT

The tracking system becomes a necessary need in the modern illumination system in the surgical room. This modern illumination system can replace the shortcomings of manually or old illumination system. This shortcomings including having to adjust surgical lamp manually and the presence of shadow due to obstacle that influence the amount of light reaching to the surgical area. To make this tracking system become efficient, a correct characterization of marker need to be identified. For this project, the characteristic of marker were proposed based on color and shape. This project was proposed to overcome two major problem which were the presence of shadow that inhibits transmission light and the changes of properties on the objects due to high illumination. So this project has focused on a few goal. Firstly was to analyze suitable object for object recognition. Both color and shape characteristic were used to recognize the marker. Black balls of different sizes were chosen due to properties of shape and color which were black and circular. Next was to locate the marker's position and to evaluate the precision of marker's position localization. Three different experiments were conducted by varying the brightness of light, size of object and the position in camera space in order to evaluate the tracking capability. Raspberry Pi with Pi camera were chosen to realize these experiments. Algorithms were written in Python with help from OpenCV libraries. While surgical lights can reach 45000 lux, these experiments were conducted at 5000 lux at maximum due to limited illumination capacity. Therefore the results may only be applicable to lower specification of surgical light such as illumination for dental surgery. It was found that marker with a diameter of 9.3 cm was consistently tracked within all quadrants in the experiment and at distance 100 cm, the error produce in marker detection was the least. The bigger the size was the better for the object to be detected in camera space and object was less susceptible to error when viewed from a longer distance since the apparent circle was smaller and the discrepancy of the approximation of the circle outer edge was minimal.

ABSTRAK

Sistem pengesan menjadi suatu keperluan dalam sistem pencahayaan moden di dalam bilik bedah. Sistem pencahayaan moden akan menambah baik sistem pencahayaan lama dan manual. Kekurangan ini termasuklah perlu menggerakkan lampu pembedahan secara manual dan kehadiran bayang oleh halangan yang mempengaruhi jumlah cahaya yang akan sampai ke kawasan pembedahan. Untuk membuat sistem pengesan ini beroperasi secara efisyen, ciri-ciri objek yang dikesan hendaklah ditentukan. Dalam projek ini, ciri-ciri yang dicadangkan ialah berdasarkan warna dan bentuk. Projek ini dicadangkan untuk mengatasi dua masalah besar iaitu kehadiran bayang yang menghalang cahaya dan perubahan sifat pada objek akibat pencahayaan yang tinggi. Oleh itu, projek ini telah memberi tumpuan kepada beberapa matlamat. Pertama ialah menganalisis objek yang sesuai untuk proses mengesan objek. Ciri yang digunakan ialah warna dan bentuk. Bola hitam dengan saiz yang berbeza telah dipilih kerana sifat bentuk yang bulat dan warna yang hitam. Seterusnya ialah mengesan kedudukan penanda and mengukur ketepatan untuk kedudukan objek. Beberapa kajian berdasarkan kajian kes telah dijalankan. Tiga eksperimen yang berbeza dilakukan dengan mengubah kecerahan cahaya, saiz objek dan kedudukan objek dalam ruang kamera untuk menilai kemampuan pengesanan. Raspberry Pi dan kamera Pi telah dipilih untuk eksperimen ini. Algoritma ditulis dalam Python dengan bantuan pustaka OpenCV. Walaupun lampu pembedahan bileh mencapai 45000 lux, tetapi eksperimen ini hanya dilakukan pada 5000 lux maksimum kerana terhadnya kapasiti pencahayaan. Oleh itu, hasilnya hanya dapat diterapkan pada spesifikasi yang lebih rendah dari cahaya pembedahan sebagai contoh pencahayaan untuk pergigian. Penanda dengan diameter 9.3 cm telah dikesan secara konsisten dalam semua kuadran dan pada jarak 100 cm, kesilapan yang dihasilkan dalam mengesan penanda adalah yang paling sedikit. Semakin besar saiz penanda adalah lebih baik untuk objek dikesan di ruang kamera dan kesilapan akan berkurang apabila objek dilihat dari jarak yang

jauh kerana bulatan lebih kecil dan pencanggahan penghampiran lingkaran luar bulatan adalah minimum

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURE

LIST OF APPENDICES

LIST ABBREVIATIONS

RGB	-	Red, Green, Blue
UV plane	-	Aperture Plane
HSV	-	Hue, Saturation, Value
HSI	-	Hue, Saturation, Intensity
ACD	-	Automatic Color Detection
GP	-	Ground Plane
ODM	-	Object Detection Mechanism
SBC	-	Single Board Computer
LAN	-	Local Area Network
RAM	-	Random Access Memory
VGA	-	Video Graphics Array
LCD	-	Liquid Crystal Display
DSP	-	Digital Signal Processor
CSI	-	Camera Serial Interface
FOV	-	Field of View
SD	-	Standard Deviation
RMSE	-	Root-Mean-Square Error
HDMI	-	High-Definition Multimedia Interface
OS	-	Operating System

CV	-	Computer Vision
CMYK	-	Cyan, Magenta, Yellow, Key
I/O	-	Input Output
JPEG	-	Joint Photographic Experts Group
PNG	-	Portable Network Graphics
GUI	-	Graphical User Interface
GTK	-	Gimp Tool Kit

CHAPTER 1

INTRODUCTION

1.1 RESEARCH BACKGROUND

One of the main components in the operating room is surgical light. Surgical light is a medical device intended to assist medical personnel during a surgical procedure by illuminating a local area or cavity of the patient. A combination of several surgical lights is often referred to as a "surgical light system".

There are several problems come from the surgical light systems. Even the surgical light was designed to eliminate shadow problem, but it still cannot remove the shadow completely. The surgeons have to adjust this surgical light manually in order to get the maximum amount of light intensity during a surgical operation. This will disturb the surgeon's concentration during operation process and the mistake will occur.

Recently, in order to counter this problem, a new illuminating system that has several functions such as many degrees of freedom, controlling the intensity and with auto tracking function is developed. Many auto tracking systems such as vision systems with the stereo camera are developed. These tracking systems are used to adapt automatically to the given light condition.

In order to improve existing technology, this study will discover and analyze suitable markers for localization. This project has been supplemented with other technology to make sure the outcome of this project are quality and usability in the medical field. The project was developed and discovered in this report by taking consideration of several methods, hardware, and software after meticulous research

1.2 MOTIVATION

The good illuminating system is very important to make sure there is no shadow which inhibits the transmission of light that can affect the amount of light reach at the surgical area and to make sure surgical operation run smoothly. A new illuminating system that has several functions such as many degrees of freedom, controlling the intensity and with auto tracking function is developed. The markers used in the surgical room must be taken into a count. The suitable object needs to be used for the tracking system due to limitedness of things in the surgical room. The high illumination in the surgical room will cause changes to the properties of an object that have been designated as markers and to counter it, the object selected can be used without any changes occur to the characteristics that cause by light condition. Therefore, this project was designed to improvise the existing illuminating system with auto tracking system by considering the shape and color as the markers characteristics. This technology will be great to be implemented in the surgical room because it does not require manpower to manually adjust the direction of the surgical illumination system and the surgeon can give their full concentration to their surgical operation. Based on the issue stated, the motivation has been found to create this project in order to counter the problem.

1.3 PROBLEM STATEMENT

The existing technology plays a big role in order to make sure the successes of the surgical operation in the surgical room. However, that technology can be improved to overcome several problems. The problem that usually occurs in the surgical room is the presence of the shadow due to the obstacle. This will inhibit the transmission of light that can affect the amount of light reach the surgical area. This will disturb the smoothness of surgical operation. The limitedness of things in the surgical room will be a problem for the selection of the object to be assigned as markers. High illumination in the surgical room also will cause changes of properties on the objects that have been designated as markers. The suitable object needs to be chosen as markers so that the high illumination will not give a great change to the presence of the object. It is important for the automatic tracking system.

1.4 OBJECTIVES

The objectives are listed below:

- To analyze suitable object for marker localization in surgical room
- To locate the marker’s position in camera space
- To evaluate the precision and accuracy of marker’s localization

1.5 SCOPE OF PROJECT

A few limitations have been set up in this project.

1. Only spherical and black color object was analyzed for choosing a marker. The reason for choosing spherical object was because of highly symmetrical shape. It also has the largest area for a given length perimeter compared to the object with the same size. The reason for choosing black color was because high illumination does not give much effect to the intensity of color and the apparent color does not change.
2. The distance between the camera and object for the test was set between 50cm to 100cm to imply the real concept of the illumination system in the real surgical room.
3. The Raspberry Pi and Pi camera was used as a tool to track the marker. They were chosen for availability. The Raspberry Pi was used in this project since the price was much cheaper compare to another microprocessor. Since the Raspberry Pi was used, the Pi camera was selected as the camera for image processing, recording video and vision in this project.
4. Precision and accuracy was measured through the radius of object when recognise in the camera space at different distance and different condition of illumination.

1.6 EXPECTED PROJECT OUTCOMES

The outcome expected for this project was to develop a marker-based tracking under extreme illumination. A marker-based tracking system for this project was developed to improve the existing technology by making the tracking process more efficiently with the proposed characterize of the marker with the simpler technique considering the brightness of the surgical lamp.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter will discuss and review about all the topics in this project. Methods for object detection and all the hardware and software used that are related to this project will be briefly discussed. The method that can be used in this project are object shape recognition which we can also use surgeon's head, detection of surgical tool, detection based on color and detection based on both color and shape. The environment for this project also will be discussed in this chapter. This project will provide an explanation of the view of the overall project.

2.2 OBJECT SHAPE RECOGNITION

The shape of the object can be considered as one method in tracking an object using vision. In previous works, Shape recognition method was proposed based on circle, square and triangle [1].

The adequate threshold of the gray level will be extracted to differentiate the object from their background. Otsu's threshold was selected to selects a threshold automatically from the gray level histogram and is a nonparametric and unsupervised method. It provide automatic threshold selection for image segmentation [1]. Optimal threshold will be selected by utilizing only the zero and first-order cumulative moments of the grey level histogram to maximise the separability of the resultant classes in gray level. The binary image clearly shows the differences between object and background using this method. To eliminate noise that exists in the image, the image fills will be used. The image fills will fill the ‘holes’ in the binary image of the input image. The ‘holes’ region will be converted to neighboring value. In order to smooth the edges and reduced the noise to a minimum, median filtering will be used. Sobel mask is used to detect shape’s outer edges. Sobel operator is an operator used in image processing, particularly for edge detection algorithm [1]. The outer edge perimeter is obtained by counting white pixels at the edge of the shape. Thinning process is the operation of removing pixels which satisfy the pattern given because there will be an increase of pixel count if the arrangement of pixels is not in straight line. Thinning operator is the subtraction of input image with the sub generating operator. The last step is shape recognition. The compactness calculation will be applied to the input image in RGB color space so that the output image will be produced. This compactness calculation is applicable to all geometric shapes but independent to a scale and orientation. The formula is shown in equation (2.1).

Compactness equation: $$c = \frac{p^2}{A} \qquad (2.1)$$

where c is compactness

p is a perimeter

A is area

c value is dimensionless.

The compactness value is fixed. The circle will have a value from 1 to 14, the square is in range 15 to 19 and triangle is from 20 to 40.

This method is sensitive to noise and lighting condition. Change of illumination intensity is very abrupt in the highly illuminated area especially in the surgical room, The poor lighting condition will give complexity in Otsu’s threshold algorithm.

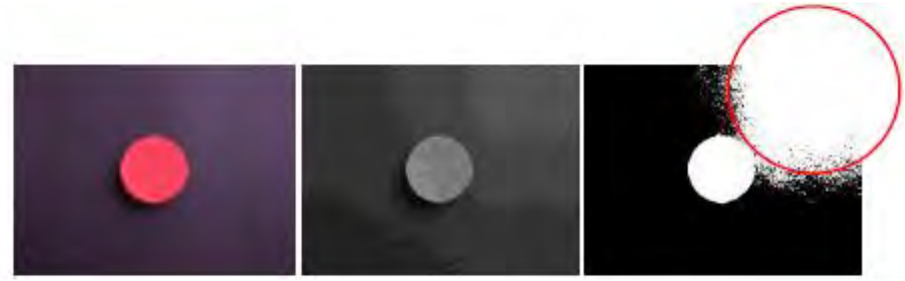


Figure 2.1: Otsu’s threshold output using gray scale image [1]

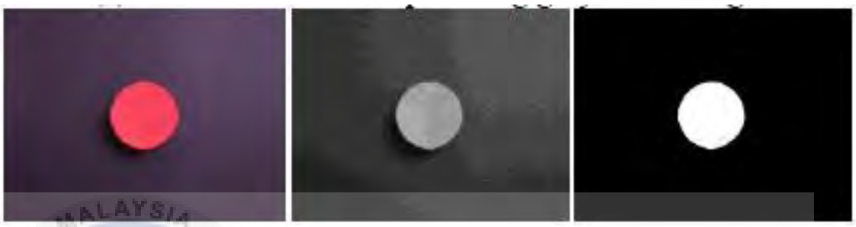


Figure 2.1: Otsu’s threshold output using lightness channel [1]

2.2.1 DETECTION OF SURGEON’S HEAD

As we know, the human head is a part of human body that includes the eyes, ears, nose, and mouth with aid in various sensory functions such as sight, hearing, smell, and taste. It is known that the gaze is a product of two contributing factors the head pose and the eye locations [2]. So detection of surgeon's head also can be considered as it related to the sight of the surgeons since we want to maximize the visualization of the surgeon during the operation. Accuracy in head pose and eye location estimation can be achieved in constrained settings [2]. The position of the head can enhance the accuracy of eye localization.

Cubistic Representation was introduced in [3]. Cubistic representation is a set of 3D surface fragments [3] and each fragment ***(f)***, contain the information on the subject’s appearance and redundantly covers the head surface**.** The shape acquisition is a sampling task that gathers a set of N_f fragments $\{f_k;\ k = 1, \ldots, N_f\}$ covering the target object surface ***S*** in this cubistic representation. Each fragment ***f*** represented by the following database coordinates, color texture, heightmap texture, and confidence mask texture. Uv plane in the world is defined as the base coordinates where it is used as a 2D texture's coordinates where the heightmap texture used to store 3D shape information. Color texture is used to store color information of the head surface [3]. The advantage of cubistic representation is this process does not add any quantization errors and distortions. It also can avoid any undesirable artifacts and approximations. The confidence mask used is in 2D texture on the UV plane as it used to store the results of pixel-level segmentation. The value is between 0 to 1 that indicates the relevance of ***εP.*** A higher value indicates less error to the corresponding color and heightmap pixel. The advantage of this approach is the capability of error detection and correlation using 3D geometric computations.

2.3 DETECTION OF SURGICAL TOOL

Detection of surgical tool may be quite suitable as we know all the surgeon will use the surgical instrument in helping them conduct their surgical operation. The surgeon vision may be focussed at the place where they conduct their surgical tool. In the previous related work, they take the surgical tool tips as their tracking object with the aid of computer vision.

Tool tracking has been performed using geometric constraints in order to identify the tool shaft [4]. Hough transform was used to identify the lines bordering tool shaft. The tooltip then can be found using color thresholding or physical measurements along the tool shaft.

A novel approach was used to identify lines in the image after the geometric constraints of the tool shaft exploited by Hough transform. The edges will be selected using dynamic threshold. The gradient threshold value depends on the resolution. If the number of edge pixels is too high, then the gradient threshold value also increased and same when the condition of edge pixels is low. The line endpoint information is sorted according to certain known geometric constraints in order to isolate the lines corresponding to the borders of the tool shaft [4]. When the correct lines been isolated, end of tool shaft contains tool wrist can be utilized from the endpoints of the lines. The endpoints are taken to be wrist and the midpoint of these endpoints taken as pixel location of the wrist. Correlation between pixel space coordinates of the tooltip and real-world coordinates will be the final step in tracking algorithm. All the step was shown in figure 2.3.

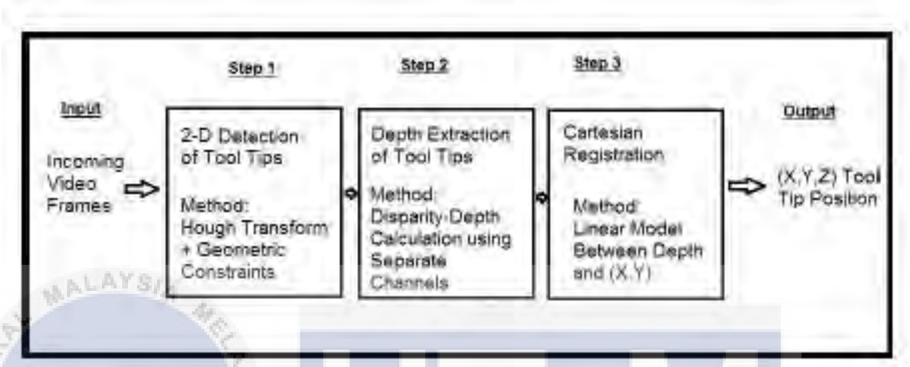

Input
Incoming Video Frames
Step 1
2-D Detection of Tool Tips
Method: Hough Transform + Geometric Constraints
Step 2
Depth Extraction of Tool Tips
Method: Disparity Depth Calculation using Separate Channels
Step 3
Cartesian Registration
Method: Linear Model Between Depth and (X,Y)
Output
(X,Y,Z) Tool Tip Position

Figure 2.2: The 3-step approach for object tracking [4]

2.4 DETECTION BASED ON THE COLOR

The limitedness of things in surgical room indirectly will reduce the variety color in our vision. Using the contra color can be considered as a method of object tracking. Then we can say that color is one of the factors that can be considered. Color-based object recognition offers an attractive option in the field of machine vision in which using color, objects which have the same morphological features will become distinctive [5]. A color image can be represented in a color space. Color spaces provide a representation of sensor outputs for easier computation as well as to accommodate the way human perceive colors [5]. RGB is one of the most suitable for multicolored object recognition when all imaging conditions are controlled [6].

There are 3 aspects in the red, green and blue color which is hue, saturation, and brightness. Color can be described and organized by this 3 aspects. Hue is known as the specific tone of color. Saturation refers to the purity, or intensity of a color. It is the intensity of a hue from grey. At maximum saturation, a color would contain no

grey at all. At minimum saturation, a color would contain mostly grey. Brightness refers to how much white, or black, is contained within a color. Chromaticity is the mixed of hue and saturation. Color can be categorized by brightness and chromaticity. The total value of the 3 color which is red, blue and green can be calculated by using the equation in the (2.3), (2.4) and (2.5).

RGB is normalized color as RGB which is red, green, and blue. I is the intensity and the equation is shown in (2.2), H is hue and S is saturation [6];

$$I(R,G,B) = R + G + B \tag{2.2}$$

$$r(R,G,B) = \frac{R}{R+G+B} \tag{2.3}$$

$$g(R.G,B) = \frac{G}{R+G+B} \tag{2.4}$$

$$\mathrm{b(R,G,B)} = \frac{B}{R+G+B} \tag{2.5}$$

The summation of R, G and B will be equal to 1.

The transformation of image can describe the hue, H. The equation is shown in (2.6)

$$H(R,G,B) = \arctan(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}) \tag{2.6}$$

And saturation, S can be measured after having particular hue. The equation of S is shown in (2.7)

$$S(R,G,B) = 1 - \frac{\min(R,G,B)}{R+G+B} \tag{2.7}$$

All color features can be estimate from the original R, G, and B values.

Hue (H) component of every pixel can also be calculated with the theory and calculation of the conversion color from RGB to HSI when considering the image in RGB color format:

$$H = \begin{cases} \theta & if\ B \leq G \\ 360^\circ - \theta & if\ B > G \end{cases} \tag{2.8}$$

Where

$$\theta = cos^{-1}\left\{\frac{0.5(R-G)+(R-B)}{(R-G)^2+(R-B)(G-B)^{0.5}}\right\} \tag{2.9}$$

Saturation components

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \tag{2.10}$$

Intensity components

$$I = \frac{1}{3}(R+G+B) \tag{2.11}$$

RGB values are in the range of [0,1]and the angle θ was measured with the respect of the red axis of HSI space.

Detection of color may cause a problem. These methods suffer from color and lighting variation issues [4]. This method very sensitive to light intensity and light reflection. It also cannot recognize shape.

All the techniques from the related journal are summarized in the table 1 below.

Table 2.1: Summary of related journal for the color, shape and texture

Type	Color [4] [5] [6] [7]	Shape[1] [2] [3] [7]	Texture(tool) [4]
Example of Technique use	• RGB • Color space	• Compactness	• Geometric constraints • Hough transform
Advantages	• Object can be identified easily • Color can be unique to an object • Less sensitive to noise • More largely robust to a view direction change and resolution	• Object can be identified • Does not depend on color	• Does not depend on color
Disadvantages	• Affected by another object with the same color as object targeted • Color is affected by illumination • Object color may not be distinctive	• Sensitive to noise and lighting • Take at least a few seconds to scan and detect	• Tool will be too small • Small area for given parameter • end of the tool shaft become occluded

2.5 DETERMINATION THAT USING BOTH COLOR AND SHAPE

In order to improve the existing technology, the use of color and shape should be taken for the characteristic of choosing a marker. The previous related work that uses this both characters will be studied in this section.

This both characteristic has been proposed in [7] and [8] for object detection. This both projects begin with color detection. In [7], the HSI (Hue, Saturation, Intensity) color space has been used while in [8], Automatic Color Detection (ACD) mechanism has been selected as a technique used to determine the color of object for the HSI. Usually only the H and I component is used. Sometimes only H component is used for the object color for the purpose of detection, recognition and etc. The conversion formula from RGB (Red, Green, Blue) components to the HSI components is shown below.

$$H\ (HUE) = cos^{-1}\left\{\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-G)(G-B)}}\right\} \tag{2.12}$$

$$R \neq G\ or\ R \neq B$$

$$If\ B > G\ then\ H = 2\pi - H$$

$$S(saturation) = 1 - \frac{3}{(R+G+B)}[\min(\mathrm{R}, \mathrm{G}, \mathrm{B})] \tag{2.13}$$

$$I = \frac{1}{3}(R + G + B) \tag{2.14}$$

In [8], Automatic Color Detection (ACD), it consists of sweeping the hue-saturation-value (HSV) color space in hue intervals and color detection mechanism. It only operates on the first image assuming the object is present in that image but can be executed later if required. For the sweep of the HSV color space step, the consecutive color filter is applied for every interval in the hue range [8]. Only the hue space is swept in the tonality in intervals of ten while the value and saturation intervals are constant so that different intensities for each color will be acquired. The GP (Ground Plane) color-filtered image will be taken for each hue interval and refilter it for the hue interval in consideration. The output image then consists only the pixels belonging to the hue interval. After that, the contours of this output image will be found. The contour that less than the certain predefined threshold value will be discarded as noise. The minimum enclosing circles for each remaining contour and the area of enclosing circles will be computed. Relative error between the area of enclosing circle and the respective internal contour also will be computed. The minimum relative error among all the contours corresponding to that hue interval is compared with the absolute minimum relative error from all the previous hue intervals (already processed), saving only the smallest [8]. After the first step is finished, the hue interval that has a contour with the least relative error is the one corresponding to the color of object [8].

Next step of this method is shaped detection. In [7], a simple and visualized method based on connected edge area and the circle's centers and radii have been proposed while in [8], Kalman Filter is used. There is 3 step of the technique proposed in [7]. The first step is to compute the image gradients using the Gaussian template. Next is the edge detection by using Canny edge detection and lastly is the boundary area, the circumscribed circle center and radius were compute for each connected edge boundary. Let S_m, R_m, n_m be the area, the circumscribed circle radius (if it has), and the number of the m$^{\text{th}}$ edge boundary.

For all edge points $pt\,(i\,)$ $(i \in [0, n_m - 1])$ in a boundary

$$S_m = \frac{1}{2}\sum_{i=1}^{n_m} x_{i-1}y_i - x_i y_{i-1} \qquad (2.15)$$

Randomly pick three edge points in a boundary

$$C_m = \{pt(1), pt(2), pt(3)\} \qquad (2.16)$$

Compute their circumscribed circle center (x_{m0}, y_{m0}) and radius R_m

$$x_{m0} = \frac{\det(\mathrm{A})}{4((x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1))} \qquad (2.17)$$

$$y_{m0} = \frac{\det(\mathrm{B})}{4((x_2-x_1)(y_3-y_1)-(x_3-x_1)(y_2-y_1))} \qquad (2.18)$$

Where det(A) and det(B) are the determinants of matrices A and B respectively

$$A = \begin{bmatrix} x_2^2 + y_2^2 - (x_1^2 + y_1^2) & 2(y_2 - y_1) \\ x_3^2 + y_3^2 - (x_1^2 + y_1^2) & 2(y_3 - y_1) \end{bmatrix} \tag{2.19}$$

$$B = \begin{bmatrix} 2(x_2 - x_1) & x_2^2 + y_2^2 - (x_1^2 + y_1^2) \\ 2(x_3 - x_1) & x_3^2 + y_3^2 - (x_1^2 + y_1^2) \end{bmatrix} \tag{2.20}$$

$$R_m = \sqrt{(x_{m0} - x_d)^2 + (y_{m0} - y_d)^2} \tag{2.21}$$

(x_d, y_d) is the coordinate of any three selected points

This equation $S_m = \pi R_m^2$ must be satisfied if m^{th} edge curve is circle.

S_m and R_m can be choose in order to detect object accurately because the shape might be change slightly due to changing of view position, noise, elimination changing and etc.

In [8], Kalman Filter technique were used. For the Kalman Filter, the first step to be taken is determines the motion of the robot and the object. The object velocity needs to be converted from the global frame to the robot frame. Let $v_{\mathrm{o}}^{\mathrm{r}}$ is the object's velocity in the robot's frame, v_0^{g} is the object's velocity in the global frame, v_{r} is the robot's linear velocitythat always in global frame and ω_{r} is the angular velocity of robot about an axis passing through its center of mass and perpendicular to GP (Ground Plane). Let $p_{\mathrm{o}}^{\mathrm{r}}$ is the position of object in robot frame.

Dot-format

$$\begin{cases} v_0^{\mathrm{r}} = v_{\mathrm{o}}^{\mathrm{g}} - v_{\mathrm{r}} - \omega_{\mathrm{r}} \times p_{\mathrm{o}}^{\mathrm{r}} \\ \dot{v}_{\mathrm{o}}^{\mathrm{r}} = \dot{v}_{\mathrm{o}}^{\mathrm{g}} - \dot{v}_{\mathrm{r}} - \dot{\omega}_{\mathrm{r}} \times p_{\mathrm{o}}^{\mathrm{r}} - \omega_{\mathrm{r}} \times v_{\mathrm{o}}^{\mathrm{r}} \end{cases} \tag{2.2}$$

Since the robot's and tracked object's acceleration are assumed to be zero, i.e,

$\dot{v}_{\mathrm{o}}^{\mathrm{r}} = \dot{v}_{\mathrm{r}} = \dot{\omega}_{\mathrm{r}} = 0.$

$$\dot{v}_{\mathrm{o}}^{\mathrm{r}} = -\omega_{\mathrm{r}} \times v_{\mathrm{o}}^{\mathrm{r}} \tag{2.23}$$

The robots add negatively to the object's velocity in the global frame so that the object's velocity in the robot frame will be obtained. When implying zero global velocity, a static object will be seen moving in the opposite direction of the robot's velocity direction when the object is viewed from the robot frame and the robot's angular velocity also affects negatively to the apparent object movement in the robot frame. The state to be estimated is denoted by x which consist 2D position and velocity of the object in the robot's frame of reference.

$$\mathrm{x}=[p_{\mathrm{o}}^{\mathrm{r}}\ v_{\mathrm{o}}^{\mathrm{r}}]^{\mathrm{T}} = \left[p_{\mathrm{o}x}^{\mathrm{r}} p_{\mathrm{o}y}^{\mathrm{r}} v_{\mathrm{o}x}^{\mathrm{r}} v_{\mathrm{o}y}^{\mathrm{r}}\right]^{\mathrm{T}} \tag{2.24}$$

The discrete time state transition model and the observation model

$$\mathrm{x}(t) = \Phi(\mathrm{t}) \times (t-1) + \Gamma(t)u(t) \tag{2.25}$$

$$\mathrm{z}(t) = H(t)\mathrm{x}(t) \tag{2.26}$$

where

$$\Phi(t) = \begin{bmatrix} \cos(\Delta\theta) & \sin(\Delta\theta) & \Delta t\cos(\Delta\theta) & \Delta t\sin(\Delta\theta) \\ -\sin(\Delta\theta) & \cos(\Delta\theta) & -\Delta t\sin(\Delta\theta) & \Delta t\cos(\Delta\theta) \\ 0 & 0 & \cos(\Delta\theta) & \sin(\Delta\theta) \\ 0 & 0 & -\sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{2.27}$$

$$\Gamma(t) = \begin{bmatrix} \frac{-\Delta t\sin(\Delta\theta)}{\Delta\theta} & \frac{\Delta t(\cos(\Delta\theta)-1)}{\Delta\theta} \\ \frac{\Delta t(\cos(\Delta\theta)-1)}{\Delta\theta} & \frac{-\Delta t\sin(\Delta\theta)}{\Delta\theta} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}` \tag{2.28}$$

$$H(t) = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \tag{2.29}$$

(t) is associated with any variable denotes its value at the timestep t.

Δt denotes the time interval between timestep t and t-1, while $\Delta\theta$ is the angular displacement and $u(t) = [u_x(t)u_y(t)]^{\mathrm{T}}$ is the linear displacement of robot between those timestep obtained from robot's odometry measurement. z(t) is the object's observation measurements obtained at the t^{th} timestep from ODM (Object Detection Mechanism). ODM is used in Kalman Filter update step as the measurement source for object's position. After the new odometry readings are obtained, the prediction step is performed whereas the update is performed when the measurements are obtained from the ODM.

- **Prediction Step**

$$\bar{x}(t) = \Phi(t)x(t-1) + \Gamma(t)u(t) \tag{2.30}$$

$$\bar{\mathbf{P}}(t) = \Phi(t)\mathbf{P}(t-1)\Phi(t)^{\mathrm{T}} + \mathbf{Q}(t) \tag{2.31}$$

- **Update Step**

$$\mathbf{K}(t) = \bar{\mathbf{P}}(t)\mathbf{H}(t)^{\mathrm{T}}(\mathbf{H}(t)\bar{\mathbf{P}}(t)\mathbf{H}(t)^{\mathrm{T}} + \mathbf{R}(t))^{-1} \tag{2.32}$$

$$\mathbf{x}(\boldsymbol{t}) = \bar{\mathbf{x}}(\boldsymbol{t}) + \mathbf{K}(t)(\mathbf{z}(\boldsymbol{t}) - \mathbf{H}(t)\bar{x}(t)) \tag{2.33}$$

$$\mathbf{P}(\boldsymbol{t}) = (\mathbf{I} - \mathbf{K}(t)\mathbf{H}(t))\Phi(t)\bar{\mathbf{P}}(t) \tag{2.34}$$

$\bar{\mathbf{P}}(t)$ and $\mathbf{P}(t)$ denotes a piori and posteriori error covariance matrices and both were identity matrices. **Q** is the process noise covariance matrix and **R** is the measurement noise covariance matrix. Both are based on ODM's measurementand odometry errors. **K** is the Kalman gain while **I** is an identity matrix.

By focusing on both color and contour based detection (shape), it will be more precise in the decision of target object and its localization. The two basic object features which are color and contour information(shape), should be taken for a job at hand [7] in the case of algorithm simplifier and reducing time consumption in order to be suitable with a realistic environment.

2.6 SURGICAL ENVIRONMENT

The environment in the surgical room needs to be considered as a factor for choosing marker color. Surgical rooms are usually dominated by white/neutral colors such as wall, bed sheets and surgical gloves, green colors such as surgical vest, patient covers, and tissues, red color such as bloodline, grey color such as surgical instruments and numerous skin color usually dominated from pearl white to brown spectrum [5]. To make the tracking process become more effective, the markers colors that are chosen are not in the color groups mentioned before.

The amount of illumination also needs to be considered to ensure safe and accurate working during a surgical operation. The term of illuminance which is the amount of visible light that strikes a surface divided by the area of that surface is quantified for the amount of the illumination needed [5]. The units use are lux which is one lumen per square meter. Table 2.2 show the example of lightning condition with their typical illuminance.

Table 2.2: Example of lighting condition with their typical illuminance [5].

Lighting condition	Typical illuminance
Threshold of seeing	< 1 lux (<0.1 footcandles)
Living room	100 lux (9 footcandles)
Professional office	500 lux (46 footcandles)
Exam room (emergency department)	1,000 lux (90 footcandles)
Operating room – general lighting	1,500 lux (140 footcandles)
Noonday sun	100,000 lux (9,300 footcandles)
Surgical task lighting	150,000 lux (14,000 footcandles)

General lighting in the surgical room is the overhead lighting system. The range of the illuminance is recommended to be in between 1000-2000 lux which can reduce eye-strain that caused by glancing from the bright area.

In the surgical site, the higher illumination is needed. It is needed because of the presence of partial obstructions of the beam that can reduce the illuminance of the surgical site. The standard range requires at least between 40,000-160,000 lux. This extreme illumination will produce sharp transition of light when interrupted and indirectly reduce the consistency of marker colors.

2.7 HARDWARE AND SOFTWARE REQUIREMENT ANALYSIS

This chapter will discuss the knowledge about the hardware and software that will be used. In this project, the hardware needed is the microprocessor that will be used in image processing and object detection. The software needed for this project is the programming language and library that will be run in the program installed in the microprocessor. There are many types of a microprocessor that can be used in image processing and object detection. Below is the list of the famous types of processor that are used in image processing and object detection.

2.7.1 RASPBERRY PI

Raspberry Pi® is an ARM-based credit card sized SBC (Single Board Computer) created by Raspberry Pi Foundation. Raspberry Pi runs Debian based GNU/Linux operating system Raspbian and ports of many other OSes exist for this SBC. This Raspberry Pi comes with wireless LAN and Bluetooth connectivity. This Raspberry Pi function as image analyzer, recognizing faces and text and looking for items of interest.

As we know, there are several generations of Raspberry Pi's have been released. Start with the first generation released in February 2012 which is Raspberry Pi 1 Model B. Then the simpler and cheaper Model A was released. The foundation then released a board with an improved design in Raspberry Pi 1 Model B+ in 2014. An improved model was then released a year after that which is A+ and B+. These boards are designed for credit-card sized and represent the standard mainline form-factor. A "Compute Module" with embedded applications was released in April 2014. The model with more RAM which is Raspberry Pi 2 was released in February 2015.

A Raspberry Pi with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities which are Raspberry Pi Zero was released in November 2015. In February 2016, Raspberry Pi 3 Model B was released. The Raspberry Pi is bundled with on-board Wi-Fi, Bluetooth and USB boot capabilities. As of January 2017, the newest mainline Raspberry Pi that is Raspberry Pi 3 Model B is released. The Raspberry Pi Zero W was launched On 28 February 2017, which is identical to the Raspberry Pi Zero but has the Wi-Fi and Bluetooth functionality of the Raspberry Pi 3.

2.7.2 BEAGLE BONE BLACK

The Beagle Bone Blackboard are tiny open-hardware and open software computers that can plug into whatever you have around yourself. Beagles are small packages with the big functionality of PCs that can be used for all kinds of applications. It can handle many tasks same as your desktop PC. In learning path, it helps students learn to programme and help the programmer to push boundaries of DIY and to move production quickly without excess noise, expense or bulk.

This board ships with the Debian GNU/Linux™ in onboard FLASH to start evaluation and development. These Linux distributions and operating systems are also supported on BeagleBone Black including Ubuntu, Android and also Fedora.BeagleBone Black's capabilities can be extended using plug-in boards called “capes”. Capes are available for, VGA, LCD, motor control, prototyping, battery power and other functionality that can be plugged into BeagleBone Black's two 46-pin dual-row expansion headers. The price is quite expensive compared to the raspberry pi.

2.7.3 DIGITAL SIGNAL PROCESSOR (DSP)

Digital Signal Processors are microprocessors specifically designed to handle Digital Signal Processing tasks. It is a particular chip which is improved the quick operational needs of advanced digital signal processing. Digital signal processor (DSP) collects, collates, processes, and filters the digitized data set.

The processor can acquire real-time data at rates as high as 150Hz from managing system timing and the imaging routines. In order to orchestrate data flow, DSP also used a complex programmable logic device which can control synchronously timed processes. The operation of the system which is integrated automated calibration, data acquisition, data organization, and signal post processing is implemented through a comprehensive graphical user interface designed with LABVIEW software. The price is quite expensive compared to both processors above.

2.7.4 RASPBERRY PI CAMERA MODULE

The Raspberry Pi Camera Module is an official product of the Raspberry Pi Foundation. The original 5-megapixel model was released in 2013, and an 8-megapixel Camera Module v2 was released in 2016. For both iterations, there are visible light and infrared versions. The function of this camera is to take pictures and record quality videos. This Pi camera can be controlled programmatically.

2.7.5 PYTHON (PROGRAMMING LANGUAGE)

Python is one programming language which is a general-purpose language. It has a wide range of applications from Web development (like Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. Python is a general-purpose programming language that became very popular quickly, mainly because of its simplicity and code readability. The programmer can express ideas in fewer lines of code without reducing readability. Python is slower compared to languages like C/C++. Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

2.7.6 OPEN CV AND SIMPLE CV

OpenCV-Python is a learning software library designed to solve computer vision problems. There are two advantages using this library. Firstly, the code is as fast as the original C/C++ code since it is the actual C++ code working in the background. Second, it is easier to code in Python than C/C++. The library has more than 2500 optimized algorithm. These algorithms are used to detect and recognize faces, identify the object, classify human actions in videos, track camera movements, track moving objects follow eye movement, etc. It supports Windows, Linux, Android, and Mac OS with C++, C, Python, Java and MATLAB interfaces

2.8 SUMMARY OF THIS CHAPTER

The method of using both color and shape was chosen in detecting marker in this project. This method was more precise in the decision of target object and its localization. It also will reduce time consumption in the case of algorithm simplifier. The color selected was black as the high illumination does not give much effect to the intensity of color and the apparent color does not change. The shape that was used is spherical or circle because this shape was highly symmetrical shape which can make tracking process become less complex and easier. It also has the largest area for a given length of parameter compare to another shape with the same size. Raspberry Pi was chosen for the processor instead of another processor because it was user-friendly and the cost was the cheapest compared to another processor. For the vision system, the Pi camera was selected since the Raspberry Pi board was chosen. The Python language was used for the programming language due to simplicity and code readability. Since the Python language was chosen, the Open CV was selected as the library to be used. It was easier to code in Python and this code as fast as the original C/C++ code. This library also suitable for tracking an object.

Hardware and software selected

Based on the microprocessor listed above, Raspberry Pi was selected as the hardware to be used in this project. Raspberry Pi was chosen instead of another processor because the understanding of using and applying this microprocessor was better than the others. It also user-friendly and regular people can use it in a wide variety of task. The price of this microprocessor was the cheapest among the others and has a huge Documentation online. This will be perfect for a project that needs a computer but does not require much processing power, save on space, robust and low cost. It was perfect for this project.

This Raspberry Pi has many models since there were several generations has been released. There were two levels of the naming system. The number indicates the generation of the model which were Pi 1, Pi 2 and Pi3. 3 was better than 2, and 2 was better than 1. The alphabet indicates the power and the features. There were model A, A+, B and B+ which A was lower than B. This different model has a different price. All of this factor needs to be considered in choosing the correct model to be used. So for this project, Raspberry Pi 3 Model B was selected. This model was chosen to meet the requirement needed so that there is no problem related to the microprocessor.

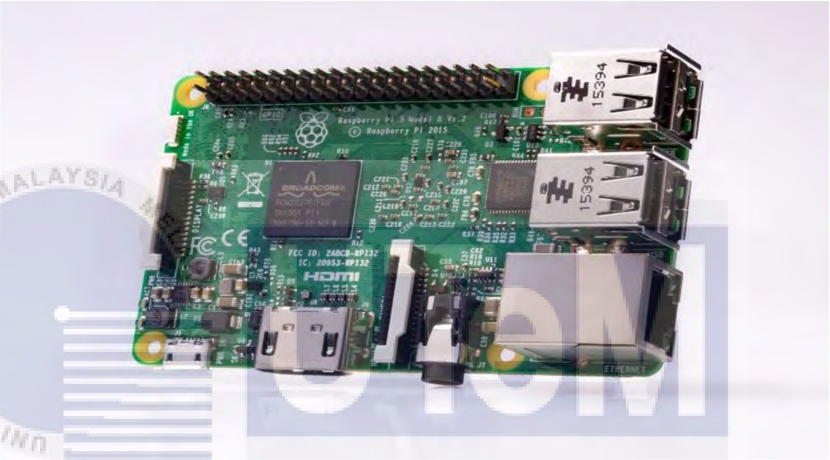


Figure 2.3: Front view of Raspberry Pi 3 Model B board

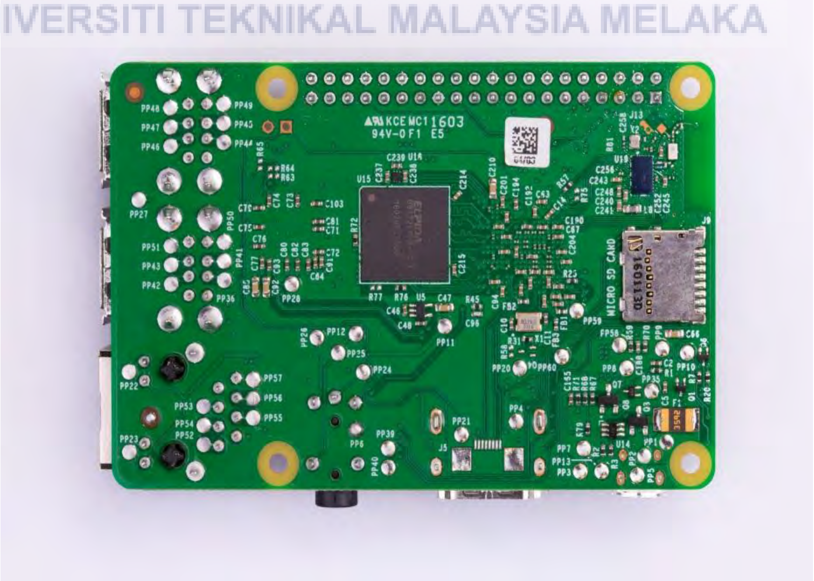


Figure 2.4: Back view of Raspberry Pi 3 Model B board

The Raspberry Pi 3 is the third-generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Below is the specification of this model:

- Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full-size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

Raspberry Pi camera board was chosen for image processing, recording video and vision in this project. It was able to deliver a crystal clear 5MP resolution image or 1080p HD video recording at 30fps with the latest version of 1.3. It also supports video recording 720p at 60fps and 640×480p 60/90. This camera board has been chosen because it was compatible with all models of raspberry pi 1, 2 and 3. This Raspberry Pi camera board plugs directly into the CSI connector on the Raspberry Pi. This board features a 5MP (2592×1944 pixels) Omnivision 5647 sensor in a fixed focus module. The size of this camera board was 20mm×25mm×9mm and has the weight of 3g. It attaches to raspberry pi by way of 15 pin ribbon cable, dedicated 15 pin MIPI Camera Serial Interface (CSI), which was designed especially for cameras interfacing. The vertical field of view (FOV) of the Pi camera is $41.41° \pm 0.11°$. The horizontal field of view (FOV) of the pi camera is $53.50° \pm 0.13°$. The CSI bus is capable to transmit high data rates and exclusively carries pixel data to the BCM2835 processor.

Figure 2.5: Raspberry Pi camera board

There were many operating systems that can be installed to run on the Raspberry Pi. Among them, there are Raspbian OS, Retro Pi, Windows 10 IoT Core, OSMC, and Pidora. All of this operating systems were the best operating system that suits for Raspberry Pi that was used in this project. The choice in choosing this operating system depends on what type of Pi project that you will do and the comfort of using it. So for this project, Raspbian OS was chosen as an operating system. This operating system suitable for image processing and object detection compares to another operating system. The system was quite similar to Linux. It was a lightweight OS that based on Debian distribution and specially designed to work on Raspberry Pi devices such as Pi camera. There were many types of raspbian. Raspbian Jessie was used in this Raspberry Pi. The OpenCV and python were installed here as the dependencies.

Five light source zoom headlamp was selected as the light source. That headlamp was chosen as it can emit 35,000 lumens of brightness. This headlamp consist 5 bulb – 1 T6, 4XPE. It has 4 functions which is glare, careless, super bright and blasting flash. It needs 2 X 18650 batteries as the power supplies. The lux of the headlamp was measured by the lux meter

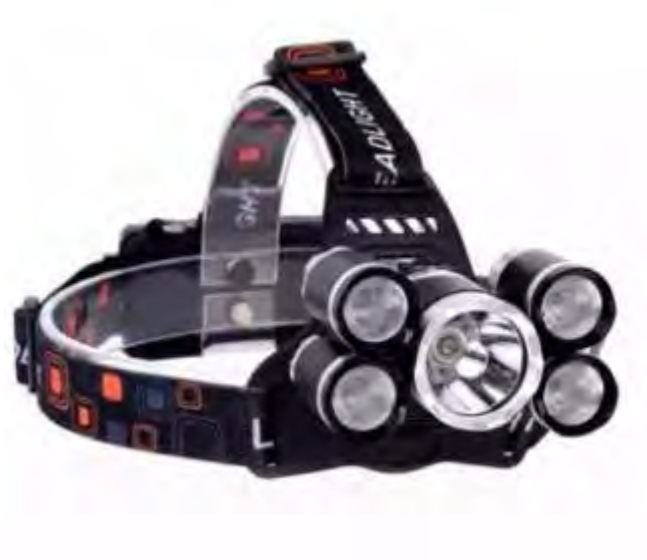


Figure 2.7: Five light source zoom headlamp

CHAPTER 3

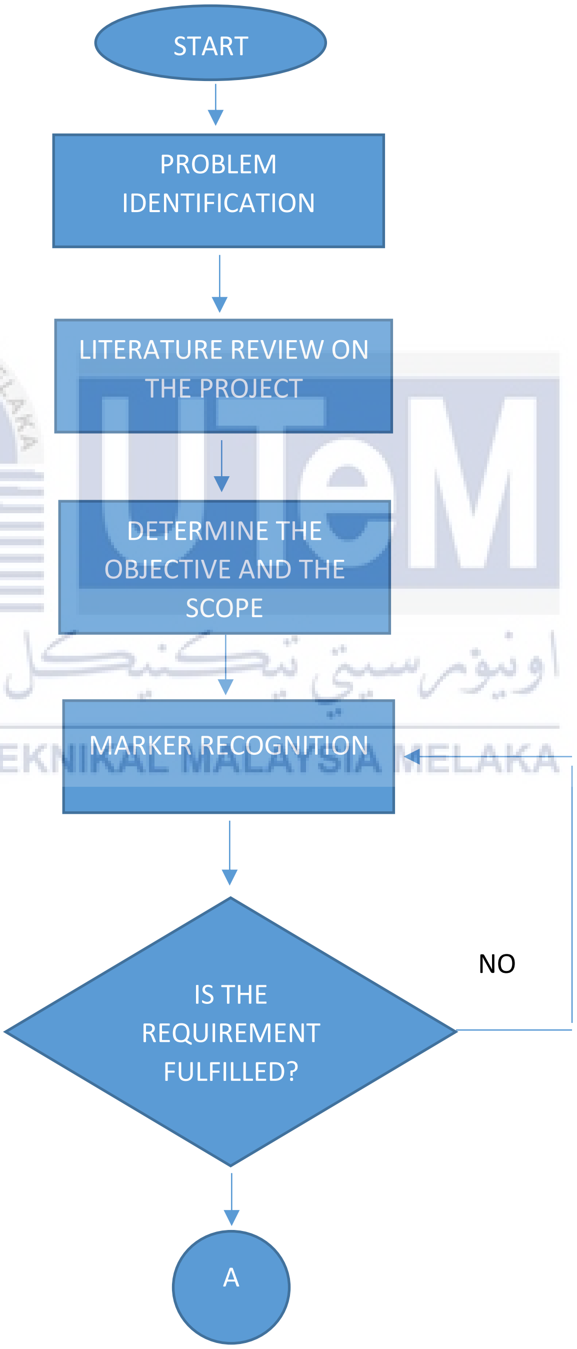
METHODOLOGY

3.1 INTRODUCTION

In this chapter, we will discuss the hardware and software used and how to implement it in this project. The technique used for this project also was discussed. A few setup of the experiment was proposed for the analysis that was done in this project. The flow of the project was explained in project flowchart. Gantt chart was also presented to display the project timeline as planned.

3.2 PROJECT FLOWCHART

The steps and methods included in this project was explained briefly in the form of a flowchart shown in figure 3.1. The overall planning flow chart show the process of the whole final year project panning.


START
PROBLEM IDENTIFICATION
LITERATURE REVIEW ON THE PROJECT
DETERMINE THE OBJECTIVE AND THE SCOPE
MARKER RECOGNITION
IS THE REQUIREMENT FULFILLED?
NO
A

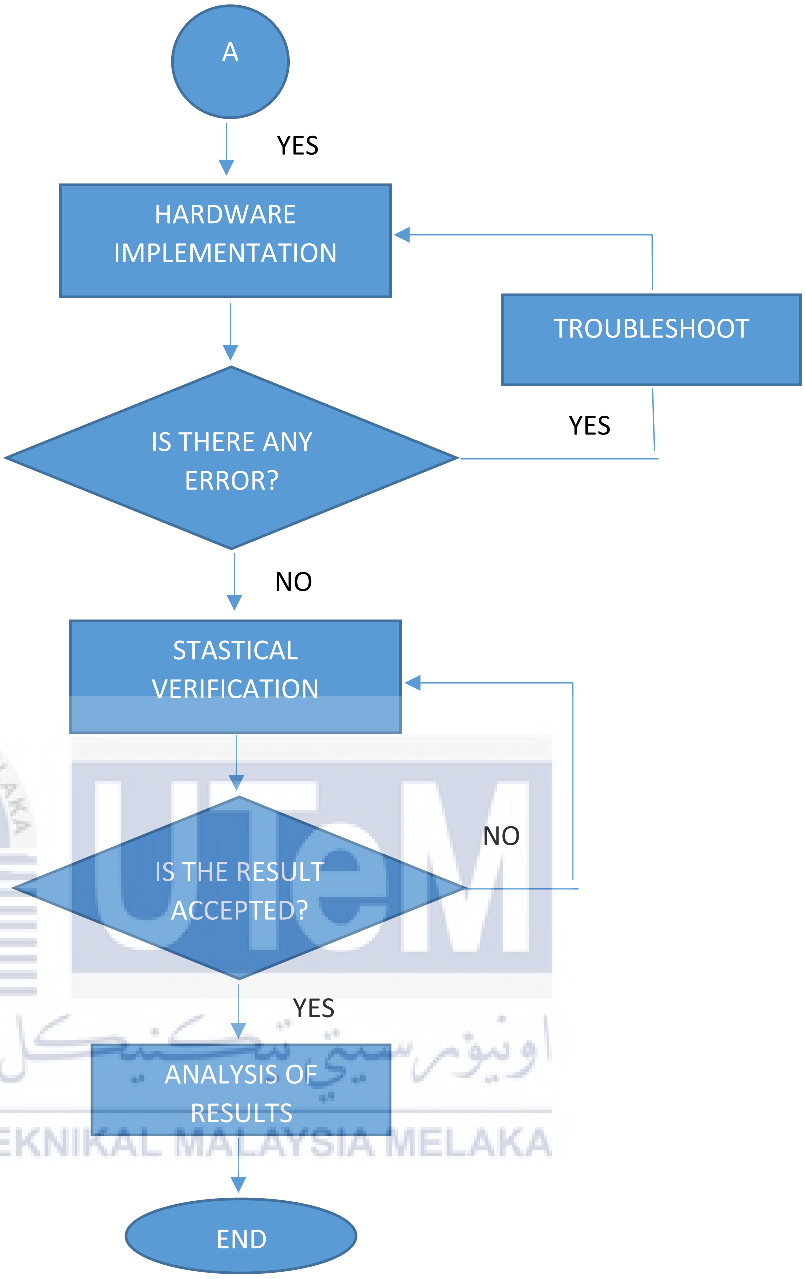


Figure 3.1: Flow chart of overall project.

Table 3.1: Gantt chart for PSM 1

FINAL YEAR PROJECT 1	Semester 1													
PROJECT ACTIVITIES	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MEET PROJECT SUPERVISOR	■	■	■	■	■	■	■	■	■	■	■	■	■	■
CONFIRMATION OF FYP TITLE	■	■												
FIND JOURNALS RELATED TO FYP TITLE AND LITERATURE REVIEW			■	■	■	■	■					■	■	
PROBLEM IDENTIFICATION			■	■	■									
STUDY THE CHARACTERISTIC OF MARKER TO BE USED					■	■	■	■						
BUILD CODING FOR MARKER TRACKING SYSTEM IN PYTHON						■	■	■	■	■	■	■	■	
DATA COLLECTION AND ANALYSIS (TESTING THE CODE)							■	■	■	■	■	■	■	
REPORT WRITING FYP 1							■	■	■	■	■	■	■	
PRESENTATION FYP 1											■			
SUBMISSION OF FYP REPORT 1														■

Table 3.2: Gantt chart for PSM 2

FINAL YEAR PROJECT 2	Semester 2													
PROJECT ACTIVITIES	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
MEET PROJECT SUPERVISOR	■	■	■	■	■	■	■	■	■	■	■	■	■	■
LITERATURE REVIEW	■	■												
DEVELOPMENT OF HARDWARE	■	■	■	■	■	■	■					■	■	
INSTALLATION OF HARDWARE			■	■	■									
TESTING THE HARDWARE BASED ON THE CASE STUDY				■	■	■	■	■	■	■				
DATA COLLECTION AND ANALYSIS				■	■	■	■	■	■	■	■	■		
IMPROVEMENT FOR THE TRACKING SYSTEM						■	■	■	■	■	■	■		
REPORT WRITING FYP 2			■	■	■	■	■	■	■	■	■	■	■	
PRESENTATION FYP 2														■
SUBMISSION OF FYP REPORT 2														■

3.3 OBJECT RECOGNITION

To achieve objective 1 which was to analyze suitable object for the marker's localization in the surgical room, the hardware connection and implementation for the tracking was proposed. For this part, the capture video of the circular moving black object from Pi camera has been used. This video was used to implies the understanding and to prove the algorithm testing to detect the marker's characteristic. To ensure an appropriate algorithm was implemented, multiple video with a dueration of 60 seconds were recorded prior to the algorithm testing. The target object was moved within the camera view to enable tracking.

The propose characteristic of the object that was used as the marker was based on color and shape. The color selected was black as the high illumination does not give much effect to the intensity of color and the apparent color does not change. Based on the analysis in HSV color space, the black color range can be obtained by adjusting the value (V) in the color space.

The shape used in this project was circular or sphere. Based on the analysis, circular shape was said to be less complex shape. It also has the largest area for a given parameter compared to another shape with the same size. This shape also was highly symmetrical object as the shape will still the same when being view on the monocular camera from any point of view.

- **Marker recognition**

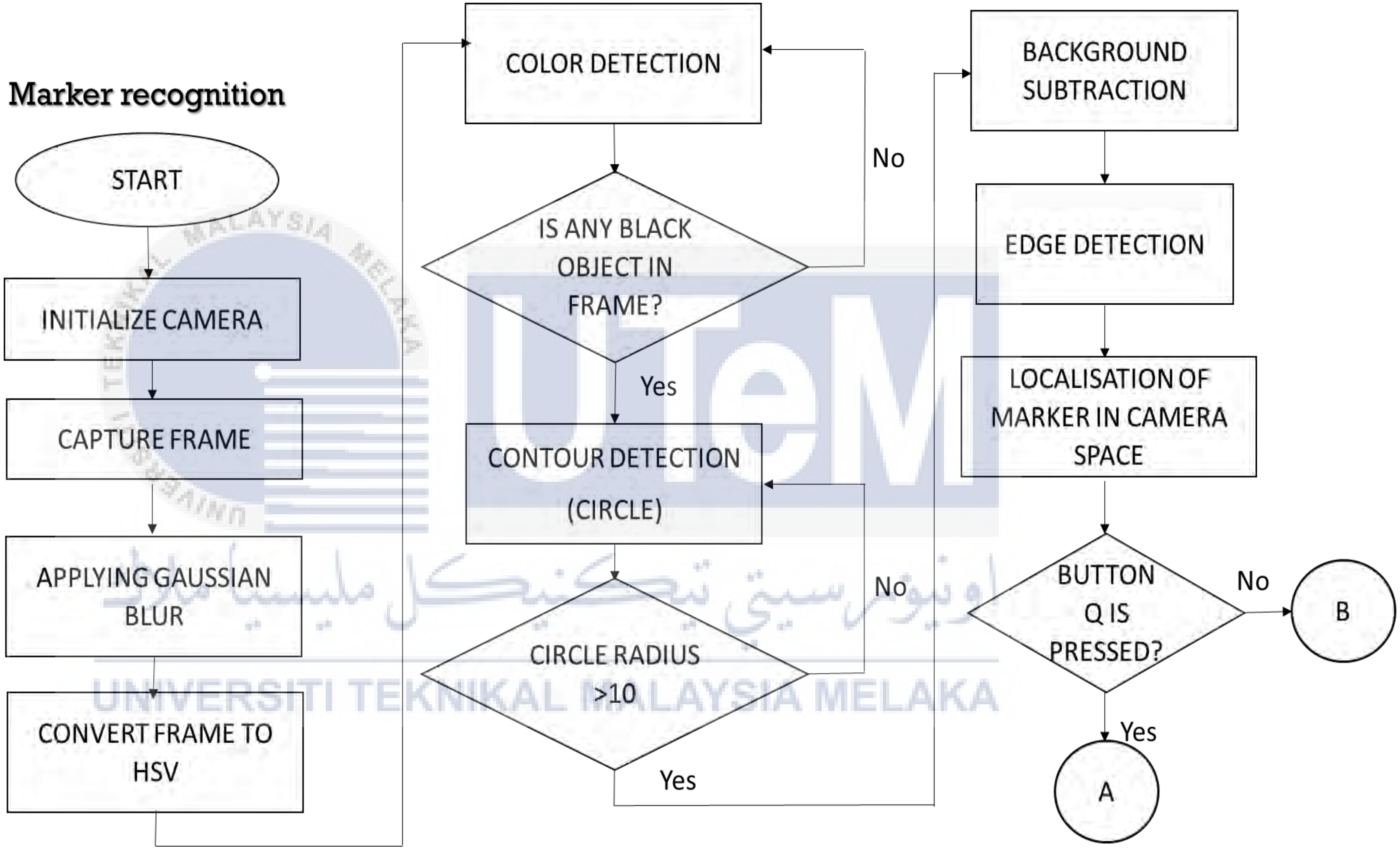
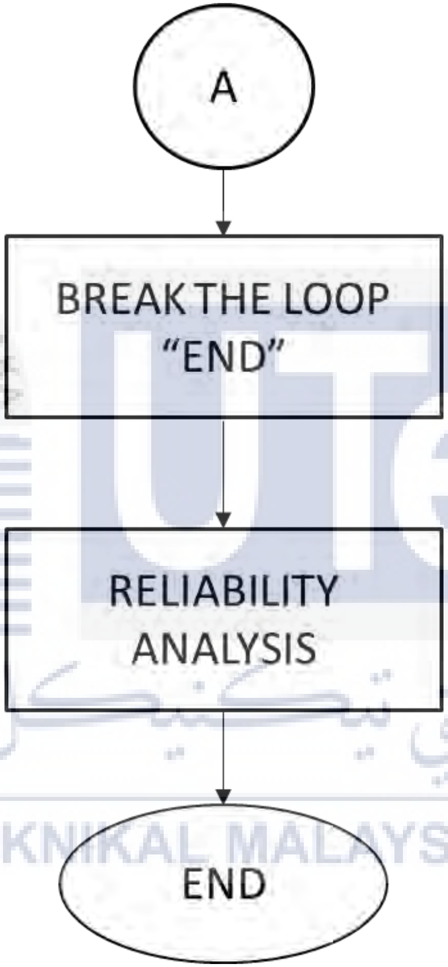


Figure 3.2 :Flow chart of marker recognition


A
BREAK THE LOOP
“END”
RELIABILITY
ANALYSIS
END

3.4 MARKER LOCALIZATION IN CAMERA SPACE

Based on the flowchart, the technique to localize the object was proposed. The outcome of this project was that the marker can be track and be localized in the camera space.

When the camera was connected and initialized at the Raspberry Pi, a view of the environment was registered into the Raspberry Pi. To begin the recognition, the target marker must be isolated from the background. To obtain this isolation, we must consider the color space and its threshold.

The threshold was a non-linear operation [9]. It converts a gray-scale or color image into the binary image where the two levels are assigned to pixels that are below or above the specified threshold value. It was simply black and white color. The areas of interest were selected by using this method while we can ignore the parts we were not concerned. Simple NumPy array manipulation was used to separate the pixels belonging the root system of a plant from the black background. This method needs to be implemented so that Pi camera will detect the location of the target markers in every movement.

After a threshold, the blurring technique which was Gaussian blur was used. Gaussian blur was also known as Gaussian smoothing. It was the result of blurring the image by applying Gaussian function. In this project, this method was used to reduce image noise and reduce detail [10]. The unwanted noise from surrounding might affect the clarity of the image. The effect of applying this technique was that the image will be smooth blur resemble in viewing it through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of the object under normal illumination. Bokeh effect was that quality of the blur produced in the out-of-focus parts of the image produced obviously by a lens. It also used as a pre-processing stage in computer vision in order to enhance image structure at different scales. This technique was used in this project so that more focus was implemented to the structure of the object required in the image.

After the Gaussian blur was applied, the frame needs to be converted to HSV. HSV was defined as a type of color space and similar to the modern RGB and CMYK models [11]. HSV color space contains three components which were Hue, Saturation, and Value. This Value sometimes was substituted with brightness and known as HSB. It also was known as the hex-cone color model.

In HSV, hue was the range of color. Hue was the angle from 0 degrees to 360 degrees. Different type of color has their different range of angles. Saturation indicates the range of grey in color space and has ranged from 0 to 100%. Sometimes it was calculated from 0 which the color was grey to 1 for the primary color. Faded color was because of lower saturation level which means color contain more grey. Value was the brightness of color and varies with color saturation. It has ranged from 0 to 100% which means 0 was totally black. When the value was an increase, the color spaces brightness will up and shows various of colors.

For this project, HSV color space was used in selecting the desired color to be tracked as a marker. The boundaries need to be set for every color that has been choose which was lower and upper boundaries. In HSV color space, the hue, saturation and value component for black color is at minimum. Black color was selected. The lower and upper boundary for this color was set. The lower boundary according to HSV in the algorithm was (0,0,0) and the upper boundary was (180, 255,40) where (H,S,V). Therefore, it is easier to be implemented in comparison to other shades of color.

After the black color object was detected in camera space, the contour detection will begin. Contour was one of the techniques used in this project. Contour can be determined as a curve joining all the continuous points along the boundary that having same color or intensity [12]. The contours are needed in this project because it was a useful tool for shape analysis, object detection and also object recognition. This process was applied to threshold process. In this project, the largest contour based on its area which was round shape ball need to be detected since it was decided as the marker. The radius was set to be more than 10 pixels' radius so that if the object that was not in the range will not be detected.

When the black object with the radius more than 10 pixels' was detected, background subtraction was applied. This technique was used in this project in order to require object localization for object tracking [13]. This technique was also known as foreground detection. The concepts were that detecting the moving object from the difference between the current frame and a reference frame which was often called "background image" or "background model". The moving object was segmented from the background. Background subtraction was based on a static background hypothesis so it was suitable to be applied in surgeon room where the illumination does not change and the scenes were not influenced by weather like outdoor scenes. Here were the simple arithmetic calculations where we can segment the object:

$$P[F(t)] = P[I(t)] - P[B] \tag{3.1}$$

in which P[I(t)] was pixel value, P[B] was a background image.

Then, edge detection was applied to the background subtraction. Edge detection was one of image processing technique to find the boundaries of objects within images. It was used for image segmentation and data extraction without disturbing structural properties of the image. Edge detection works by detection of discontinuities in brightness. Applying edge detection to an image significantly reduce the amount of data to be processed and filter out the less relevant information.

To enable a systematic evaluation of tracking capability, a quadrant based camera space was introduced in the experiment. Location of the object based in the quadrant view from the camera space was depicted in Figure 3.3.

Q1	Q2
Q3	Q3

Figure 3.3: Figure of quadrant

It was easier when using this quadrant of view when locating the marker position. The marker position can be identified according to the quadrant. After the quadrant used was initialized, the location was lock. The motion of the object was tracked based on the *(x, y)*-coordinates. The coordinates of the object tracked were depend on the deque. The larger the deque, the more coordinates will be tracked. This size of the deque was depended on [--buffer] command. In this experiment, the deque was set to be 32.which means the buffer of *(x,y)*-coordinates of object in this experiment will maintain only for the previous 32 frames. The centroid of the object was simply the center *(x,y)*-coordinates of the object. The direction of the object was based on the deltas different between the *x* and *y* coordinates of the current frame and a frame towards the end of the buffer. The direction was determined based on the sign of dX and dY.

On the future project, this Raspberry Pi and Pi camera will be mount at the surgical platform. The camera will be placed near the light source. The data of the marker position will be sent to the controller and then the controller will send signal to DC motor that will move this lamp based on pan and tilt system.

3.5 STATISTICAL VERIFICATION

A few test has been conducted to fulfill the objective 3 which was to evaluates the precision of marker's localization. All the test conducted in this project was repeated until normality was observed.

3.5.1 PRECISION

Precision refers to the closeness of two or more measurements to each other. Precision of measurement can be determined by a statistical method which was standard deviation. The standard deviation (SD), was a measure that was used to quantify the amount of variation or dispersion of a set of data values. It was how much on average of measurements differ from each other. A low standard deviation indicates that the data points tend to be close to the mean of the set which indicates high precision, while a high standard deviation indicates that the data points are spread out over a wider range of values which indicate low precision.

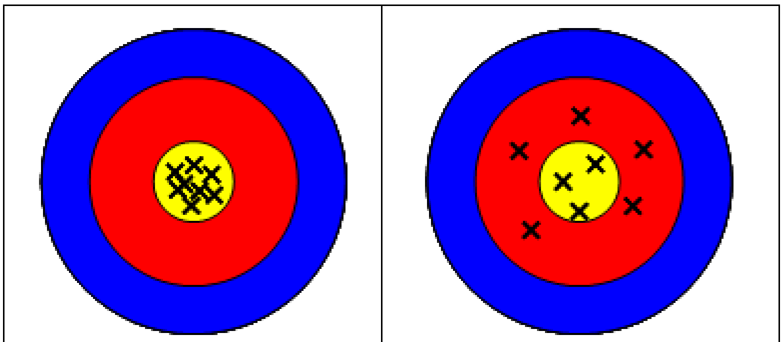
Formula for standard deviation:

$$SD = \sqrt{\frac{\sum(\mathrm{X}-\bar{\mathrm{X}})^2}{n-1}} \tag{3.2}$$

Where:X = value in data set

$\bar{\mathrm{X}}$ = mean of data set

N = number of data points



precise because the x marks close to each other

not precise because the x mark are scattered

Figure 3.4: Example of precision system

3.5.2 ROOT-MEAN-SQUARE ERROR (RMSE)

Root-mean-square error (RMSE) was also known as root-mean-square deviation (RMSD). RMSE was the square root for the average squared errors. RMSE was frequently used to measure the differences between values which are sample and population values that were predicted by a model or an estimator with the values that was observed. In simple words, it was difference between actual/observed values and predicted values. The different between the actual/observed and predicted values can be positive or negative as the predicted value under or over estimates the actual values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2}{n}} \quad (3.3)$$

Where: $\bar{y}$ is the predicted values

y is the actual/observed values

n is total data size

3.5.3 PERCENTAGE OF ERROR

The reliability and accuracy of the system was determined from the error produce by the system. The more the error produce, the less the reliability and accuracy of the system and vice versa. So to make a good system, less error was required.

This was the formula to calculate the error of the system:

$$\%error = \frac{|Exact\ value - Approximate\ value\,|}{Exact\ value} \times 100 \qquad (3.4)$$

3.5.4 FIELD OF VIEW (FOV)

Every optical device has the open observable area similar to the observable area of a person can see through his or her eyes. This open observable area was known as field of view (FOV). FOV allows the coverage of an area more than a single focus point. Wider FOV provides better sensor coverage. Angle of view can be measured horizontally, vertically, or diagonally. The horizontal field of view will be used and the other two will be derived from horizontal field of view formula [14].

horizontal field of view = 2 atan(0.5 width / focallength)[14] (3.5)

Width was the horizontal width of the sensor and this formula can similarly be used to calculate the vertical aperture using the vertical height of the film area shown below.

vertical field of view = 2 atan(0.5 height / focallength)[14] (3.6)

In the machine vision, field of view was the area of the inspection captured on the camera's space. The relationship between the field of view and the working distance was set up by the lens focal length and the size image produce by the sensor. The image resolution was affected by the size of the field of view and the size of the imager view from the camera. This was the factor in determining the accuracy of the sensor and the camera. Working distance for this project was the distance of the camera and the marker. In this study, the Pythagorean theorem triangle was simply applied based on the understanding of the concept in order to get the value of cm/pixel.

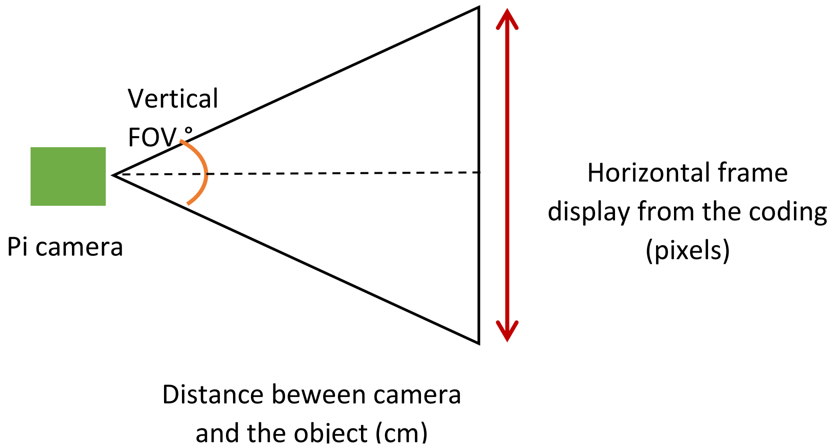


Figure 3.5: Field of view of Pi camera

The vertical FOV of the Pi camera is 41.41° ± 0.11°. The distance between the camera and the object was fixed (30cm, 50cm, 70cm, 100cm). The horizontal frame was 440 pixels. Applying the Pythagorean theorem, the vertical FOV and the horizontal frame was divided by 2 in order to get the value for the half triangle. So the equation will be:

$$\tan\left(\frac{\theta}{2}\right) = \frac{(\frac{y}{2})}{x} \qquad (3.7)$$

Where: θ is vertical field of view in degree

x is the distance between camera and the object in cm

y is the vertical frame in cm

The image captured by the Pi camera was recorded at vertical height of 440 pixels. Therefore, to determine the ratio of the actual height to the pixel counts, the actual height was determined first. For example, the furthest distance which was 100 cm was used as the distance between the camera sensor and the object.

Height, y/2 in cm = tan (41.4°/2) × 100cm (3.8)

= 37.786 cm

Therefore each pixel corresponds to $=\frac{37.786}{440/2} = 0.17\ cm$ of oject height in physical world.

Table 3.3: Conversion of unit of cm/pixel to pixel/cm

Distance beween camera and the object (cm)	$\frac{\text{Vertical frame (cm)}}{\text{original frame (pixel)}}$	$\frac{1}{\frac{Vertical\ frame\ (cm)}{original\ frame\ (pixel)}}$
30	0.0515	19.4175
50	0.0858	11.6550
70	0.1202	8.3195
100	0.1717	5.8241

3.6 HARDWARE IMPLEMENTATION

In this project, the Raspberry Pi 3 model B was used. The Raspberry Pi was protected in the casing with the cooling system. The purpose of this casing was to protect this microprocessor from a bump or drop and prevent it from overheat during operation. This Raspberry Pi only can be connected to an external power supply rated at 5V dc and a minimum current of 2 Amp. So the correct power supply needs to be considered to avoid the malfunction of the component in this processor. The HDMI-VGA cable was used to connect this processor to the monitor.

The installation of software in the Raspberry Pi was the most important part of this project. Without the installation, the tracking system cannot be done. This installation must be correct so that this microprocessor can work well with the software installed which is python and OpenCV library. The first step in this installation is download Raspbian Jessie at the official site which is https://www.raspberrypi.org/. After that is expand the filesystem to include all available space on your micro-SD card. The Kingston 16GB micro SD has been used as a memory card. The command for this process is $ sudo raspi-config. Once prompted select Expand File System and hit Enter. After that select <Finish> button and then reboot the Raspberry Pi, $ sudo reboot. After rebooting, the file system should have been expanded to include all the available space on micro-SD card. This can be verified by executing df –h and examining the output.

Next step is the installation of the OpenCV. It takes about 1 hour and 12 minutes to compile it. The first step was updating and upgrading any existing packages. To configure the OpenCV build process, we need to install some developer tools which include Cmake. Next was install some image I/O packages that allow us to load various image file formats from disk, for example, JPEG, PNG, TIFF, etc. We also need video I/O packages. These libraries allow the various video file to be read from disk as well as work directly with video streams. The OpenCV comes with sub-module named highgui which was used to display images on our screen and build basic GUIs. GTK needs to be installed in order to compile the highgui. Many operations inside the OpenCV can be optimized further by installing a few of dependencies. These optimizations were important for resource-constrained devices.

Lastly was the installation of python. To make the OpenCV can work with both Python 2.7 and Python 3 header files, install both will compile the OpenCV with Python bindings. If this step were skipped, an error related to the Python.h header file will not be found when running makes to compile OpenCV. Before compiling OpenCV on Raspberry Pi, pip needs to be install. It is a Python package manager. Python virtual environment is highly recommended in this Raspberry Pi. A virtual environment was a special tool that keeps the dependencies required by different projects in separate places by creating isolated, independent Python environments for each of them. The virtual environment was used in computer vision. Next was the installation of Numpy. Numpy was the only python dependency which was used for numerical processing. After all the above step, we can already compile and install OpenCV in Raspberry Pi.

The next process was the installation of Pi camera. Be aware and careful in handling this Pi camera. The camera can be damaged by static electricity, so before removing the camera from its grey anti-static bed, you have to make sure to discharge yourself by touching an earthed object, for example a radiator or PC Chassis. Install the Raspberry Pi camera module by inserting the cable slot into the connector that situated between the Ethernet and HDMI ports with the silver connectors facing the HDMI port. After that, boot up the Raspberry Pi. Open the prompt in the Raspberry Pi then run “sudo raspi-config”. Navigate to the camera option and enable it. After that select “Finish” and reboot the Raspberry Pi. This camera was situated near to the lamp as shown in figure 3.6.

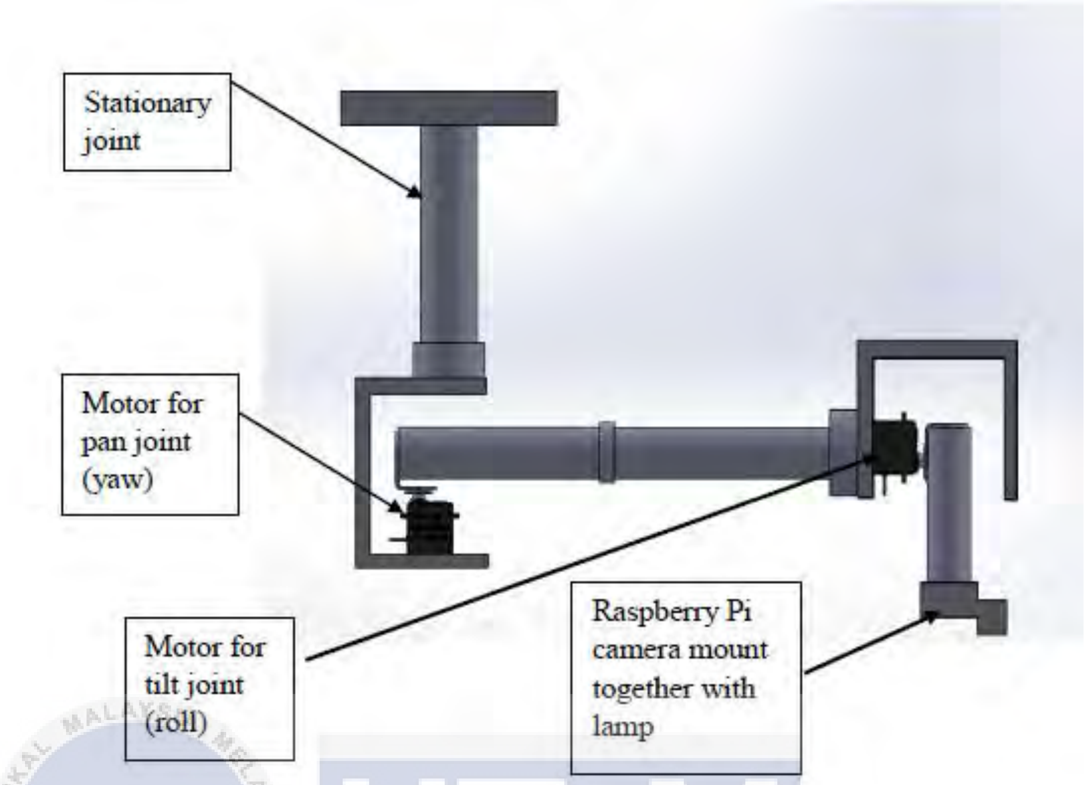

Stationary joint
Motor for pan joint (yaw)
Motor for tilt joint (roll)
Raspberry Pi camera mount together with lamp

Figure 3.6:Hardware design of the surgical illumination system

This design was proposed to be used in this project. The Pi camera was mounted together near the lamp. When this Pi camera has detected the marker, it will send information to the controller. The controller then sends a signal to dc motor that enables the surgical lamp moved to the target direction.

3.7 EXPERIMENTAL SETUP FOR ANALYSIS

This section explained the setup to carry experiment for analysis. The performance of the system was analyzed based on three experiments which were the brightness of light, size of the object based on the distance of the object from the camera and lastly the presence of the object in the camera space.

3.7.1 EXPERIMENT 1: BRIGHTNESS OF LIGHT

This experiment was conducted in a controlled intensity of light to differentiate the intensity of light from the condition of normal room even though the lux produce from the light source is not in range within 40,000 to 160,000. The five light source zoom headlamp in figure 3.7 has been chosen as the light source as it can produce bright light. This headlamp can produce 35000 lumens of brightness and can produce 5000- 6000 lux of light which is 4 times brighter than general lighting of operating room. The camera was mounted near to the headlamp so that the optimum light intensity can be produce by this light and the object can be detected by the camera.

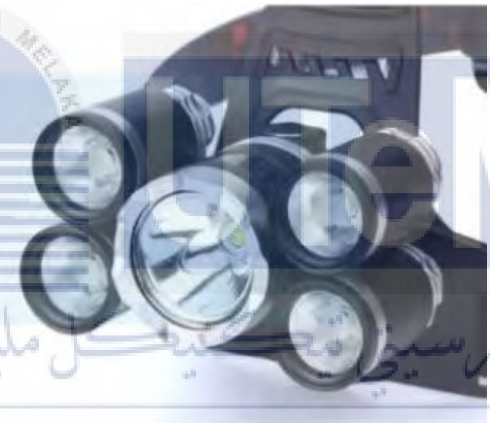


Figure 3.7: Five light source zoom headlamp

This experiment was conducted in two condition which is in condition 1 was in normal lighting condition and condition 2 was in the dark place. This both experiment was conducted to differentiate the different between the intensity of light and to make a conclusion whether it will give effect to the effectiveness of the system in tracking the marker. In condition 1, there was another source of light which was the fluorescent lamps. This condition quite similar to the environment of operation room even though it was not bright as in real situation. As we know, the surgical room were supplied with high intensity of light and it was not only from one source of light so that the presence of the shadow that would disturb the doctor's vision can be overcome.

In the surgical room, the normal operating theatre light was used to avoid adaptation problem. For condition 2, the experiment was conducted in very limited intensity of light. This experiment was conducted in a dark place so that only the light source from the head lamp will be use. Even though this experiment was conducted in a dark place, but the light produce by the headlamp still has higher level of lux compare to the general lighting of operating room.

The brightness of lights was measured by using digital lux meter. The unit of the brightness will be in 1x. Lux was the unit for the amount of visible light. Three black circular ball with different diameter were used as a marker. The lux level for every condition at different distance was shown in table 3.4.

Table 3.4: Lux level for both condition at different distance

Condition Distance form light source (cm)	With fluorescent lamp	No fluorescent lamp
30	5200-5300 lux	4900-5000 lux
50	2700-2800 lux	2400-2500 lux
70	1800-2000 lux	1700-1800 lux
100	1200-1300 lux	1100-1200 lux

3.7.2 EXPERIMENT 2: SIZE OF OBJECT IN CAMERA SPACE

The distance of the object from the camera has become the manipulated variable in this setup. Logically, the size of the object will become smaller when viewed at the camera after the distance was added. Different distances were used in this setup.

The distances were set to 30cm, 50cm, 70cm and 100cm. The distances were measured by using measuring tape. This range of distance was used as a prove to the real concept of the distance between the surgical illumination system to the surgeon working area. The markers was set according to this distance. The optimum distance was selected so that the view of the markers from the camera was not too small and the amount of light that reaches the surgical area were maximum. This distance also important for the placement of the camera. Three black circular ball with different size were used as a marker. This experiment was conducted as stated in case study 1.

Table 3.5: Table of experiment for radius of marker in camera space based on the distance of marker from camera

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixels)				Mean	Standard deviation	root-mean-square-error (RMSE)
30											
50											
70											
100											

3.7.3 EXPERIMENT 3: POSITION OF OBJECT IN CAMERA SPACE

The Pi camera was used to detect the marker based on the characteristic stated before. To make sure the illuminating system working perfectly, the marker must be situated in the camera space. This test was conducted to ensure that the marker can be detected at all part of the camera space and to test the precision of the Pi camera in detecting the marker.

The camera space was divided equally into 4 quadrants which is Q1 (from (0,0) to (300,220)), Q2 (from (301,0) to (600,220)), Q3 (from (0,221) to (300,440)) and Q4 (from (301,221) to (600,440)). All the marker was observed in each quadrant. The marker selected must situated in the camera space so that the amount of light reach at the surgical area will be maximum as the surgical lamp is in the correct position. In this experiment, three markers with different diameter were placed at a certain distance. This experiment was conducted as stated in experiment 1.

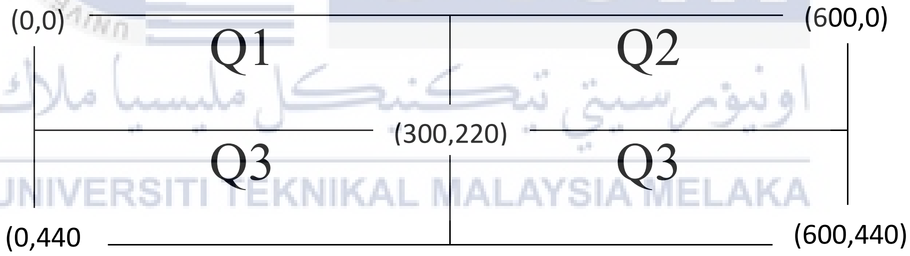


Figure 3.8: The 4 quadrant in camera space and its coordinate

Table 3.6: Table of experiment for Radius of marker in each quadrant based on different distance

Marker size (cm)	Detection Yes / No				Radius of marker in camera space (pixels)				Mean	Standard deviation	root-mean-square-error (RMSE)
4.0											
7.4											
9.3											

3.8 EXPERIMENTAL REQUIREMENT

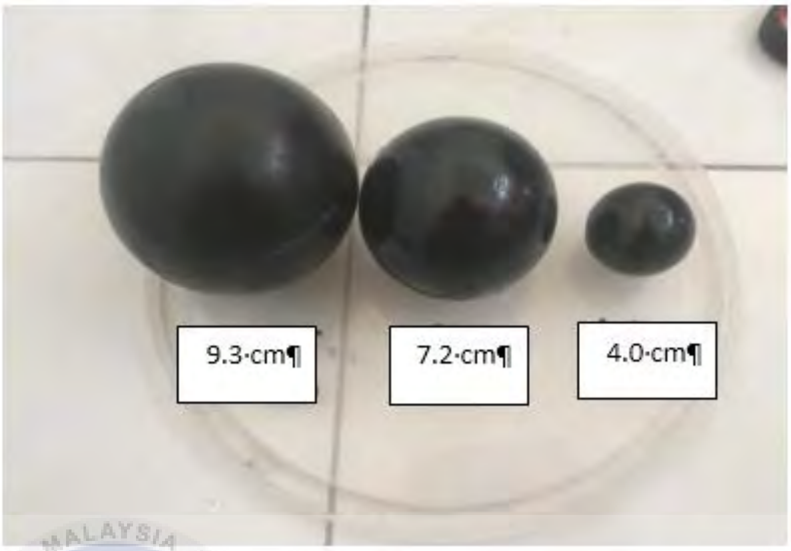


Figure 3.9: Three black ball with different size

3 black balls with different size were used in this experiment as the marker same as in Figure 3.9.

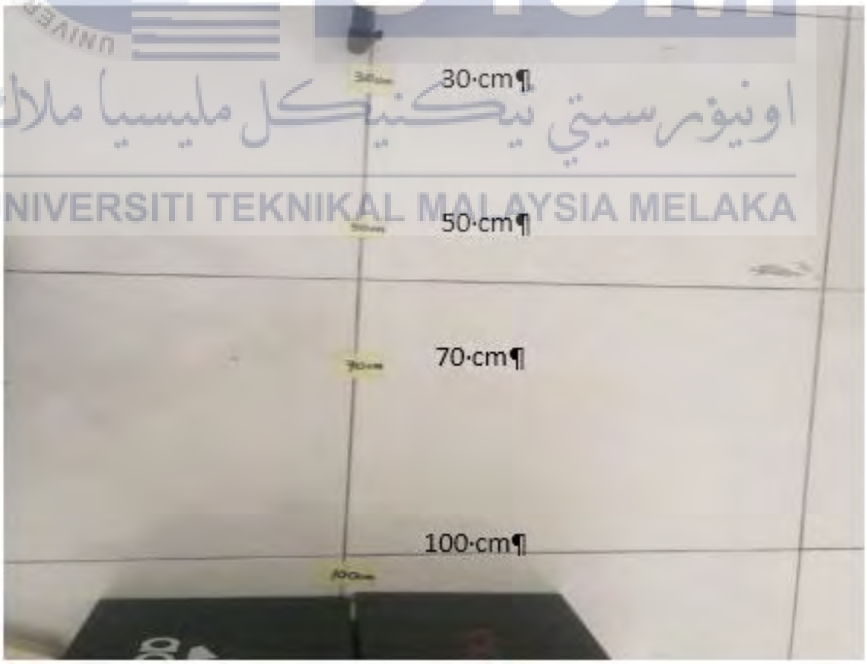


Figure 3.10: 4 different distance between object and the camera

4 different distance of the marker from the Pi camera which were 30cm, 50cm, 70cm and 100cm that were measured by measuring tape and being mark on the floor as shown in the Figure 3.10.

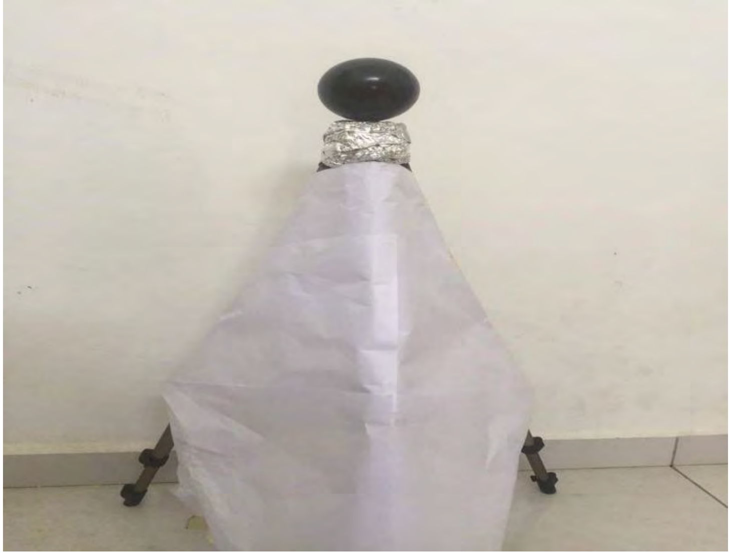


Figure 3.11: Position of object on the tripod

The ball was placed on the tripod same as in Figure 3.11

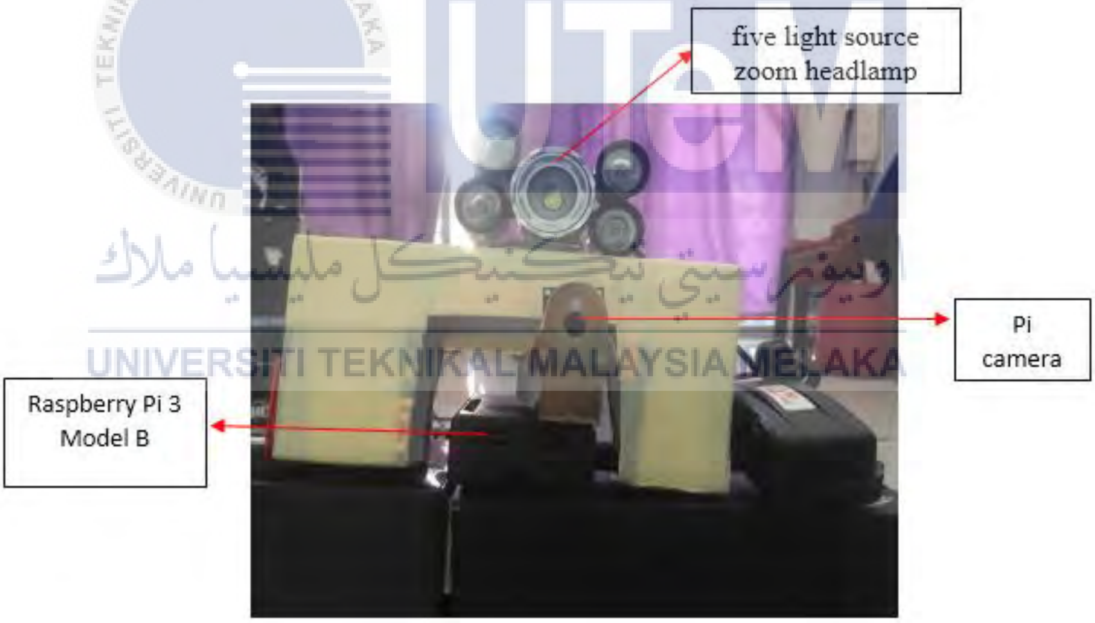


Figure 3.12: The setup of camera and the headlamp

The Pi camera and the headlamp was setup according to Figure 3.12

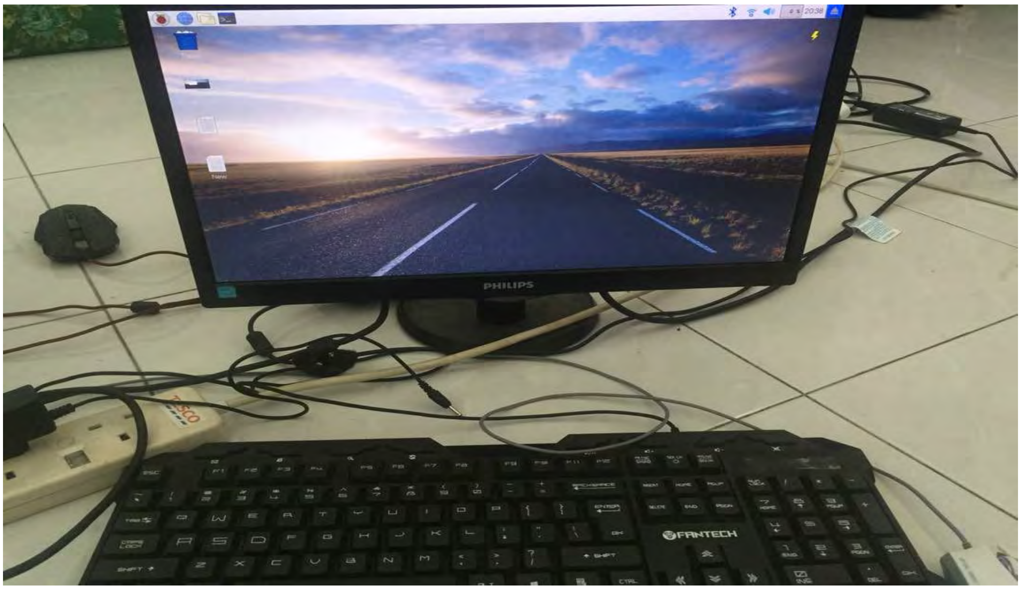
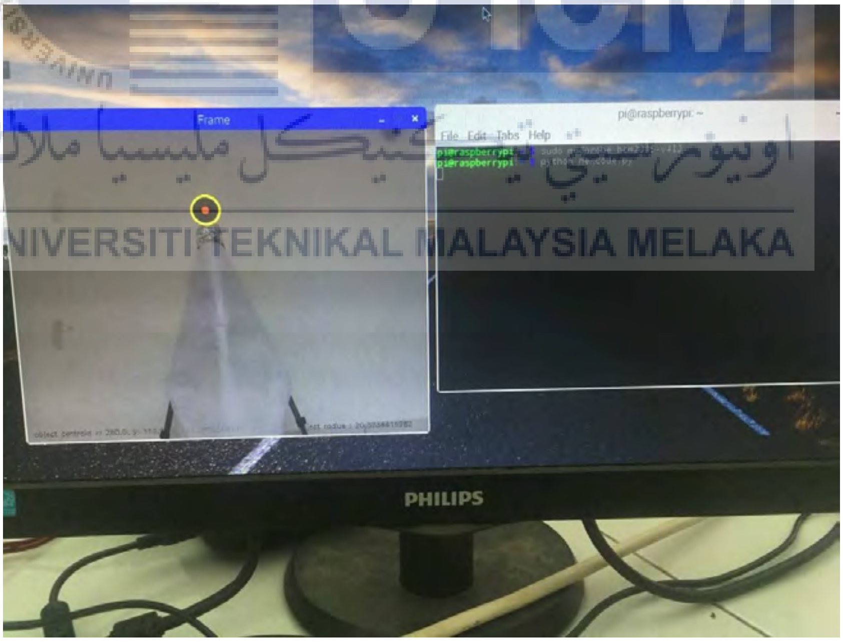


Figure 3.13: The monitor displays the program desktop of Raspberry Pi

The Raspberry Pi was connected to the monitor by using the VGA-HDMI cable so that all the program in the Raspberry Pi can be display at the screen monitor like in figure 3.13



```
pi@raspberrypi:~ $ sudo modprobe bcm2835-v4l2
pi@raspberrypi:~ $ python [illegible]
```

Figure 3.14: The display of the camera space in the monitor

Open the terminal and enter the command of sudo modprobe bcm2835-v4l2. sudo modprobe bcm2835-v4l2 is a driver. The Raspberry Pi need to activate the driver after any reboot so that the camera will have no problem in capturing a picture or video. After that, enter the command to run the algorithm. The view recorded by camera was displayed by the monitor same in Figure 3.14.

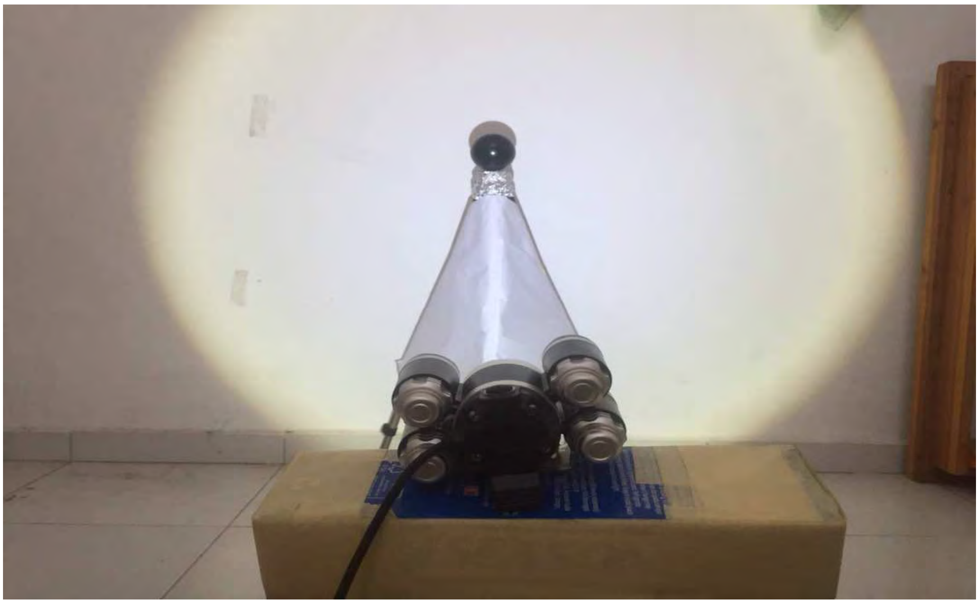


Figure 3.15: Final setup of the experiment with the source of light from headlamp

Final setup of the experiment was same in Figure 3.15 before data was collected. The marker captured was displayed in frame at the monitor screen. The frame was displayed the centroid and the radius which was the size of the marker. This experiment was conducted according to the experiment stated before.

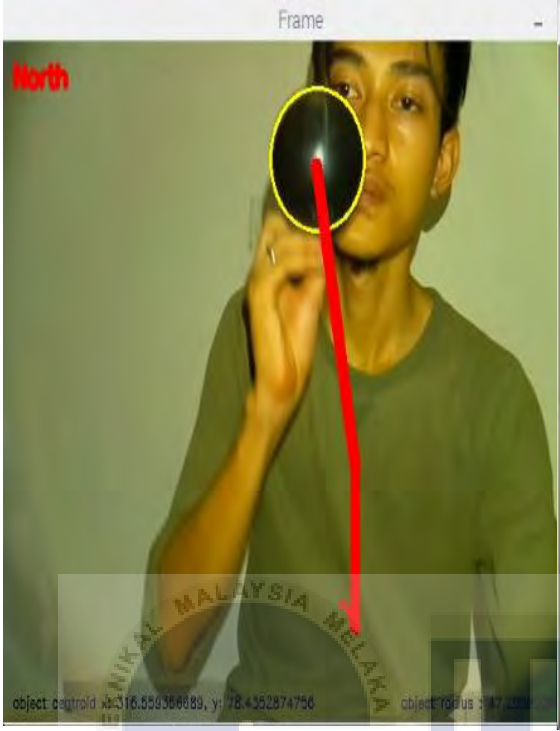
CHAPTER 4

RESULT AND DISCUSSION

4.1 RESULT

This chapter explained about the results obtained from the execution of algorithm for marker detection based on color and shape. This execution was done in Raspberry Pi and the object was captured by the Pi camera.

4.1.1 DIRECTION OF THE OBJECT

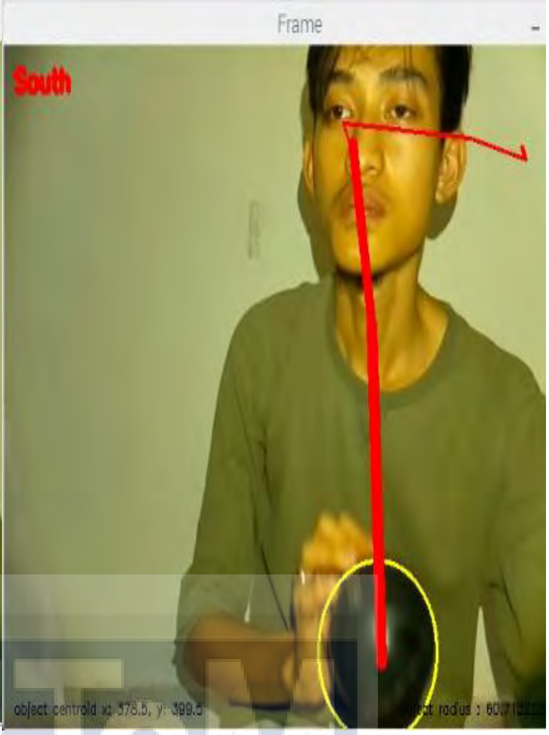


object centroid x: 316.559386689, y: 78.4352874756

object radius :

Object move to North

dY is positive

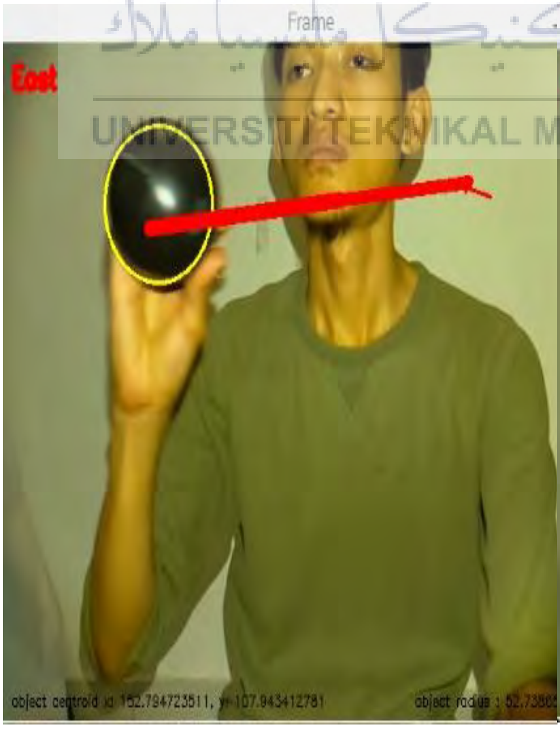


object centroid x: 378.5, y: 399.5

object radius : 60.71

Object move to South

dY is negative

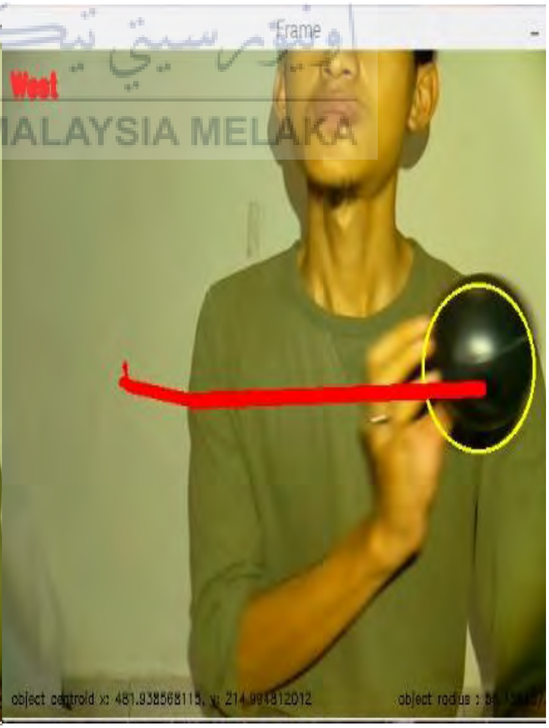


object centroid x: 152.794723511, y: 107.943412781

object radius : 52.7386

Object move to East

dX is positive

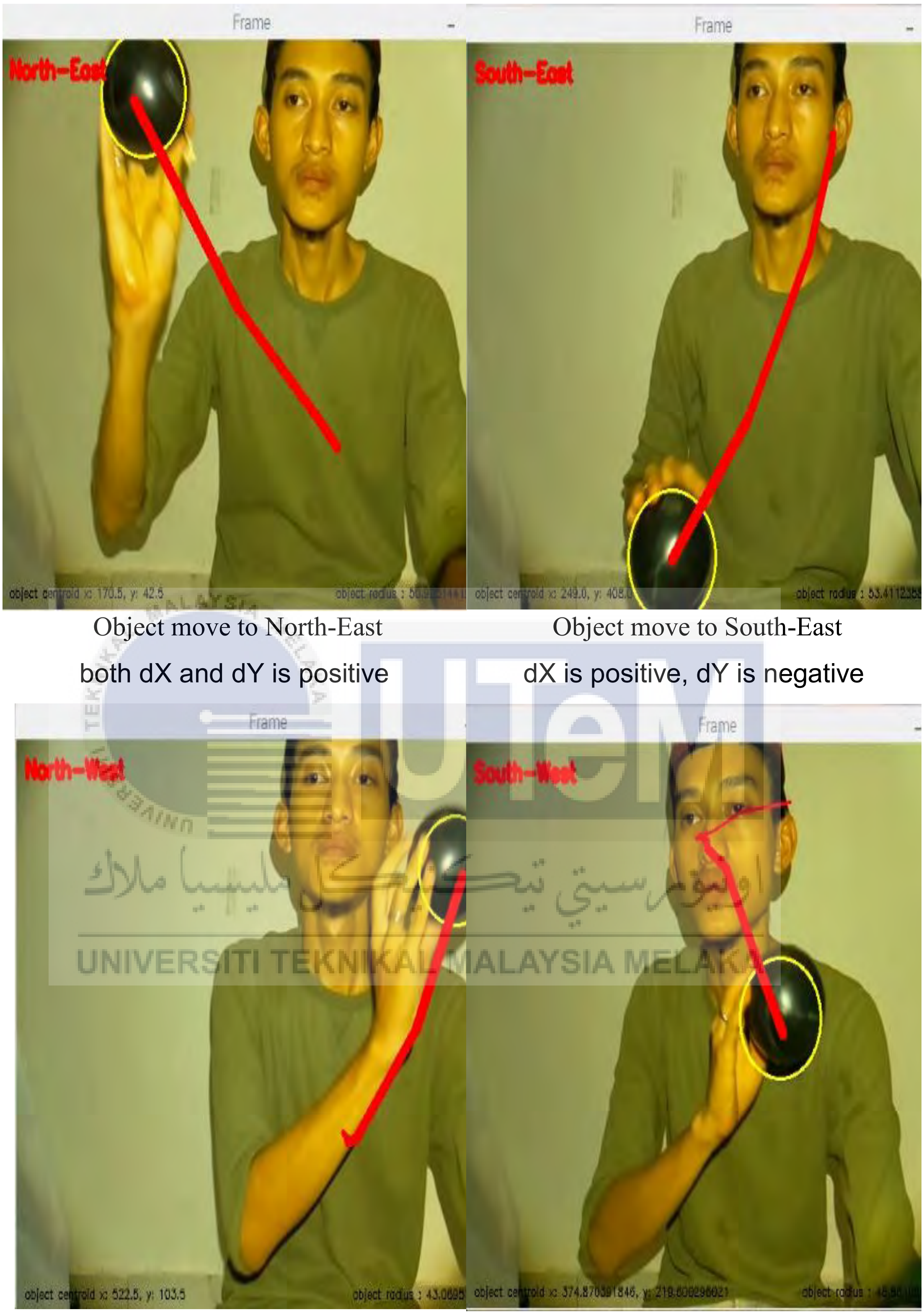


object centroid x: 481.938568115, y: 214.994812012

object radius :

Object move to West

dX is negative


Frame
North-East
object centroid x: 170.5, y: 42.5
Frame
South-East
object centroid x: 249.0, y: 408.0
object radius : 53.41123
Frame
North-West
object centroid x: 522.5, y: 103.5
object radius : 43.0695
Frame
South-West
object centroid x: 374.870391846, y: 219.609296021

Object move to North-East

both dX and dY is positive

Object move to South-East

dX is positive, dY is negative

Object move to North-West

dX is negative, dY is positive

Object move to South-West

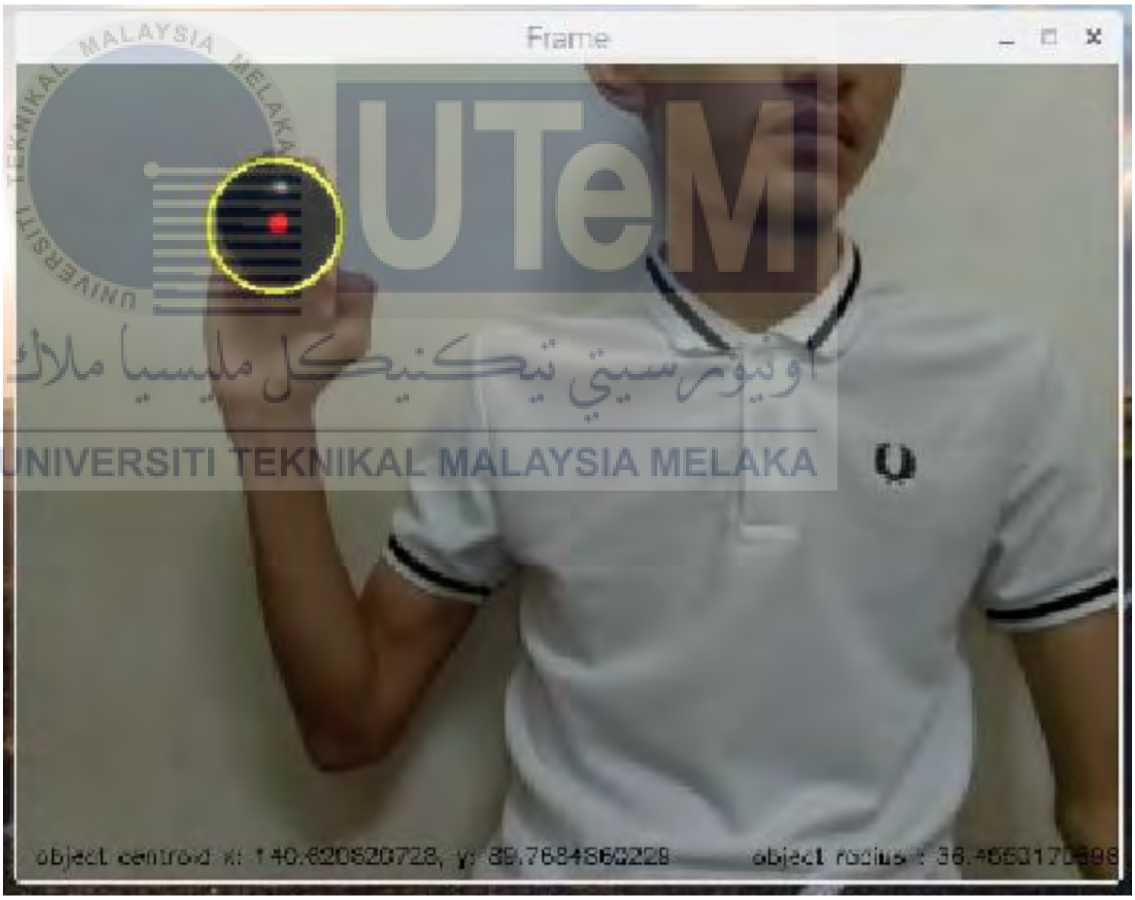
Both dX and dY is negative

Figure 6:Result of object direction from coding execution

The motion of the object was tracked based on the (x, y)-coordinates. The centroid of the object was simply the center (x, y)-coordinates of the object. The direction was determined based on the sign of dX and dY. This results present the test of localization algorithm

4.1.2 POSITION OF OBJECT AT EVERY QUADRANT

Quadrant 1



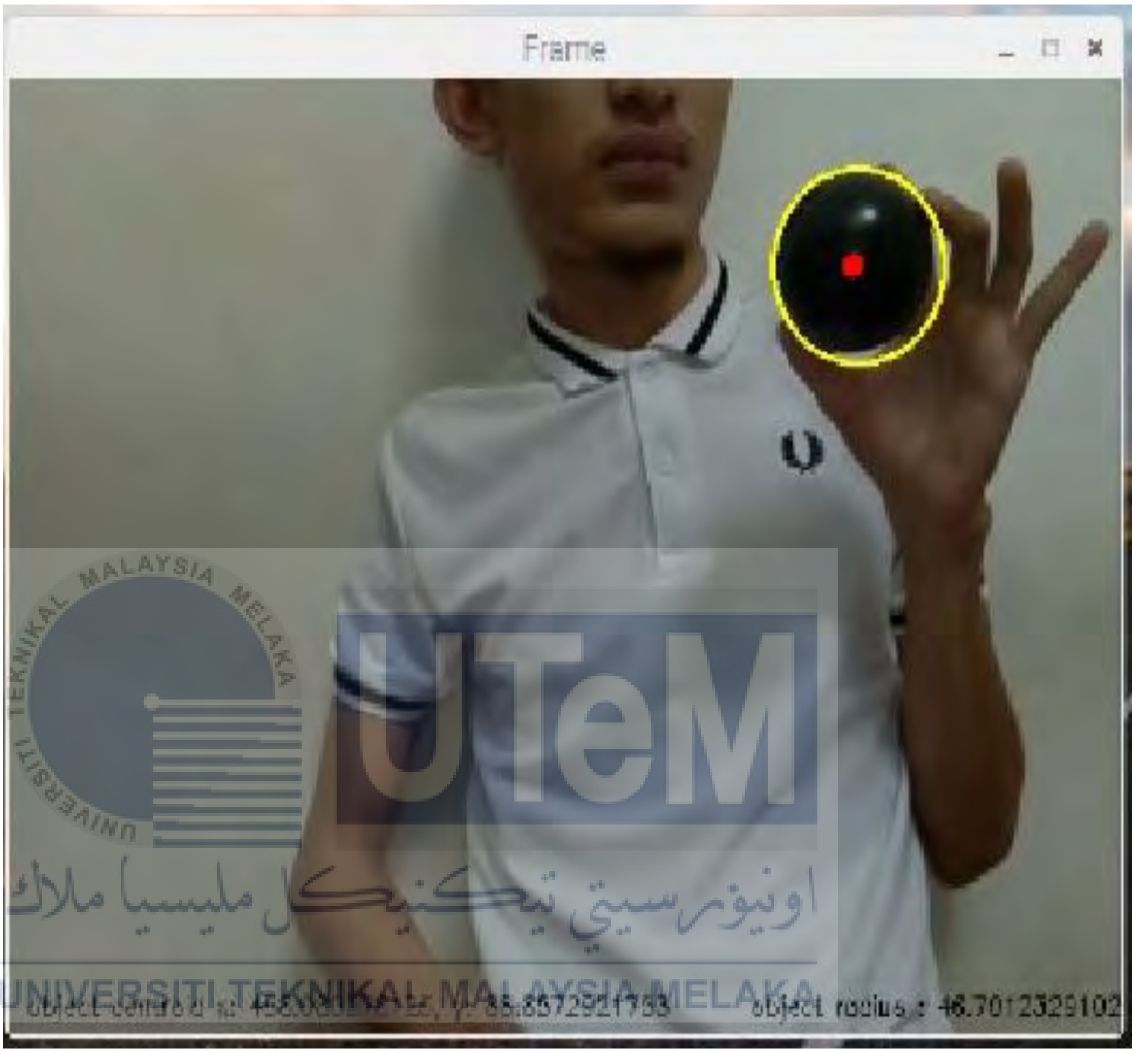
```
object centroid x: 140.820820728, y: 89.7684860228        object radius: 36.4553170898
```

The object centroid is at x: 140.82, y:89.76

Figure 4.2: Position of ball at quadrant 1

Based on the figure 4.2, the ball was situated in quadrant 1 since the centroid was detected to be at x: 140.82, y: 89.76

Quadrant 2



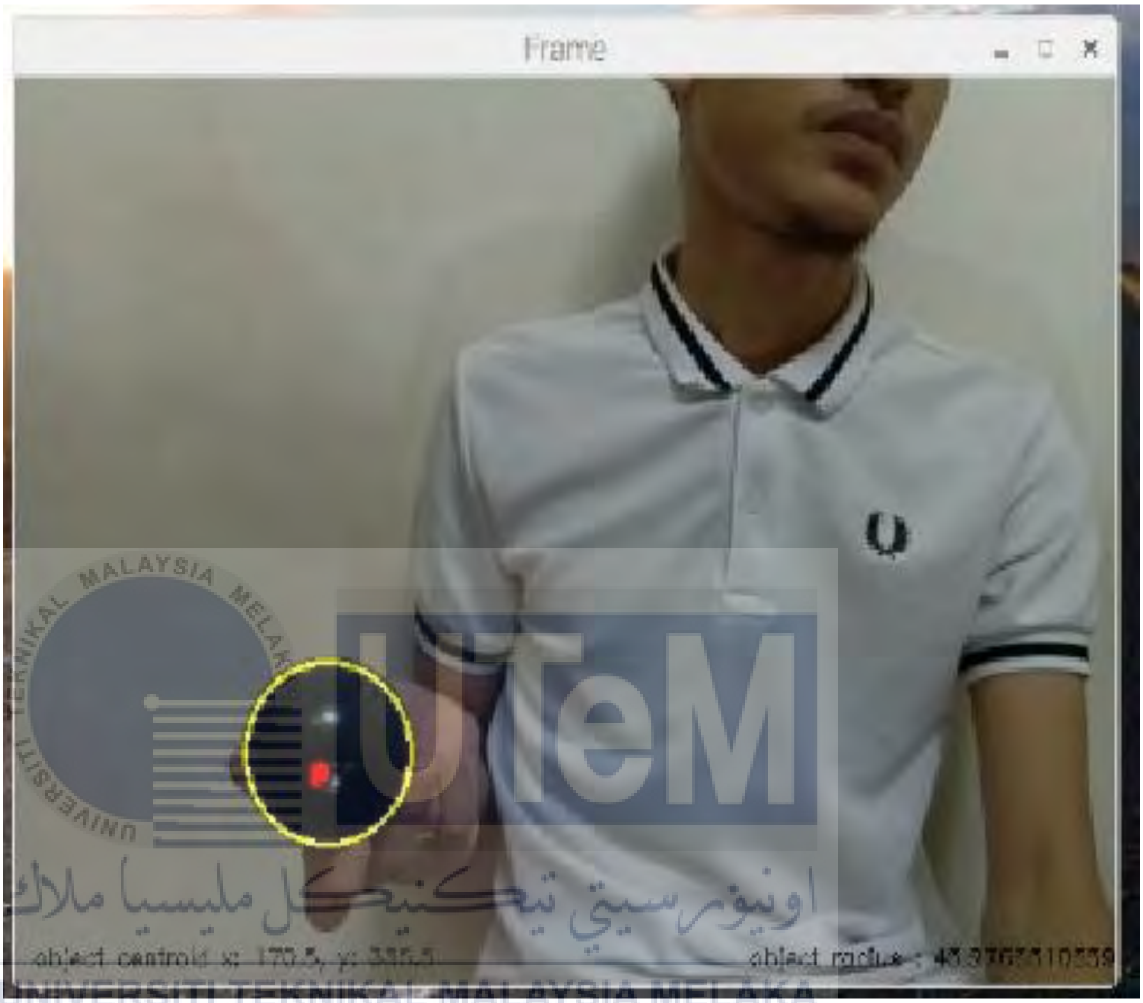
```
object centroid x: 458.08[illegible], y: 88.8572921753          object radius : 46.7012329102
```

The object centroid is at x: 458.08, y:88.85

Figure 4.3: Position of ball at quadrant 2

Based on the figure 4.3 the ball was situated in quadrant 2 since the centroid was detected to be at x: 458.08, y: 88.85

Quadrant 3

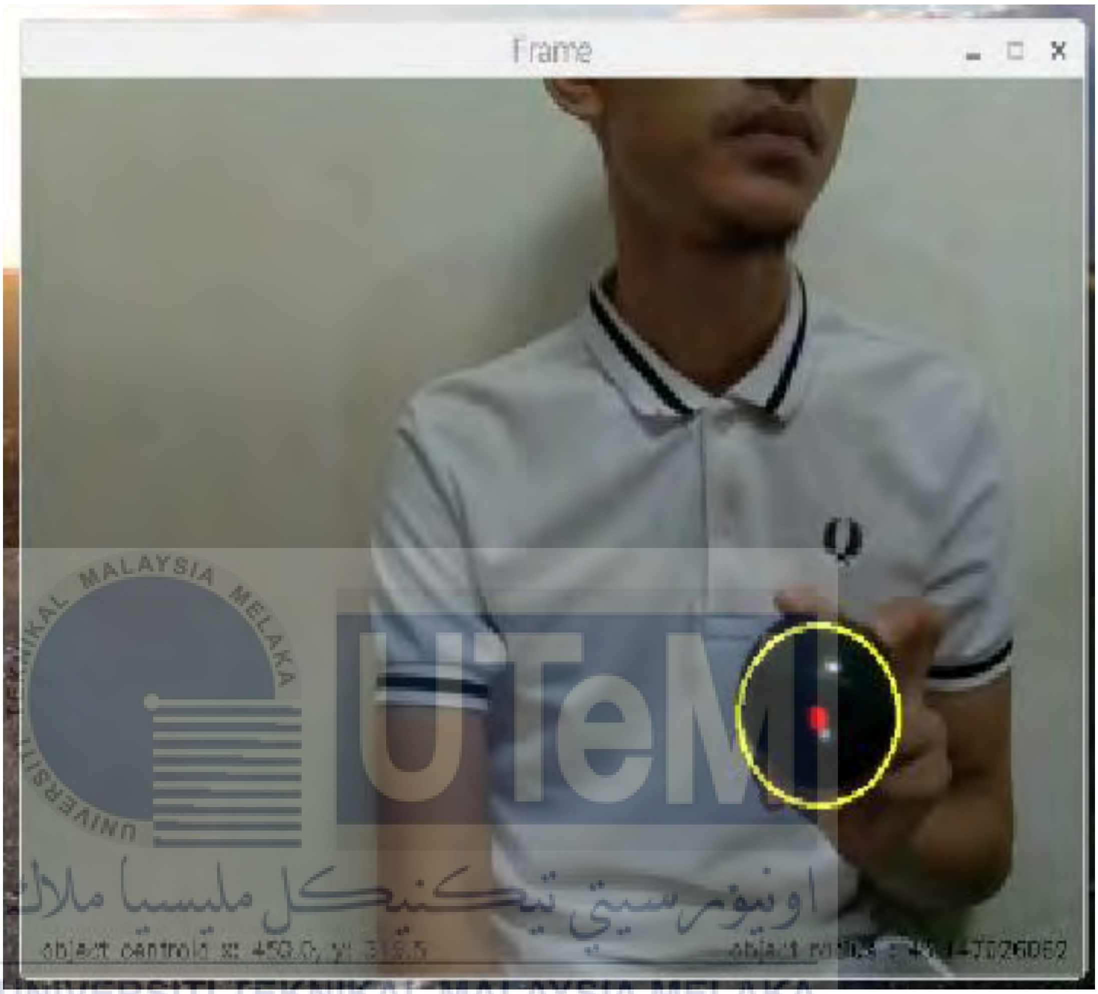

Frame
object centroid x: 170.5, y: 335.5
object radius : 40.3767310339

The centroid of object is at x: 170.5, y: 335.5

Figure 4.4: Position of ball at quadrant 3

Based on the figure 4.4 the ball was situated in quadrant 3 since the centroid was detected to be at x: 170.50, y: 335.50

Quadrant 4



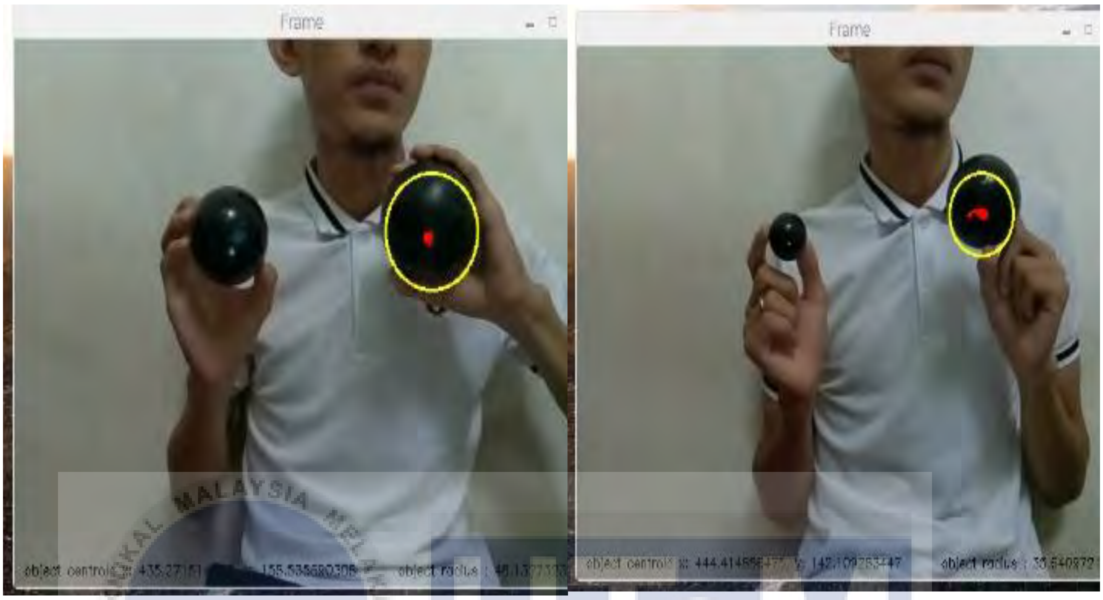
```
object centroid x: 450.0, y: 319.5
object radius : [illegible]
```

The centroid of object is at x: 450.00, y: 319.50

Figure 4.5: Position of ball at quadrant 4

Based on the figure 4.5 the ball was situated in quadrant 4 since the centroid was detected to be at x: 450.00, y: 319.50

4.1.3 DETECTION OF OBJECT BASED ON SIZE



object centroid x: 435.27151 [illegible] 155.535680308 object radius : 46.1327333

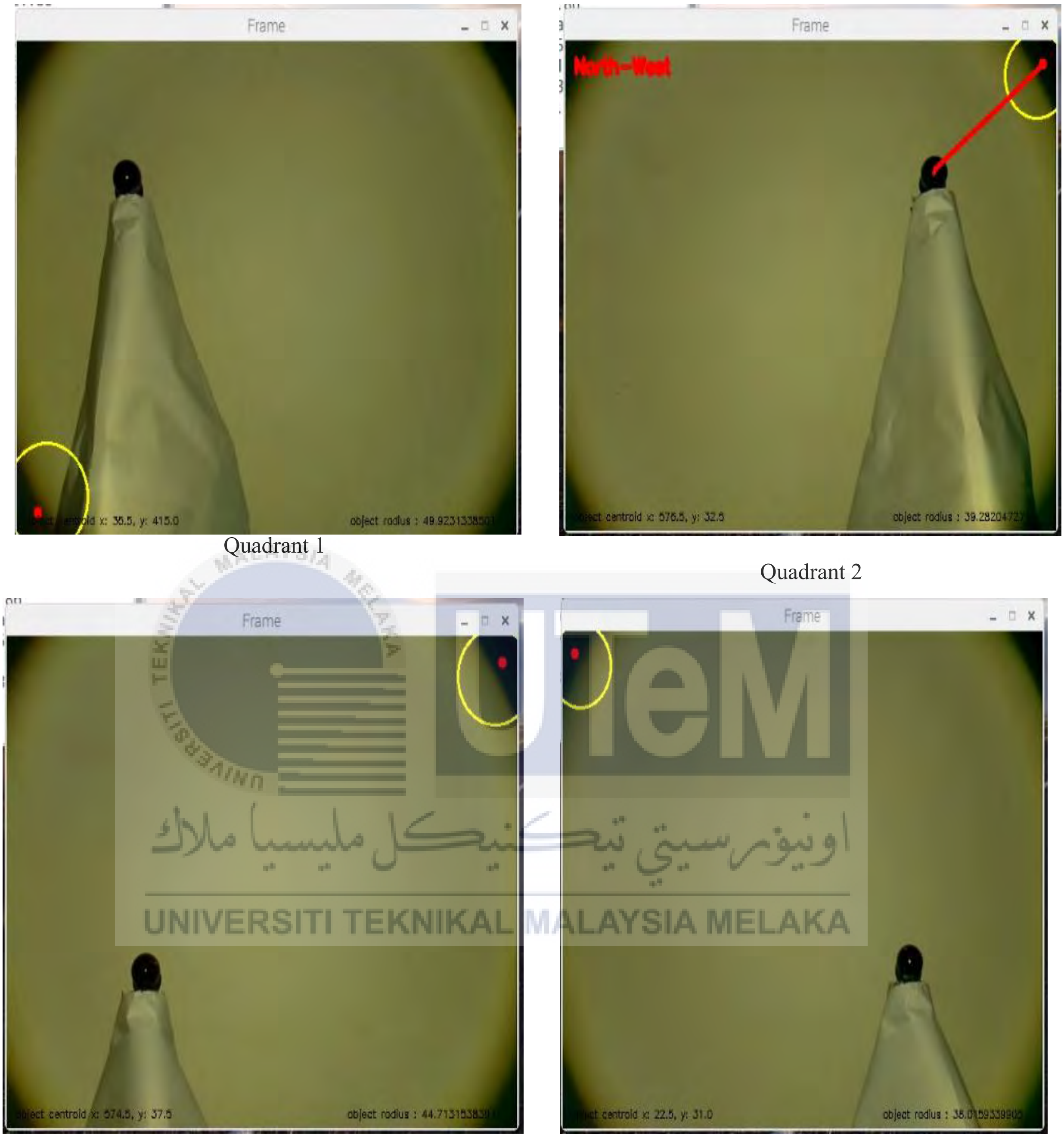
object centroid x: 444.414856477 y: 142.109263447 object radius : 33.640872

9.3 cm ball has been detected by the camera compare to 7.2 cm ball

7.2 cm ball has been detected by the camera compare to 4.0 cm ball

Figure 4.6: Detection of camera to different size of object

From figure 4.6, the camera was detecting the larger size of black ball. The system was successfully run according to the algorihm build for the system which need the camera to detect larger radius of object which was more than 10 pixel.



ct centroid x: 36.5, y: 415.0

object radius : 49.9231338501

ect centroid x: 576.5, y: 32.5

object radius : 39.2820472

Quadrant 1

Quadrant 2

object centroid x: 574.5, y: 37.5

object radius : 44.7131538391

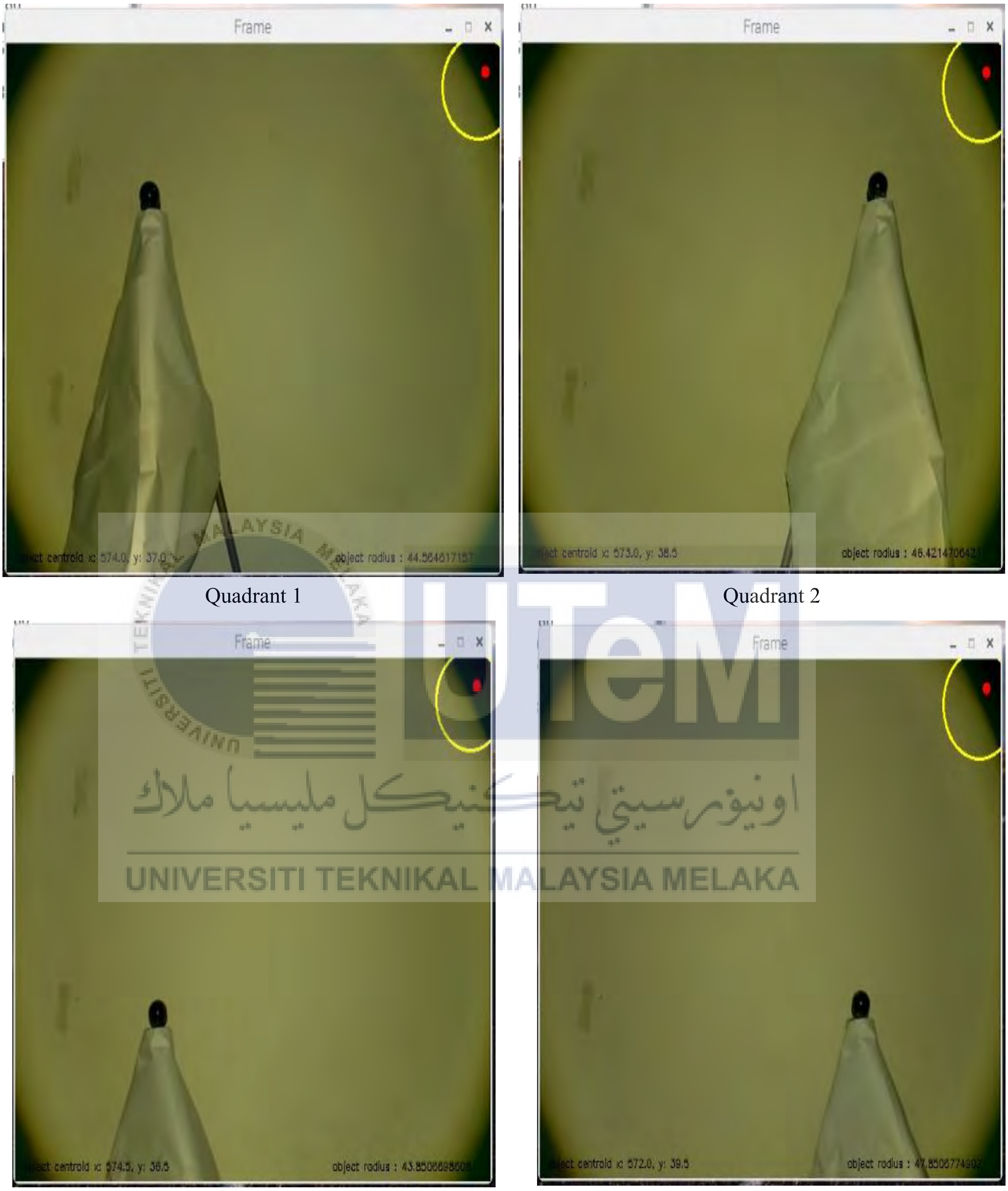
ct centroid x: 22.5, y: 31.0

object radius : 38.0159339905

Quadrant 3

Quadrant 4

Figure 4.7: Error for 4.0 cm ball at distance 70 cm from the camera



```
ject centroid x: 574.0, y: 37.0        object radius : 44.564617157
```

Quadrant 1

```
ject centroid x: 573.0, y: 38.5        object radius : 46.4214706421
```

Quadrant 2

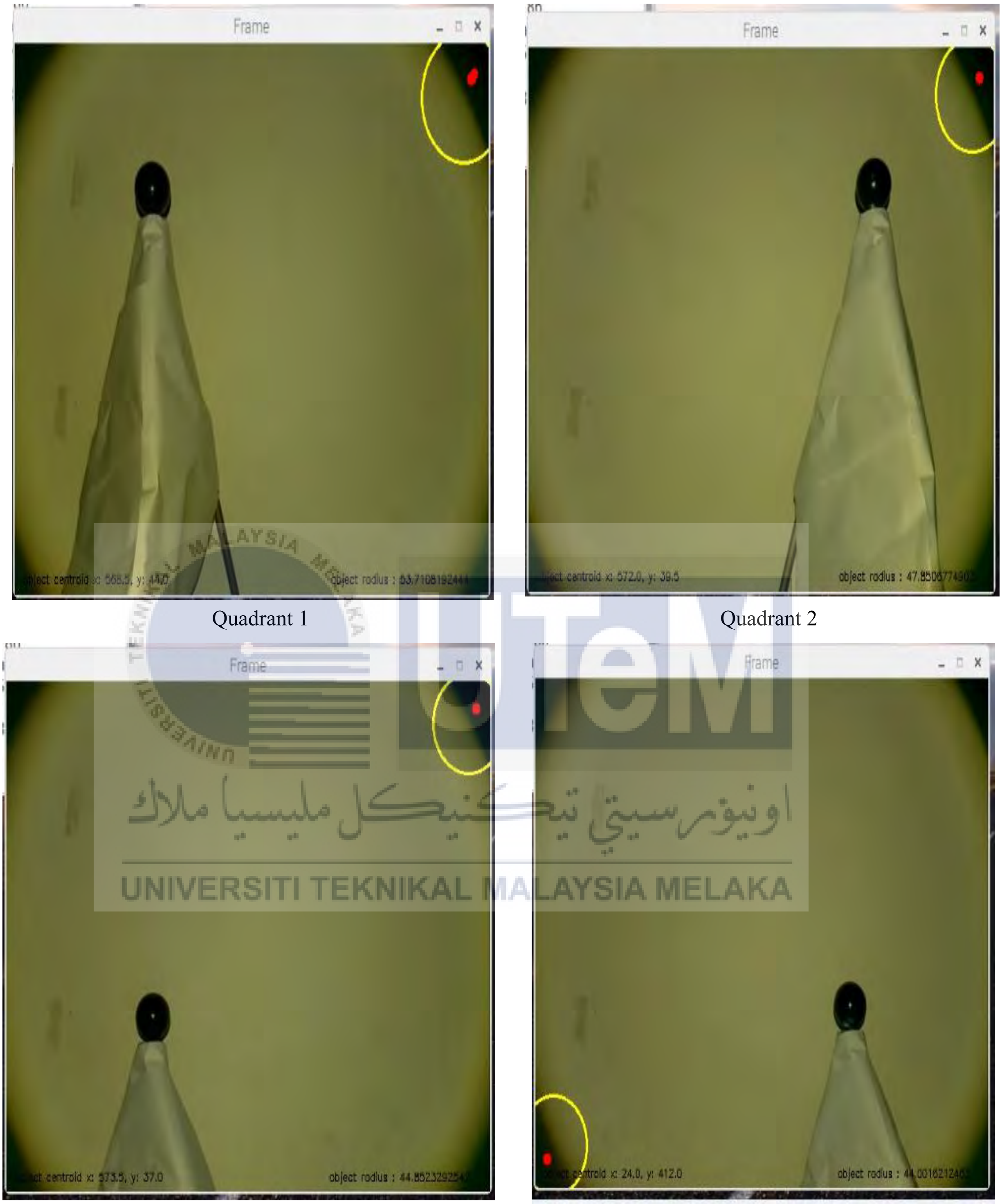
```
ject centroid x: 574.5, y: 36.5        object radius : 43.8506698608
```

Quadrant 3

```
ject centroid x: 572.0, y: 39.5        object radius : 47.8506774902
```

Quadrant 4

Figure 4.8: Error for 4.0 cm ball at distance 100 cm from the camera



```
object centroid x: 568.5, y: 44.0          object radius : 53.7108192444
```

Quadrant 1

```
object centroid x: 572.0, y: 39.5          object radius : 47.8506774903
```

Quadrant 2

```
object centroid x: 573.5, y: 37.0          object radius : 44.852329254
```

Quadrant 3

```
object centroid x: 24.0, y: 412.0          object radius : 44.0016212403
```

Quadrant 4

Figure 4.9: Error for 7.2 cm ball at distance 100 cm from the camera

The error occur in figure 4.7, 4.8 and 4.9 was because the camera tend to detect the shadow which look darker and bigger than the ball.

4.1.4 TABLE OF RESULT

Condition 1 (Headlamp+2 fluoroscent lamp)

Quadrant 1

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:206.53 y:134.07	38.84
7.2	x:167.50 y:118.00	69.90
9.3	x:179.20 y:115.04	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
21.80
6.97
1.59

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixels)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:216.50 y:139.50	x:208.55 y:135.66	x:208.55 y:135.66	x:208.50 y:135.65	51.79	45.82	45.82	45.80	47.31	2.72	2.59
7.2	Yes	Yes	Yes	Yes	x:170.53 y:115.74	x:170.61 y:115.79	x:170.97 y:115.97	x:170.81 y:115.81	74.83	74.75	74.84	74.65	74.77	0.09	0.08
9.3	Yes	Yes	Yes	Yes	x:184.77 y:111.28	x:184.41 y:111.41	x:184.60 y:111.44	x:184.60 y:111.44	91.74	91.80	91.68	91.68	91.73	0.06	0.05

Table 4.1: Table of results for radius of marker in quadrant 1 at 30 cm in condition 1

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:153.00 y:184.00	23.31
7.2	x:161.00 y:170.00	41.96
9.3	x:154.00 y:159.50	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
10.60
8.82
9.34

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:151.50 y:185.50	x:152.50 y:185.00	x:154.17 y:184.42	x:154.17 y:184.42	26.51	25.57	25.51	25.51	25.78	0.49	0.43
7.2	Yes	Yes	Yes	Yes	x:160.30 y:169.70	x:160.57 y:169.80	x:160.81 y:169.99	x:161.00 y:169.50	45.48	45.78	45.70	45.67	45.66	0.13	0.11
9.3	Yes	Yes	Yes	Yes	x:154.73 y:159.09	x:154.73 y:159.09	x:154.85 y:159.05	x:155.05 y:158.98	59.02	59.02	59.00	58.98	59.26	0.30	0.26

Table 4.2: Table of results for radius of marker in quadrant 1 at 50 cm in condition 1

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:153.33 y:155.67	16.64
7.2	x:151.00 y:152.00	29.95
9.3	x:149.500 y:146.500	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
8.17
4.34
2.07

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:155.01 y:156.00	x:155.01 y:156.01	x:155.01 y:156.01	x:155.01 y:156.01	18.00	18.00	18.00	18.00	18.00	0	0
7.2	Yes	Yes	Yes	Yes	x:151.31 y:152.31	x:151.31` y:152.31	x:150.90 y:151.90	x:151.31 y:152.31	31.38	31.38	30.86	31.38	31.25	0.26	0.23
9.3	Yes	Yes	Yes	Yes	x:150.42 y:144.42	x:150.19 y:144.50	x:150.30 y:144.57	x:150.42 y:144.42	39.50	39.53	39.42	39.50	39.49	0.05	0.04

Table 4.3: Table of results for radius of marker in quadrant 1 at 70 cm in condition 1

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:124.00 y:135.50	11.65
7.2	x:118.50 y:122.50	20.97
9.3	x:145.00 y:108.00	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
2.58
5.53
4.91

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:124.00 y:130.50	x:124.00 y:130.00	x:124.00 y:130.50	x:124.00 y:130.50	12.02	11.74	12.02	12.02	11.95	0.14	0.12
7.2	Yes	Yes	Yes	Yes	x:118.50 y:121.50	x:118.45 y:122.86	x:118.00 y:123.50	x:118.00 y:123.00	22.15	22.22	22.12	22.01	22.13	0.09	0.08
9.3	Yes	Yes	Yes	Yes	x:144.75 y:107.75	x:144.75 y:107.75	x:144.75 y:107.75	x:144.75 y:107.75	28.41	28.41	28.41	28.41	28.41	0	0

Table 4.4: Table of results for radius of marker in quadrant 1 at 100 cm in condition 1

Quadrant 2

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:448.36 y:130.35	38.84
7.2	x:416.74 y:103.41	69.90
9.3	x:408.58 y:123.93	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
35.94
6.09
4.61

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:448.00 y:157.50	x:448.50 y:158.00	x:448.50 y:158.00	x:448.00 y:157.50	52.80	52.80	52.80	52.80	52.80	0	0
7.2	Yes	Yes	Yes	Yes	x:420.39 y:114.51	x:420.39 y:114.52	x:420.45 y:114.13	x:419.24 y:112.24	75.19	75.19	74.30	71.95	74.16	1.53	1.33
9.3	Yes	Yes	Yes	Yes	x:409.48 y:125.20	x:409.73 y:125.23	x:409.48 y:125.21	x:409.60 y:125.27	94.44	94.50	94.45	94.42	94.45	0.03	0.03

Table 4.5: Table of results for radius of marker in quadrant 2 at 30 cm in condition 1

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:471.42 y:163.35	23.31
7.2	x:483.30 y:144.78	41.96
9.3	x:470.40 y:138.43	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
14.80
15.13
7.43

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:471..45 y:164.82	x:471.66 y:164.87	x:471.45 y:164.82	x:471.59 y:164.79	26.72	26.79	26.72	26.82	26.76	0.05	0.04
7.2	Yes	Yes	Yes	Yes	x:485.29 y:147.56	x:488.76 y:147.62	x:488.67 y:147.00	x:487.77 y:147.22	48.55	48.67	47.88	48.12	48.31	0.37	0.32
9.3	Yes	Yes	Yes	Yes	x:481.62 y:144.06	x:480.00 y:139.00	x:481.61 y:138.00	x:482.30 y:138.53	61.63	57.00	57.28	57.02	58.23	2.27	1.96

Table 4.6: Table of results for radius of marker in quadrant 2 at 50 cm in condition 1

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:445.00 y:132.59	16.64
7.2	x:447.60 y:123.47	29.95
9.3	x:445.78 y:116.61	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
3.67
4.57
2.53

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:445.68 y:132.93	x:445.55 y:133.06	x:445.68 y:132.93	x:445.61 y:133.12	17.26	17.19	17.26	17.27	17.25	0.04	0.03
7.2	Yes	Yes	Yes	Yes	x:447.93 y:122.53	x:447.54 y:121.73	x:448.05 y:122.56	x:448.29 y:122.69	31.51	30.66	31.55	31.57	31.32	0.44	0.38
9.3	Yes	Yes	Yes	Yes	x:446.30 y:116.10	x:446.30 y:116.10	x:446.34 y:116.13	x:446.34 y:116.13	39.67	39.67	39.66	39.66	39.67	0.01	0.01

Table 4.7: Table of results for radius of marker in quadrant 2 at 70 cm in condition 1

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:462.21 y:103.32	11.65
7.2	x:441.53 y:98.24	20.97
9.3	x:440.00 y:91.00	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
3.69
5.01
2.58

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:463.00 y:103.50	x:463.00 y:103.50	x:462.73 y:103.50	x:461.88 y:102.40	12.15	12.15	11.99	12.04	12.08	0.08	0.07
7.2	Yes	Yes	Yes	Yes	x:443.22 y:98.26	x:443.42 y:97.80	x:444.50 y:99.00	x:444.75 y:98.76	21.95	22.13	22.00	22.00	22.02	0.08	0.07
9.3	Yes	Yes	Yes	Yes	x:442.02 y:91.96	x:442.02 y:91.96	x:442.02 y:91.96	x:441.93 y:91.94	27.78	27.78	27.78	27.76	27.78	0.01	0.01

Table 4.8: Table of results for radius of marker in quadrant 2 at 100 cm in condition 1

Quadrant 3

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:167.28 y:302.45	38.84
7.2	x:164.49 y:272.27	69.90
9.3	x:176.90 y:313.33	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
13.59
24.01
10.81

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:167.50 y:300.50	x:166.00 y:302.00	x:166.00 y:302.00	x:166.00 y:302.00	44.19	44.09	44.09	44.09	44.12	0.04	0.04
7.2	Yes	Yes	Yes	Yes	x:199.21 y:283.83	x:199.21 y:283.83	x:199.21 y:283.77	x:199.21 y:283.77	86.66	86.66	86.70	86.70	86.68	0.02	0.02
9.3	Yes	Yes	Yes	Yes	x:170.39 y:325.76	x:169.48 y:325.73	x:169.01 y:325.99	x:167.12 y:326.01	99.95	100.02	100.10	100.12	100.05	0.08	0.07

Table 4.9: Table of results for radius of marker in quadrant 3 at 30 cm in condition 1

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:173.15 y:328.50	23.31
7.2	x:161.67 y:305.95	41.96
9.3	x:154.88 y:314.48	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
13.38
20.75
10.68

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:173.50 y:330.31	x:173.50 y:330.50	x:173.07 y:330.82	x:173.92 y:329.85	26.32	26.71	26.78	25.91	26.43	0.40	0.35
7.2	Yes	Yes	Yes	Yes	x:164.19 y:311.70	x:162.69 y:312.17	x:164.04 y:312.50	x:164.06 y:312.59	50.71	50.41	50.81	50.73	50.67	0.18	0.15
9.3	Yes	Yes	Yes	Yes	x:156.28 y:318.41	x:156.29 y:318.89	x:156.79 y:319.03	x:158.45 y:318.87	60.18	59.73	59.80	60.24	59.99	0.26	0.22

Table 4.10: Table of results for radius of marker in quadrant 3 at 50 cm in condition 1

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:165.58 y:318.52	16.64
7.2	x:174.52 y:304.78	29.95
9.3	x:152.73 y:299.97	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
2.16
2.60
1.45

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:165.97 y:317.95	x:165.80 y:318.07	x:165.40 y:317.73	x:165.64 y:318.30	16.95	16.80	17.25	17.00	17.00	0.19	0.16
7.2	Yes	Yes	Yes	Yes	x:174.63 y:305.14	x:174.63 y:305.14	x:175.24 y:305.65	x:175.00 y:306.00	30.39	30.39	31.01	31.14	30.73	0.40	0.35
9.3	Yes	Yes	Yes	Yes	x:152.50 y:300.50	x:152.60 y:300.69	x:151.83 y:300.00	x:153.24 y:300.80	37.88	37.89	38.78	37.98	38.13	0.43	0.38

Table 4.11: Table of results for radius of marker in quadrant 3 at 70 cm in condition 1

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:146.30 y:348.74	11.65
7.2	x:147.62 y:340.27	20.97
9.3	x:155.08 y:334.59	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
0.87
4.20
1.92

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:147.00 y:349.00	x:146.50 y:349.00	x:147.00 y:349.00	x:146.27 y:348.66	11.52	11.57	11.52	11.57	11.55	0.03	0.03
7.2	Yes	Yes	Yes	Yes	x:147.85 y:340.32	x:147.99 y:340.26	x:147.99 y:340.26	x:147.99 y:340.26	21.75	21.88	21.88	21.88	21.85	0.07	0.06
9.3	Yes	Yes	Yes	Yes	x:154.19 y:333.65	x:154.21 y:333.66	x:155.53 y:334.53	x:155.65 y:334.56	27.62	27.62	27.56	27.61	27.60	0.03	0.03

Table 4.12: Table of results for radius of marker in quadrant 3 at 100 cm in condition 1

Quadrant 4

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:431.00 y:273.00	38.84
7.2	x:389.50 y:275.15	69.90
9.3	x:410.55 y:288.77	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
27.68
19.84
8.55

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixels)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:424.00 y:286.00	x:424.50 y:286.50	x:424.08 y:286.04	x:424.12 y:286.09	49.61	49.57	49.58	49.58	49.59	0.02	0.02
7.2	Yes	Yes	Yes	Yes	x:390.52 y:298.96	x:390.01 y:298.78	x:390.30 y:299.02	x:390.32 y:299.12	83.75	83.62	83.89	83.81	83.77	0.11	0.10
9.3	Yes	Yes	Yes	Yes	x:408.79 y:299.00	x:408.33 y:299.20	x:406.33 y:299.20	x:406.33 y:299.20	97.85	98.06	98.06	98.06	98.01	0.11	0.10

Table 4.13: Table of results for radius of marker in quadrant 4 at 30 cm in condition 1

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:467.00 y:309.65	23.31
7.2	x:458.84 y:290.56	41.96
9.3	x:464.90 y:279.27	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
10.55
9.13
3.43

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:467.14 y:308.77	x:467.22 y:308.78	x:467.22 y:308.78	x:467.05 y:308.95	25.71	25.74	25.74	25.89	25.77	0.08	0.07
7.2	Yes	Yes	Yes	Yes	x:459.52 y:290.26	x:459.68 y:290.68	x:459.68 y:290.42	x:459.52 y:290.26	45.62	46.08	45.82	45.62	45.79	0.22	0.19
9.3	Yes	Yes	Yes	Yes	x:466.52 y:278.11	x:468.00 y:277.50	x:468.50 y:279.00	x:469.50 y:279.00	56.55	55.91	55.88	55.88	56.06	0.33	0.29

Table 4.14: Table of results for radius of marker in quadrant 4 at 50 cm in condition 1

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:465.50 y:297.00	16.64
7.2	x:438.00 y:287.00	29.95
9.3	x:442.50 y:278.00	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
1.26
4.71
0.31

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:466.00 y:297.50	x:465.61 y:298.18	x:466.00 y:297.50	x:466.00 y:297.50	16.46	16.34	16.46	16.46	16.43	0.06	0.05
7.2	Yes	Yes	Yes	Yes	x:439.33 y:286.01	x:439.50 y:285.50	x:439.50 y:286.50	x:439.50 y:286.50	31.40	31.35	31.35	31.35	31.36	0.03	0.03
9.3	Yes	Yes	Yes	Yes	x:444.00 y:279.50	x:444.00 y:279.50	x:444.28 y:279.81	x:445.00 y:278.00	38.69	38.89	38.87	38.79	38.81	0.09	0.08

Table 4.15: Table of results for radius of marker in quadrant 4 at 70 cm in condition 1

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:467.50 y:327.00	11.65
7.2	x:429.44 y:320.54	20.97
9.3	x:426.64 y:316.51	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
2.66
1.10
0.08

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:468.10 y:327.10	x:468.24 y:327.24	x:468.24 y:327.24	x:468.10 y:327.10	11.87	12.05	12.05	11.87	11.96	0.10	0.09
7.2	Yes	Yes	Yes	Yes	x:429.81 y:320.32	x:429.81 y:320.32	x:430.16 y:320.45	x:430.00 y:320.00	21.13	21.13	21.31	21.23	21.20	0.09	0.08
9.3	Yes	Yes	Yes	Yes	x:427.68 y:315.68	x:427.81 y:315.69	x:427.92 y:315.13	x:427.64 y:315.84	26.60	26.75	27.21	26.88	26.86	0.26	0.23

Table 4.16: Table of results for radius of marker in quadrant 4 at 100 cm in condition 1

Condition 2 (Headlamp)

Quadrant 1

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:95.00 y:104.00	38.84
7.2	x:204.25 y:192.37	69.90
9.3	x:203.24 y:178.23	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
11.92
23.61
15.49

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:93.77 y:104.76	x:93.76 y:104.76	x:93.53 y:104.86	x:93.76 y:104.76	43.42	43.42	43.22	43.42	43.47	0.15	0.13
7.2	Yes	Yes	Yes	Yes	x:199.94 y:201.48	x:199.79 y:201.35	x:199.79 y:201.35	x:199.16 y:201.63	86.28	86.32	86.33	86.65	86.40	0.17	0.16
9.3	Yes	Yes	Yes	Yes	x:198.86 y:185.62	x:199.04 y:185.98	x:198.93 y:185.49	x:199.01 y:185.88	103.96	104.20	104.64	104.30	104.28	0.28	0.24

Table 4.17: Table of results for radius of marker in quadrant 1 at 30 cm in condition 2

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:147.50 y:152.00	23.31
7.2	x:141.52 y:134.89	41.96
9.3	x:141.23 y:126.99	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
33.76
30.65
17.71

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:152.53 y:154.81	x:152.23 y:155.29	x:151.50 y:155.50	x:151.50 y:155.50	31.07	30.95	31.35	31.35	31.18	0.20	0.18
7.2	Yes	Yes	Yes	Yes	x:146.24 y:141.54	x:145.84 y:141.78	x:145.39 y:141.69	x:145.39 y:141.70	54.81	54.95	54.76	54.76	54.82	0.09	0.08
9.3	Yes	Yes	Yes	Yes	x:137.31 y:132.91	x:136.70 y:133.19	x:136.39 y:133.20	x:141.46 y:133.60	62.97	63.40	63.28	65.55	63.80	1.18	1.02

Table 4.18: Table of results for radius of marker in quadrant 1 at 50 cm in condition 2

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:132.00 y:124.50	16.64
7.2	x:133.00 y:112.50	29.95
9.3	x:133.50 y:104.00	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
11.09
9.33

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	Yes	Yes	Yes	Yes	x:133.53 y:113.31	x:133.74 y:113.56	x:134.03 y:113.88	x:130.90 y:115.10	32.59	32.87	33.45	34.15	33.27	0.69	0.60
9.3	Yes	Yes	Yes	Yes	x:130.41 y:107.82	x:130.38 y:107.75	x:130.41 y:107.82	x:130.41 y:107.82	42.29	42.34	42.29	42.29	42.30	0.03	0.03

Table 4.19: Table of results for radius of marker in quadrant 1 at 70 cm in condition 2

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:173.25 y:129.66	11.65
7.2	x:175.50 y:120.00	20.97
9.3	x:169.68 y:113.80	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
100
15.14

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
9.3	Yes	Yes	Yes	Yes	x:169.54 y:116.56	x:168.80 y:116.39	x:167.14 y:116.01	x:166.88 y:115.96	31.59	31.36	30.91	30.85	31.18	0.36	0.31

Table 4.20: Table of results for radius of marker in quadrant 1 at 100 cm in condition 2

Quadrant 2

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:488.11 y:96.76	38.84
7.2	x:456.87 y:182.78	69.90
9.3	x:459.60 y:173.60	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
59.50
28.10
12.98

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:477.00 y:110.00	x:477.00 y:110.00	x:477.00 y:110.00	x:477.00 y:110.00	61.95	61.95	61.95	61.95	61.95	0	0
7.2	Yes	Yes	Yes	Yes	x:454.00 y:196.00	x:449.50 y:192.50	x:454.00 y:196.00	x:454.00 y:196.50	89.28	89.93	89.28	89.68	89.54	0.32	0.28
9.3	Yes	Yes	Yes	Yes	x:453.59 y:182.76	x:453.59 y:182.76	x:452.42 y:182.44	x:453.72 y:182.85	102.09	102.09	101.77	102.09	102.01	0.16	0.14

Table 4.21: Table of results for radius of marker in quadrant 2 at 30 cm in condition 2

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:425.32 y:130.70	23.31
7.2	x:419.90 y:115.19	41.96
9.3	x:420.20 y:108.86	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
56.37
40.80
27.31

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:421.00 y:140.00	x:421.00 y:140.00	x:421.00 y:140.00	x:421.00 y:140.00	36.45	36.45	36.45	36.45	36.45	0	0
7.2	Yes	Yes	Yes	Yes	x:415.17 y:124.30	x:416.04 y:125.11	x:415.11 y:124.24	x:416.50 y:125.50	59.10	59.05	59.10	59.07	59.08	0.02	0.02
9.3	Yes	Yes	Yes	Yes	x:416.00 y:115.50	x:416.00 y:115.50	x:418.40 y:117.69	x:417.00 y:116.00	69.03	69.03	68.71	69.22	69.00	0.21	0.18

Table 4.22: Table of results for radius of marker in quadrant 2 at 50 cm in condition 2

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:450.43 y:121.78	16.64
7.2	x:446.88 y:110.95	29.95
9.3	x:446.06 y:103.72	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
16.99
9.51

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	Yes	Yes	Yes	Yes	x:444.87 y:115.04	x:444.87 y:115.04	x:444.87 y:115.04	x:444.03 y:114.78	35.11	35.11	35.11	34.81	35.04	0.15	0.12
9.3	Yes	Yes	Yes	Yes	x:446.81 y:104.41	x:447.50 y:104.50	x:440.00 y:107.00	x:440.00 y:107.00	39.17	39.39	45.37	45.53	42.37	3.56	2.88

Table 4.23: Table of results for radius of marker in quadrant 2 at 70 cm in condition 2

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:445.30 y:122.33	11.65
7.2	x:445.39 y:114.05	20.97
9.3	x:438.85 y:108.01	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
100
21.34

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
9.3	Yes	Yes	Yes	Yes	x:438.00 y:111.00	x:438.00 y:111.00	x:438.00 y:111.00	x:438.00 y:111.00	32.86	32.86	32.86	32.86	32.86	0	0

Table 4.24: Table of results for radius of marker in quadrant 2 at 100 cm in condition 2

Quadrant 3

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:110.60 y:314.98	38.84
7.2	x:119.13 y:284.91	69.90
9.3	x:96.87 y:272.41	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
22.01
18.91
5.15

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:109.08 y:317.26	x:113.96 y:319.17	x:109.08 y:317.26	x:114.40 y:318.84	46.01	48.88	46.01	48.66	47.39	1.60	1.38
7.2	Yes	Yes	Yes	Yes	x:119.94 y:293.66	x:120.08 y:293.77	x:120.08 y:293.77	x:119.93 y:293.59	83.12	83.10	83.10	83.16	83.12	0.03	0.02
9.3	Yes	Yes	Yes	Yes	x:107.79 y:276.79	x:107.31 y:277.02	x:104.74 y:277.80	x:108.01 y:277.36	94.75	94.75	95.20	95.05	94.94	0.23	0.19

Table 4.25: Table of results for radius of marker in quadrant 3 at 30 cm in condition 2

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:191.50 y:303.50	23.31
7.2	x:176.16 y:287.50	41.96
9.3	x:188.06 y:276.11	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
19.99
19.40
12.90

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:190.89 y:306.51	x:190.89 y:306.51	x:190.89 y:306.51	x:190.89 y:306.51	27.97	27.97	27.97	27.97	27.97	0	0
7.2	Yes	Yes	Yes	Yes	x:188.14 y:288.76	x:189.02 y:289.10	x:189.02 y:289.10	x:189.02 y:289.10	49.74	50.22	50.22	50.22	50.10	0.24	0.21
9.3	Yes	Yes	Yes	Yes	x:187.12 y:279.14	x:186.72 y:279.31	x:187.12 y:279.09	x:187.12 y:279.14	61.14	61.28	61.19	61.14	61.19	0.07	0.04

Table 4.26: Table of results for radius of marker in quadrant 3 at 50 cm in condition 2

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:163.17 y:307.22	16.64
7.2	x:158.93 y:295.07	29.95
9.3	x:164.18 y:288.82	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
13.19
5.45

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	Yes	Yes	Yes	Yes	x:157.58 y:297.10	x:157.59 y:297.22	x:157.59 y:297.22	x:157.59 y:297.22	33.97	33.87	33.87	33.87	33.90	0.05	0.04
9.3	Yes	Yes	Yes	Yes	x:165.95 y:290.79	x:165.96 y:290.79	x:166.33 y:290.96	x:166.96 y:290.79	40.79	40.79	40.84	40.79	40.80	0.03	0.02

Table 4.27: Table of results for radius of marker in quadrant 3 at 70 cm in condition 2

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:179.62 y:305.91	11.65
7.2	x:182.65 y:296.36	20.97
9.3	x:177.56 y:291.35	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
100
3.29

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
9.3	Yes	Yes	Yes	Yes	x:177.98 y:292.01	x:177.98 y:292.07	x:177.25 y:291.96	x:177.14 y:291.95	28.12	28.06	27.87	27.84	27.97	0.14	0.12

Table 4.28: Table of results for radius of marker in quadrant 3 at 100 cm in condition 2

Quadrant 4

Distance from camera: 30cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:469.00 y:307.00	38.84
7.2	x:470.40 y:275.92	69.90
9.3	x:457.11 y:262.95	90.29

Error (%) between the mean value and the radius of marker in camera space (pixel)
26.21
16.88
6.14

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:46.350 y:307.00	x:463.50 y:307.00	x:463.50 y:307.00	x:464.50 y:308.50	49.03	49.03	49.03	48.99	49.02	0.02	0.02
7.2	Yes	Yes	Yes	Yes	x:478.00 y:282.50	x:479.00 y:281.50	x:480.00 y:280.00	x:484.00 y:282.50	81.44	81.44	81.78	82.12	81.70	0.33	0.30
9.3	Yes	Yes	Yes	Yes	x:454.69 y:267.71	x:455.57 y:267.94	x:456.24 y:268.15	x:456.24 y:268.15	95.51	95.80	96.00	96.00	95.83	0.23	0.20

Table 4.29: Table of results for radius of marker in quadrant 4 at 30 cm in condition 2

Distance from camera: 50cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:440.57 y:294.09	23.31
7.2	x:440.21 y:276.14	41.96
9.3	x:437.87 y:266.76	54.20

Error (%) between the mean value and the radius of marker in camera space (pixel)
33.03
25.81
17.21

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	Yes	Yes	Yes	Yes	x:438.50 y:298.50	x:438.50 y:298.50	x:438.50 y:298.50	x:438.50 y:298.50	31.01	31.01	31.01	31.01	31.01	0	0
7.2	Yes	Yes	Yes	Yes	x:436.50 y:279.00	x:438.50 y:281.00	x:438.06 y:280.99	x:438.50 y:281.00	52.95	52.81	52.57	52.81	52.79	0.17	0.14
9.3	Yes	Yes	Yes	Yes	x:437.86 y:269.94	x:438.31 y:270.31	x:437.64 y:269.67	x:438.34 y:269.87	63.43	63.39	63.52	63.77	63.53	0.17	0.14

Table 4.30: Table of results for radius of marker in quadrant 4 at 50 cm in condition 2

Distance from camera: 70cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:433.81 y:300.89	16.64
7.2	x:431.47 y:287.70	29.95
9.3	x:437.00 y:280.50	38.69

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
11.35
5.43

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	Yes	Yes	Yes	Yes	x:431.69 y:289.28	x:431.69 y:289.32	x:431.69 y:289.28	x:431.69 y:289.28	33.36	33.33	33.36	33.36	33.35	0.02	0.02
9.3	Yes	Yes	Yes	Yes	x:441.78 y:283.74	x:428.76 y:282.28	x:428.76 y:282.28	x:428.50 y:282.22	40.09	41.05	41.05	40.95	40.79	0.41	0.40

Table 4.31: Table of results for radius of marker in quadrant 4 at 70 cm in condition 2

Distance from camera: 100cm

Original value:

Marker size (cm)	Centroid	Radius of marker in camera space (pixel)
4.0	x:418.03 y:298.69	11.65
7.2	x:416.00 y:289.50	20.97
9.3	x:421.75 y:284.63	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
100
100
4.17

Experimental value

Marker size (cm)	Detection Yes / No				Centroid				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
4.0	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
7.2	No	No	No	No	-	-	-	-	-	-	-	-	-	-	-
9.3	Yes	Yes	Yes	Yes	x:416.52 y:285.08	x:416.52 y:285.08	x:417.63 y:284.59	x:416.74 y:285.19	28.28	28.28	27.99	28.27	28.21	0.14	0.12

Table 4.32: Table of results for radius of marker in quadrant 4 at 100 cm in condition 2

Table of result for distance of marker from camera

Condition 1 (headlamp+2 fluorescent lamp)

Quadrant 1

4.0 cm ball in

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
21.80
10.60
8.17
2.58

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	51.79	45.82	45.82	45.80	47.31	2.72	2.59
50	Yes	Yes	Yes	Yes	26.51	25.57	25.51	25.51	25.78	0.49	0.43
70	Yes	Yes	Yes	Yes	18.00	18.00	18.00	18.00	18.00	0	0
100	Yes	Yes	Yes	Yes	12.02	11.74	12.02	12.02	11.95	0.14	0.12

Table 4.33: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 1 in condition 1

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
6.97
8.82
4.34
5.53

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	74.83	74.75	74.84	74.65	74.77	0.09	0.08
50	Yes	Yes	Yes	Yes	45.48	45.78	45.70	45.67	45.66	0.13	0.11
70	Yes	Yes	Yes	Yes	31.38	31.38	30.86	31.38	31.25	0.26	0.23
100	Yes	Yes	Yes	Yes	22.15	22.22	22.12	22.01	22.13	0.09	0.08

Table 4.34: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 1 in condition 1

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
1.59
9.34
2.07
4.91

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	91.74	91.80	91.68	91.68	91.73	0.06	0.05
50	Yes	Yes	Yes	Yes	59.02	59.02	59.00	58.98	59.26	0.30	0.26
70	Yes	Yes	Yes	Yes	39.50	39.53	39.42	39.50	39.49	0.05	0.04
100	Yes	Yes	Yes	Yes	28.41	28.41	28.41	28.41	28.41	0	0

Table 4.35: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 1 in condition 1

Quadrant 2

4.0 cm ball in

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
35.94
14.80
3.67
3.69

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	52.80	52.80	52.80	52.80	52.80	0	0
50	Yes	Yes	Yes	Yes	26.72	26.79	26.72	26.82	26.76	0.05	0.04
70	Yes	Yes	Yes	Yes	17.26	17.19	17.26	17.27	17.25	0.04	0.03
100	Yes	Yes	Yes	Yes	12.15	12.15	11.99	12.04	12.08	0.08	0.07

Table 4.36: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 2 in condition 1

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
6.09
15.13
4.57
5.01

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	75.19	75.19	74.30	71.95	74.16	1.53	1.33
50	Yes	Yes	Yes	Yes	48.55	48.67	47.88	48.12	48.31	0.37	0.32
70	Yes	Yes	Yes	Yes	31.51	30.66	31.55	31.57	31.32	0.44	0.38
100	Yes	Yes	Yes	Yes	21.95	22.13	22.00	22.00	22.02	0.08	0.07

Table 4.37: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 2 in condition 1

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
4.61
7.43
2.53
2.58

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	94.44	94.50	94.45	94.42	94.45	0.03	0.03
50	Yes	Yes	Yes	Yes	61.63	57.00	57.28	57.02	58.23	2.27	1.96
70	Yes	Yes	Yes	Yes	39.67	39.67	39.66	39.66	39.67	0.01	0.01
100	Yes	Yes	Yes	Yes	27.78	27.78	27.78	27.76	27.78	0.01	0.01

Table 4.38: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 2 in condition 1

Quadrant 3

4.0 cm ball in

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
13.59
13.38
2.16
0.87

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	44.19	44.09	44.09	44.09	44.12	0.04	0.04
50	Yes	Yes	Yes	Yes	26.32	26.71	26.78	25.91	26.43	0.40	0.35
70	Yes	Yes	Yes	Yes	16.95	16.80	17.25	17.00	17.00	0.19	0.16
100	Yes	Yes	Yes	Yes	11.52	11.57	11.52	11.57	11.55	0.03	0.03

Table 4.39: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 3 in condition 1

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
24.01
20.75
2.60
4.20

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	86.66	86.66	86.70	86.70	86.68	0.02	0.02
50	Yes	Yes	Yes	Yes	50.71	50.41	50.81	50.73	50.67	0.18	0.15
70	Yes	Yes	Yes	Yes	30.39	30.39	31.01	31.14	30.73	0.40	0.35
100	Yes	Yes	Yes	Yes	21.75	21.88	21.88	21.88	21.85	0.07	0.06

Table 4.40: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 3 in condition 1

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
10.81
10.68
1.45
1.92

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	99.95	100.02	100.10	100.12	100.05	0.08	0.07
50	Yes	Yes	Yes	Yes	60.18	59.73	59.80	60.24	59.99	0.26	0.22
70	Yes	Yes	Yes	Yes	37.88	37.89	38.78	37.98	38.13	0.43	0.38
100	Yes	Yes	Yes	Yes	27.62	27.62	27.56	27.61	27.60	0.03	0.03

Table 4.41: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 3 in condition 1

Quadrant 4

4.0 cm ball in

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
27.68
10.55
1.26
2.66

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	49.61	49.57	49.58	49.58	49.59	0.02	0.02
50	Yes	Yes	Yes	Yes	25.71	25.74	25.74	25.89	25.77	0.08	0.07
70	Yes	Yes	Yes	Yes	16.46	16.34	16.46	16.46	16.43	0.06	0.05
100	Yes	Yes	Yes	Yes	11.87	12.05	12.05	11.87	11.96	0.10	0.09

Table 4.42: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 4 in condition 1

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
19.84
9.13
4.71
1.10

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	83.75	83.62	83.89	83.81	83.77	0.11	0.10
50	Yes	Yes	Yes	Yes	45.62	46.08	45.82	45.62	45.79	0.22	0.19
70	Yes	Yes	Yes	Yes	31.40	31.35	31.35	31.35	31.36	0.03	0.03
100	Yes	Yes	Yes	Yes	21.13	21.13	21.31	21.23	21.20	0.09	0.08

Table 4.43: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 4 in condition 1

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
8.55
3.43
0.31
0.08

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	97.85	98.06	98.06	98.06	98.01	0.11	0.10
50	Yes	Yes	Yes	Yes	56.55	55.91	55.88	55.88	56.06	0.33	0.29
70	Yes	Yes	Yes	Yes	38.69	38.89	38.87	38.79	38.81	0.09	0.08
100	Yes	Yes	Yes	Yes	26.60	26.75	27.21	26.88	26.86	0.26	0.23

Table 4.44: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 4 in condition 1

Condition 2 (headlamp)

Quadrant 1

4.0 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
11.92
33.76
100
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	43.42	43.42	43.22	43.42	43.47	0.15	0.13
50	Yes	Yes	Yes	Yes	31.07	30.95	31.35	31.35	31.18	0.20	0.18
70	No	No	No	No	-	-	-	-	-	-	-
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.45: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 1 in condition 2

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
23.61
30.65
11.09
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	86.28	86.32	86.33	86.65	86.40	0.17	0.16
50	Yes	Yes	Yes	Yes	54.81	54.95	54.76	54.76	54.82	0.09	0.08
70	Yes	Yes	Yes	Yes	32.59	32.87	33.45	34.15	33.27	0.69	0.60
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.46: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 1 in condition 2

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
15.49
17.71
9.33
15.14

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	103.96	104.20	104.64	104.30	104.28	0.28	0.24
50	Yes	Yes	Yes	Yes	62.97	63.40	63.28	65.55	63.80	1.18	1.02
70	Yes	Yes	Yes	Yes	42.29	42.34	42.29	42.29	42.30	0.03	0.03
100	Yes	Yes	Yes	Yes	31.59	31.36	30.91	30.85	31.18	0.36	0.31

Table 4.47: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 1 in condition 2

Quadrant 2

4.0 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
59.50
56.37
100
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	61.95	61.95	61.95	61.95	61.95	0	0
50	Yes	Yes	Yes	Yes	36.45	36.45	36.45	36.45	36.45	0	0
70	No	No	No	No	-	-	-	-	-	-	-
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.48: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 2 in condition 2

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
28.10
40.80
16.99
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	89.28	89.93	89.28	89.68	89.54	0.32	0.28
50	Yes	Yes	Yes	Yes	59.10	59.05	59.10	59.07	59.08	0.02	0.02
70	Yes	Yes	Yes	Yes	35.11	35.11	35.11	34.81	35.04	0.15	0.12
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.49: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 2 in condition 2

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
12.98
27.31
9.51
21.34

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	102.09	102.09	101.77	102.09	102.01	0.16	0.14
50	Yes	Yes	Yes	Yes	69.03	69.03	68.71	69.22	69.00	0.21	0.18
70	Yes	Yes	Yes	Yes	39.17	39.39	45.37	45.53	42.37	3.56	2.88
100	Yes	Yes	Yes	Yes	32.86	32.86	32.86	32.86	32.86	0	0

Table 4.50: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 2 in condition 2

Quadrant 3

4.0 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
22.01
19.99
100
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	46.01	48.88	46.01	48.66	47.39	1.60	1.38
50	Yes	Yes	Yes	Yes	27.97	27.97	27.97	27.97	27.97	0	0
70	No	No	No	No	-	-	-	-	-	-	-
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.51: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 3 in condition 2

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
18.91
19.40
13.19
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	83.12	83.10	83.10	83.16	83.12	0.03	0.02
50	Yes	Yes	Yes	Yes	49.74	50.22	50.22	50.22	50.10	0.24	0.21
70	Yes	Yes	Yes	Yes	33.97	33.87	33.87	33.87	33.90	0.05	0.04
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.52: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 3 in condition 2

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
5.15
12.90
5.45
3.29

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	94.75	94.75	95.20	95.05	94.94	0.23	0.19
50	Yes	Yes	Yes	Yes	61.14	61.28	61.19	61.14	61.19	0.07	0.04
70	Yes	Yes	Yes	Yes	40.79	40.79	40.84	40.79	40.80	0.03	0.02
100	Yes	Yes	Yes	Yes	28.12	28.06	27.87	27.84	27.97	0.14	0.12

Table 4.53: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 3 in condition 2

Quadrant 4

4.0 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	38.84
50	23.31
70	16.64
100	11.65

Error (%) between the mean value and the radius of marker in camera space (pixel)
26.21
33.03
100
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	49.03	49.03	49.03	48.99	49.02	0.02	0.02
50	Yes	Yes	Yes	Yes	31.01	31.01	31.01	31.01	31.01	0	0
70	No	No	No	No	-	-	-	-	-	-	-
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.54: Table of result for radius of 4.0 cm ball in camera space based on the distance of marker from camera in quadrant 4 in condition 2

7.2 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	69.90
50	41.96
70	29.95
100	20.97

Error (%) between the mean value and the radius of marker in camera space (pixel)
16.88
25.81
11.35
100

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	81.44	81.44	81.78	82.12	81.70	0.33	0.30
50	Yes	Yes	Yes	Yes	52.95	52.81	52.57	52.81	52.79	0.17	0.14
70	Yes	Yes	Yes	Yes	33.36	33.33	33.36	33.36	33.35	0.02	0.02
100	No	No	No	No	-	-	-	-	-	-	-

Table 4.55: Table of result for radius of 7.2 cm ball in camera space based on the distance of marker from camera in quadrant 4 in condition 2

9.3 cm ball

Original value

Distance of marker from camera (cm)	Radius of marker in camera space (pixel)
30	90.29
50	54.20
70	38.69
100	27.08

Error (%) between the mean value and the radius of marker in camera space (pixel)
6.14
17.21
5.43
4.17

Experimental value

Distance of marker from camera (cm)	Detection Yes / No				Radius of marker in camera space (pixel)				Mean (pixel)	Standard deviation (pixel)	root-mean-square-error (RMSE)
30	Yes	Yes	Yes	Yes	95.51	95.80	96.00	96.00	95.83	0.23	0.20
50	Yes	Yes	Yes	Yes	63.43	63.39	63.52	63.77	63.53	0.17	0.14
70	Yes	Yes	Yes	Yes	40.09	41.05	41.05	40.95	40.79	0.41	0.40
100	Yes	Yes	Yes	Yes	28.28	28.28	27.99	28.27	28.21	0.14	0.12

Table 4.56: Table of result for radius of 9.3 cm ball in camera space based on the distance of marker from camera in quadrant 4 in condition 2

Table 4.57: Mean error for each quadrant in condition 1

Mean error for each quadrant % Size of object (cm)	At distance 30 cm	At distance 50 cm	At distance 70 cm	At distance 100 cm
4	24.75	12.33	3.82	2.45
7.2	14.23	13.46	4.05	3.96
9.3	6.39	7.72	1.59	2.37

Table 4.58: Mean error for each quadrant in condition 2

Mean error for each quadrant % Size of object (cm)	At distance 30 cm	At distance 50 cm	At distance 70 cm	At distance 100 cm
4.0	22.91	35.79	100	100
7.2	21.88	29.17	13.16	100
9.3	9.94	18.78	7.43	10.99

The error value in this table was obtained from the difference between the actuall size of ball and the mean size of ball capture in the camera space during the experiment for both condition.

4.2 STASTICAL VERIFICATION

Graph from the result

Condition 1

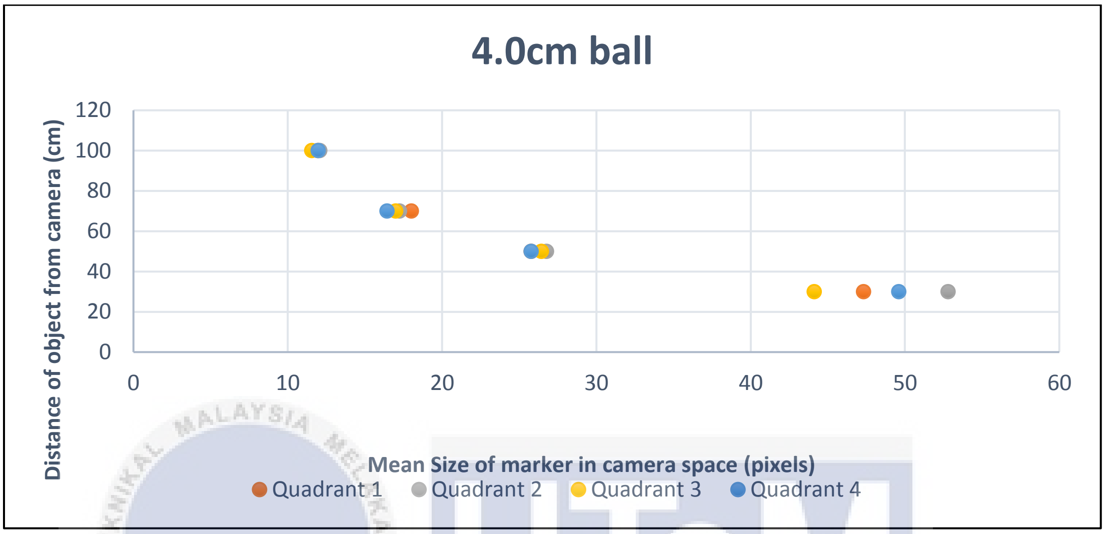


Figure 4.10: Graph of mean size of marker in camera space against distance of object from camera for 4.0 cm ball for condition 1

The point plotted in the graph shown that the object was detected at each quadrant at all distance. At the point plotted for the 4.0 cm ball in camera space at each quadrant was more concentrated at distance of 100 cm compare to 30 cm, 50 cm and 70 cm. At 30 cm distance, the point plotted in the graph was most scattered

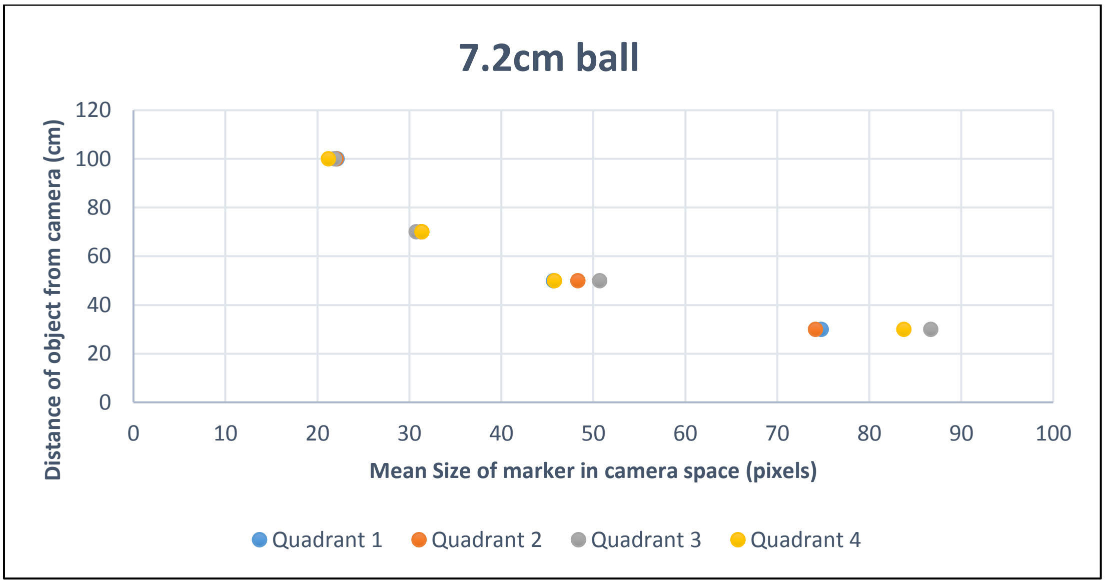


Figure 4.11: Graph of mean size of marker in camera space against distance of object from camera for 7.2 cm ball for condition 1

The point plotted in the graph shown that the object was detected at each quadrant at all distance. The point plotted for the 7.2 cm ball in camera space at each quadrant was more concentrated at distance of 100 cm compare to 30 cm, 50 cm and 70 cm. At 30 cm distance, the point plotted in the graph was most scattered

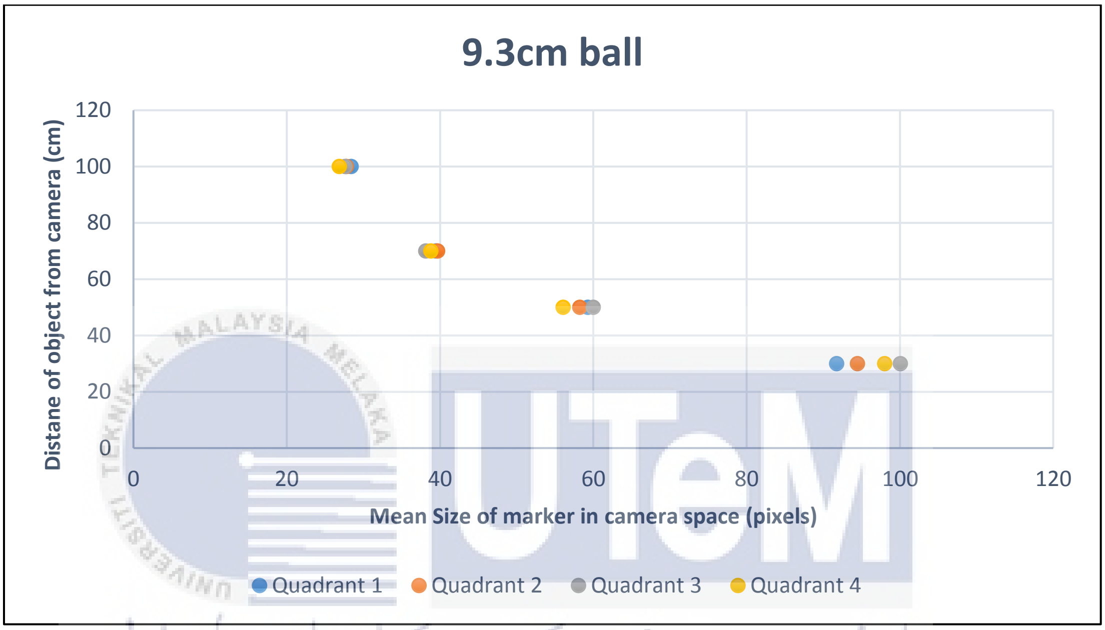


Figure 4.12: Graph of mean size of marker in camera space against distance of object from camera for 9.3 cm ball for condition 1

The point plotted in the graph shown that the object was detected at each quadrant at all distance. The point plotted for the 9.3 cm ball in camera space at each quadrant was more concentrated at distance of 100 cm compare to 30 cm, 50 cm and 70 cm. At 30 cm distance, the point plotted in the graph was most scattered

As a conclusion, there was no problem regarding to the object detection in this condition. We can conclude that furthest the distance of object to the camera, the more precise the camera in detecting the marker since the mean size of object in camera space for every quadrant does not differ much.

Condition 2

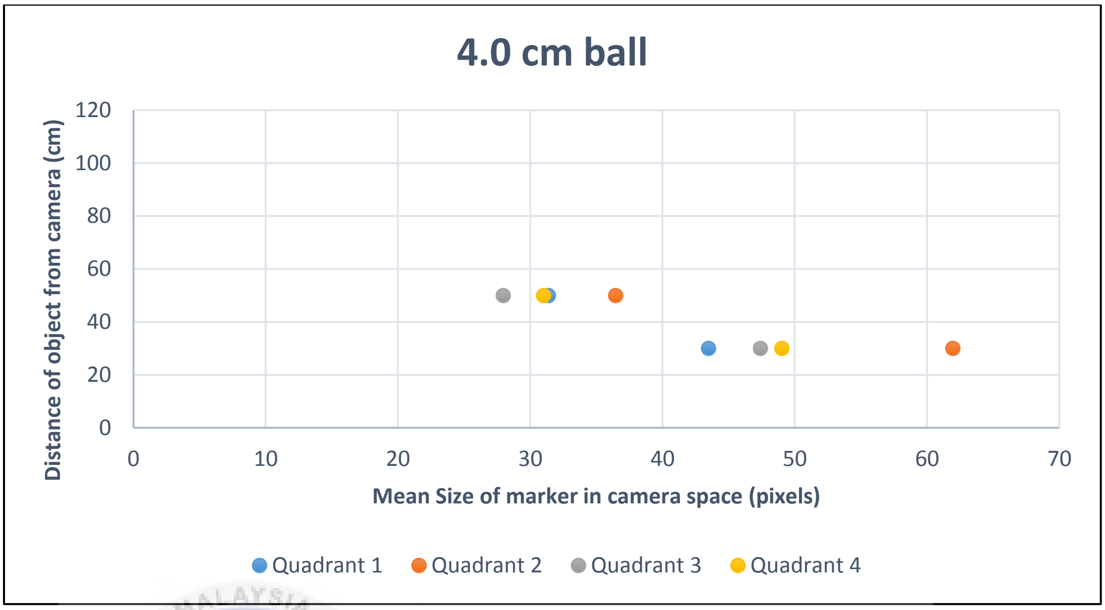


Figure 4.13: Graph of mean size of marker in camera space against distance of object from camera for 4.0 cm ball for condition 2

From the graph above, there were no object detected at 70 cm and 100 cm distance from each quadrant because the camera tend to detect the presence of shadow in camera space due to low level of lux which appear more larger compare to the size of ball. The performance also was bad compared to the detection of 4.0 cm ball at condition 1

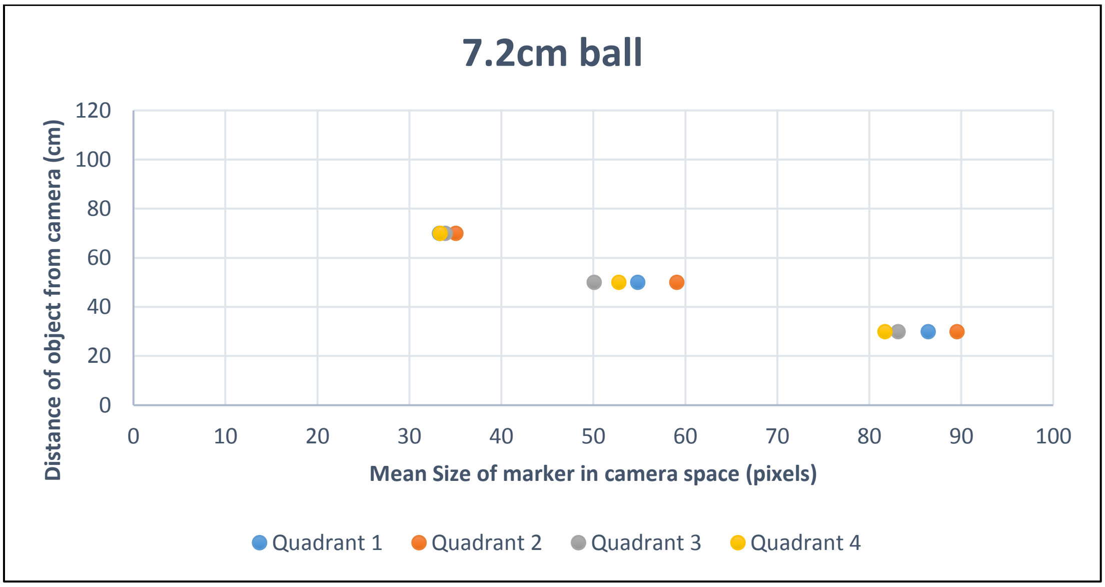


Figure 4.14: Graph of mean size of marker in camera space against distance of object from camera for 7.2 cm ball for condition 2

From the graph above, there were no detection for 7.2 cm ball at 100 cm distance from each quadrant. because the camera tend to detect the presence of shadow in camera space due to low level of lux which appear more larger compare to the size of ball. The performance also was bad compared to the detection of 7.2 cm ball at condition 1.

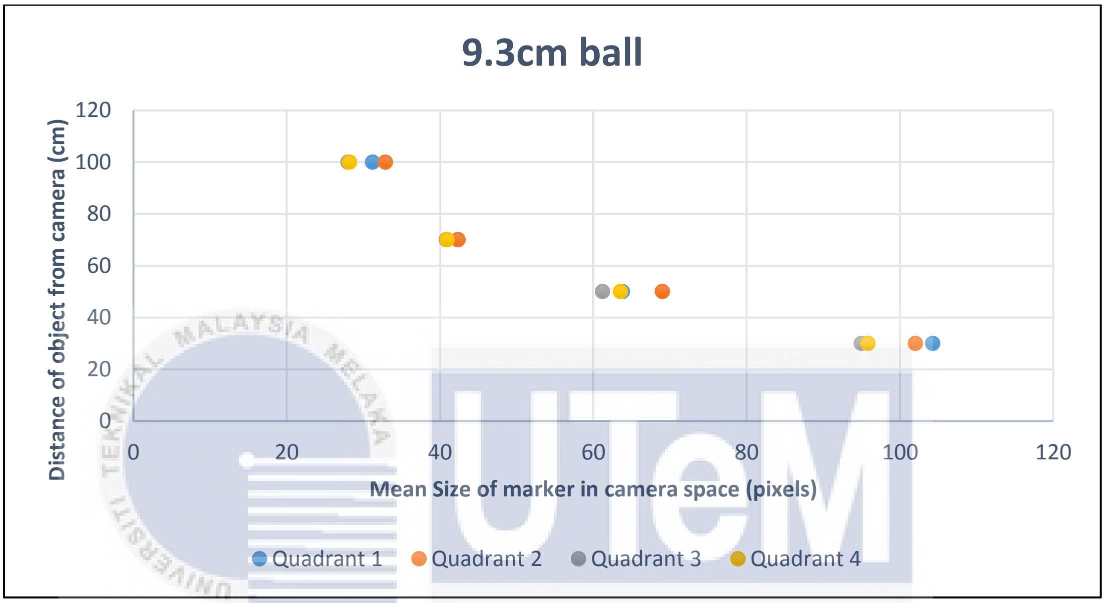


Figure 4.15: Graph of mean size of marker in camera space against distance of object from camera for 9.3 cm ball for condition 2

Based on the graph above, there was no problem in detecting the 9.3 cm ball since the size of the ball was more bigger than the shadow produce in the low level of lux condition supplied by the light source but the performance of the detection was not good as in condition 1.

For the conclusion, based from the level of lux produce by the light source, condition 1 was more preferred since the shadow produce was not much as in condition 2. The performance in object detection for condition 1 was better than in condition 2. Presence of shadow was disturbing the performance of the system in detecting the marker. The high level of light intensity was more preferrable to apply this system and it was suitable for the real condition in the surgical lamp.

Condition 1

Mean of error

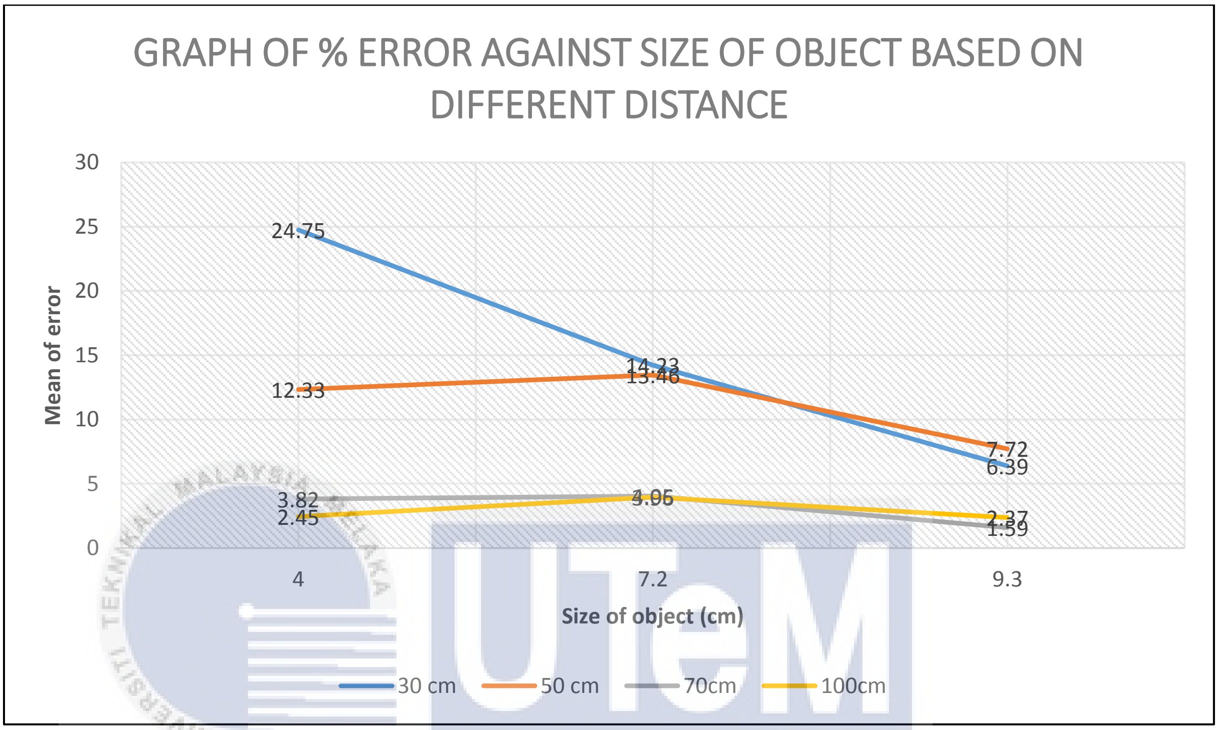


Figure 4.16: Graph of mean % error against size of object based on different distance at all quadrant for condition 1

From the graph above, the error produce in the distance of 100 cm for object detection for every size of object was the lowest compare to 30 cm, 50 cm and also 70 cm. This shown that at the furthest distance, the accuracy and the precision of the system in detecting the marker was higher due to the less error produce.

For the size of object, the error for the 9.3 cm ball at every distance was the lowest compare to the others. This prove that the larger the size was the better for the object detection. The larger the size, the easier it will be for the background subtraction process.

Condition 2

Mean of error

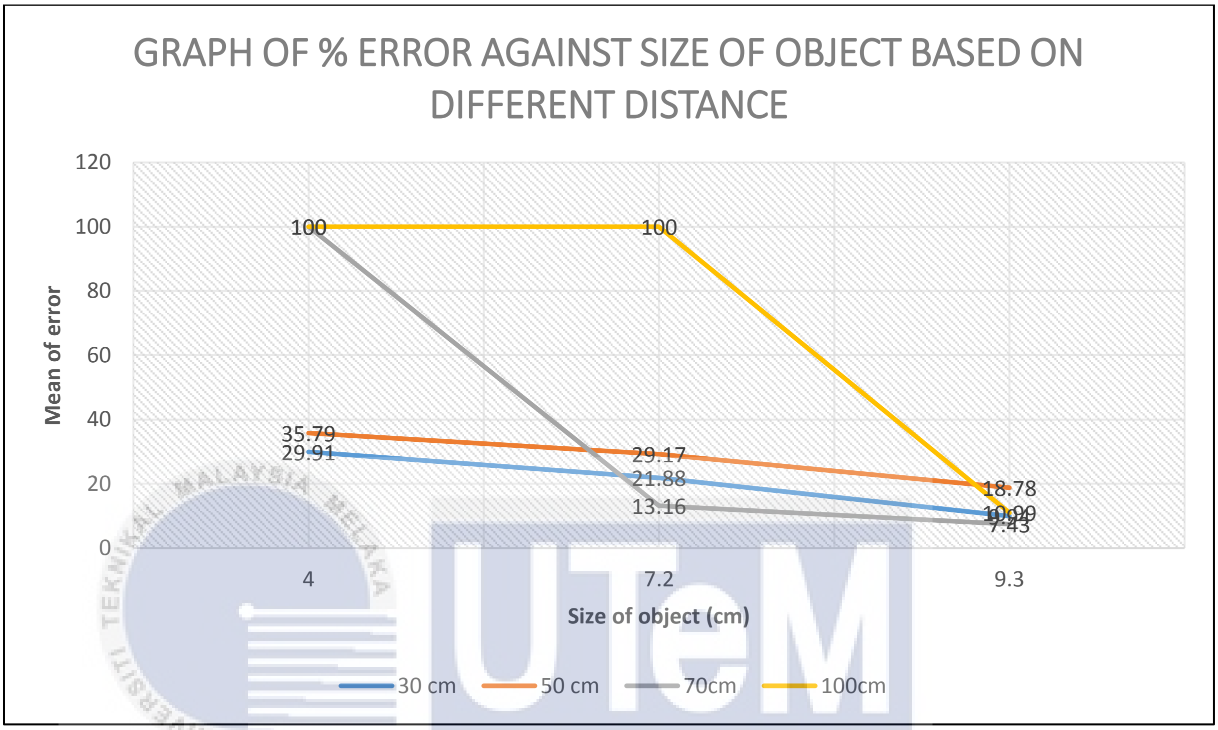


Figure 4.17: Graph of mean % error against size of object based on different distance at all quadrant for condition 2

Based on the graph, there was no detection when the 4.0 cm and 7.3 cm ball was placed at the furthest distance which is 100 cm and 70 cm. This shown that condition 2 was not suitable for object detection since it cannot reduce the presence of the shadow that influence the system in detecting the marker.

4.3 DISCUSSION ON THE DATA COLLECTED

As a result of experiments conducted, several conclusions were made to answer the stated objectives.

4.3.1 SOURCE OF LIGHT

Based on the result obtained from the experiment, additional of fluorescent light has give the advantage for the condition 1 to be more preferred in detecting markers. Condition 1 produced higher lux compare to the condition 2. All three black marker with different size were detected successfully from all distance at all quadrant that has been set for each experiment. To proof this results was valid, graph of percentage mean error against size of object at different distance for both condition has been made. In condition 2, the percentage of error for 4.0 cm size ball at 70cm and 100 cm was 100%. For 7.2 cm size ball the 100% of error occur at the distance of 100cm. This mean that the camera fail in tracking and detecting the small size of markers in condition 2. This problem occur because of the presence of shadow in the camera space that make the system track the shadow which look bigger instead of the 4.0 cm and 7.2 cm marker diameter. Condition 2 become less precise and not accurate to be used for tracking the marker. For the condition 1, all the marker can be detected successfully in the camera space because of lack of the shadow formed This lack of the shadow presence is because of the added light source that come from the 2 fluorescent lamps. From the graph above, condition 1 more precise than condition 2 because the mean value for each quadrant at different distance was concentrated and their value is close to each other. For the further study on this case, it was suggested that the experiment to be conduct in the high intensity of illumination with more than one light source to get a better tracking system.

4.3.2 DISTANCE OF OBJECT FROM CAMERA

From the results obtained in the experiment, the further the distance of the marker from the camera, the lesser the error produced. This prove that the object was less susceptible to error when viewed from a longer distance since the apparent circle was smaller and the discrepancy of the approximation of the circle outer edge was minimal. The cause of the problem for the nearest distance was because when the distance of object was near to the camera which mean the object was also near to the light source, the lux produce was high so the shadow produce behind the marker become darker. The camera will track the marker along with the shadow produce behind the object which make the object look much bigger in the camera space. To prevent this problem in future, we need to add additional lamp to this system so that the presence of shadow can be minimized. From the graph of the mean size of object collected from each quadrant, the point plot was very close to each other for the furthest distance compare to the nearest one which look more scattered and not concentrated. But this experiment produces bad result for the small size of object especially with the low level of lux. The furthest the distance, the smaller the size of marker that will be detected in camera space. So this test was limited only for the object that is bigger in size. For example, in the condition 2 which was affected by the decreasing amount of lux. The Pi camera failed in detecting the small object which was 4.0 cm ball and 7.2 cm ball in distance of 100cm because of the presence of shadow that has been track around the environment of the experiment which were larger than the size of both ball. This condition totally produces 100% error for both 4.0 cm and 7.2 cm ball when in 100cm of distance. Since the size of 4.0 cm ball is too small, it also produces 100% error when at distance of 70cm.

4.3.3 SIZE OF MARKER

The larger the size of marker, the higher the chance it will be detected in the camera space. Basically the algorithm build for this project is based on the size of marker and also about the shape since the method was limited to recognition of marker based on size, shape and color. The shape decided for this project was circular or round shape object with the minimum radius of 10 pixels. So the three ball with different size in cm has been used. The larger the size was the better for background subtraction process. It will make the process easier to differentiate the marker with the background. If the size of the object was large, it will not cause problems if the distance from the camera goes further. It is able to accommodate the addition of the distance for the object to be detected in the camera space. It also produce less percentage of error compared to the smaller size of object. From the experiment, the biggest size of marker which is 9.3 cm still can be detected in camera space even in condition 2, which has low level of lux compare to condition 1. It is because, the marker's size was larger than shadow produce around the environment of the experiment so the Pi camera track the marker which was the largest in the camera space. Different with the smaller size of object in this experiment which is 4.0 cm and 7.2 cm. The camera prefer to track the shadow produce around the experiment environment in condion 2 like in figure 4.7, figure 4.8 and figure 4.9 because the size of shadow was bigger than the size of object in the camera space. In condition 1, there was no problem in detecting the small size object since the light source produce high intensity of light. The shadow that produce was not much compare to the condition 2 so there was no problem regarding to the detection and tracking the smaller size of object. For the further study on this case, it was suggested that the experiment to be conduct in the high intensity of illumination with more than one light source to get a better tracking system.

4.3.4 POSITION OF OBJECT IN CAMERA SPACE

All the marker with the different size was placed in each quadrant of the camera space for the purpose of precision of the camera and also the accuracy of the camera. Based on the result obtained from the experiment, this camera show that all the quadrant working well as it can track all the object according to the executed program from the algorithm in tracking a round or circular black object with the radius more than 10 pixel. The algorithm was build to track an object which was the biggest in the camera space. Based on the results, the mean percentage error for all different size of object is below than 40% except for the condition 2 that produce 100% of error for 4.0 cm at distance 70 cm and 100 cm while on 7.2 cm object at distance 100 cm. This error occur was not that the camera space was fail to track, but due to the low level of lux that supplied in condition 2 that lead to the present of the shadow around the environment of the experiment. The camera track the the bigger size of black colour object that was viewed in the camera space like in figure 4.7, figure 4.8 and figure 4.9. To overcome this problem, it was suggested that the experiment to be conducted in the high intensity of illumination system with more than one light source to get a better tracking system.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

In conclusion, marker of size 9.3 cm was appropriate to be used for localization under 5000 lux illumination at all different distance for all quadrant. However, it was found that this system produced slightly error when determine the actual size of object in the camera space. This result may have been influenced by the simple algorithm used to detect the marker in this system.

The objective 1 was successfully achieve since the black circular object with the radius more than 10 pixels can be detected in the camera space and the movement can be track by investigating two previous (x, y)-coordinates of object in the frame. The marker with size 9.3 cm was well detected at all distance within all 4 quadrant. The error produce by this size of marker was less compared to the 7.2 cm and 4.0 cm size of marker. The error produce when the distance was 100 cm was the least compared to the 30 cm, 50 cm and 70 cm which mean that this distance was suitable for object localization to be applied to the real situation.

The objective 2 which to locate marker's position in camera was achieved by conducting the experiment by observing the three different size of black marker in all part of camera space that is divided equally into 4 quadrant which is quadrant 1, quadrant 2, quadrant 3, and quadrant 4. This experiment was conducted in two condition with the different distance from the camera. For condition 1, there were another source of light added to the headlamp which was fluorescent lamp and in condition 2, there was no other source of light except the light from the headlamp. All the three black markers was detected successfully in all quadrant except for the condition 2. At the furthest distance, the error become larger and even become undetected. This was due to the presence of the shadow cause by low level of lux from the light source that make the camera fail to detect the smaller object that cause by the increasing of distance and tend to detect the shadow which look bigger. A better algorithm to remove the shadows need to be implemented in the future study to improve detection.

The objective 3 which was identify the precision and accuracy of the system was fulfilled by collected data from the experiment and the plotted graph of mean size of marker in camera space (pixels) against distance of object from camera (cm) and also graph of percentage error against size of object based on different distance. This graph was plotted for the observation of the trend of the data recorded. From the graph, the marker with larger size which was 9.3 cm was more precise in object detection since the value taken in each quadrant for every distance was not differ much compared 4.0 cm and 7.2 cm marker. The suitable distance also was identified from the plotted graph that show when the marker was in the furthest distance, the error produce was less except for condition 2. A few condition that make the system more precise and accurate has been identified. For the source of illumination, the higher level of lux was better for object localization and also for object tracking. The further the distance of object from the camera, the less error produce by the system but it also depends to the size of object. The larger the object, the further the distance it can be detected in the camera space. It was able to accommodate the addition of the distance for the object to be detected in the camera space The larger the object, the higher the chance it will be detected in the camera space. Basically the algorithm build for this project was based on the size of marker and also about the shape since the method was limited to recognition of marker based on size, shape and color.

5.2 FUTURE DEVELOPMENT

The current implementation has some weaknesses that can be overcome by the next researcher in order to improvise this tracking system. Firstly, the system cannot do the detection well in low lux level of illumination system. Further study on stabilizing the illumination and more robust shadow removal algorithm can be studied to improve the detection. Next recommendation is using the better camera with high specification for example the latest version of Pi camera to produce a better performance of camera in tracking the marker. Lastly, it is suggested to use the more powerful processor such as Beagle Bone Back which has better specification compare to raspberry pi so that the processor can operate with high speed without causing any delay that will interrupt the tracking system.

REFERNCES

[1] M. F. Zakaria, H. S. Choon, and S. A. Suandi, “Object Shape Recognition in Image for Machine Vision Application,” *Int. J. Comput. Theory Eng.*, vol. 4, no. 1, pp. 76–80, 2012.

[2] R. Valenti, N. Sebe, and T. Gevers, “Combining head pose and eye location information for gaze estimation,” *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 802–815, 2012.

[3] H. Yoshimoto and Y. Nakamura, “Free-angle 3D head pose tracking based on online shape acquisition,” *Proc. - 2nd IAPR Asian Conf. Pattern Recognition, ACPR 2013*, pp. 798–802, 2013.

[4] R. Dockter and T. M. Kowalewski, “A Low-Cost Computer Vision-Based Approach for Tracking Surgical Robotic Tools,” vol. 7, no. September 2013, pp. 2013–2014, 2014.

[5] N. Binti Nordin and M. N. Bin Muhammad, “An investigation of color based marker recognition for 3D instrument localization in surgical applications,” *2011 IEEE Symp. Ind. Electron. Appl. ISIEA 2011*, pp. 405–409, 2011.

[6] Y. Çelik, “Color-Based Moving Object Tracking with An Active Camera Using Motion Information,” 2017.

[7] T. T. Trang and C. Ha, “Irregular Moving Object Detecting and Tracking Based on Color and Shape in Real-time System,” pp. 415–419, 2013.

[8] A. Troppan, E. Guerreiro, F. Celiberti, G. Santos, A. Ahmad, and P. U. Lima, “Unknown-color spherical object detection and tracking,” *Proc. 2013 13th Int. Conf. Auton. Robot Syst. Robot. 2013*, pp. 1–4, 2013.

[9] Gonzalez, Rafael C. & Woods, Richard E. (2002). Thresholding. In Digital Image Processing, pp. 595–611. Pearson Education

[10] Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 153. Prentice Hall, 2001

[11] Spencer, Will (2013), HSV (Hue, Saturation, and Value), Retrieved from http://www.tech-faq.com/hsv.html

[12] Alexander Mordvintsev & Abid K. Revision 43532856. (2013), Retrieved from http://opencvpythontutroals.readthedocs.io/en/latest/py_tutorials/ py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#contours-getting-started

[13] B. Tamersoy (September 29, 2009). "Background Subtraction – Lecture Notes"(PDF). University of Texas at Austin.

[14] Field of view and focal lengthWritten by Paul Bourke April 2003

APENDICES

Appendix A: Algorithm written in OpenCV

```
# import the necessary packages
from collections import deque
from imutils.video import VideoStream
import numpy as np
import argparse
import cv2
import imutils
import time
# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video",
	help="path to the (optional) video file")
ap.add_argument("-b", "--buffer", type=int, default=32,
	help="max buffer size")
args = vars(ap.parse_args())
# define the lower and upper boundaries of the "black"
# ball in the HSV color space
```

```
blackLower = (0, 0, 0)

blackUpper = (180, 255, 40)


# initialize the list of tracked points, the frame counter,

# and the coordinate deltas

pts = deque(maxlen=args["buffer"])

counter = 0

(x, y) = (0, 0)

direction = ""


# if a video path was not supplied, grab the reference

# to the pi camera

if not args.get("video", False):

	camera = cv2.VideoCapture(0)


# otherwise, grab a reference to the video file

else:

	camera = cv2.VideoCapture(args["video"])

# allow the camera or video to warm up

time.sleep(2.0)

# keep looping

while True:

	# grab the current frame
```

```
(grabbed, frame) = camera.read()


# if we are viewing a video and we did not grab a frame,
# then we have reached the end of the video
if args.get("video") and not grabbed:
        break


# resize the frame, blur it, and convert it to the HSV
# color space
frame = imutils.resize(frame, width=600)
blurred = cv2.GaussianBlur(frame, (11, 11), 0)
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)


# construct a mask for the color "black", then perform
# a series of dilations and erosions to remove any small
# blobs left in the mask
mask = cv2.inRange(hsv, blackLower, blackUpper)
mask = cv2.erode(mask, None, iterations=2)
mask = cv2.dilate(mask, None, iterations=2)


# find contours in the mask and initialize the current
# (x, y) center of the ball
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)[-2]
    center = None
    # only proceed if at least one contour was found
    if len(cnts) > 0:
        # find the largest contour in the mask, then use
        # it to compute the minimum enclosing circle and
        # centroid
        c = max(cnts, key=cv2.contourArea)
        ((x, y), radius) = cv2.minEnclosingCircle(c)
        M = cv2.moments(c)
        center = (int(M["m10"] / M["m00"]), int(M["m01"] /
M["m00"]))

        # only proceed if the radius meets a minimum size
        if radius > 10:
            # draw the circle and centroid on the frame,
            # then update the list of tracked points
            cv2.circle(frame, (int(x), int(y)), int(radius),
                (0, 255, 255), 2)
            cv2.circle(frame, center, 5, (0, 0, 255), -1)
            pts.appendleft(center)

    # loop over the set of tracked points
```

```
for i in np.arange(1, len(pts)):

    # if either of the tracked points are None, ignore them

    if pts[i - 1] is None or pts[i] is None:

        continue


    # check to see if enough points have been accumulated in

    # the buffer

    if counter >= 10 and i == 1 and len(pts)>=10 is not None:

        # compute the difference between the x and y

        # coordinates and re-initialize the direction

        # text variables

        dX = pts[-10][0] - pts[i][0]

        dY = pts[-10][1] - pts[i][1]

        (dirX, dirY) = ("", "")


        # ensure there is significant movement in the

        # x-direction

        if np.abs(dX) > 20:

            dirX = "East" if np.sign(dX) == 1 else "West"


        # ensure there is significant movement in the

        # y-direction

        if np.abs(dY) > 20:
```

```
                    dirY = "North" if np.sign(dY) == 1 else "South"


                # handle when both directions are non-empty
                if dirX != "" and dirY != "":
                    direction = "{}-{}".format(dirY, dirX)


                # otherwise, only one direction is non-empty
                else:
                    direction = dirX if dirX != "" else dirY
            # otherwise, compute the thickness of the line and
            # draw the connecting lines
            thickness = int(np.sqrt(args["buffer"] / float(i + 1)) * 2.5)
            cv2.line(frame, pts[i - 1], pts[i], (0, 0, 255), thickness)
            # show the movement deltas and the direction of movement on
            #the frame
        cv2.putText(frame, direction, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX,
            0.65, (0, 0, 255), 3)
        cv2.putText(frame, "object centroid x: {}, y: {}".format(x, y),
            (10, frame.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.4, (0, 0, 0), 1)
       cv2.putText(frame, "object radius : {}".format(radius),
            (400, frame.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX,
        0.4, (0, 0, 0), 1)


    # show the frame to our screen and increment the frame counter
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    counter += 1


    # if the 'q' key is pressed, stop the loop
    if key == ord("q"):
        break


# cleanup the camera and close any open windows
camera.release()
cv2.destroyAllWindows()
```