VISUAL MOUSE (COLOR TRACKING)

By

MUHAMMAD MUSTAFA BIN BAHRUDIN

FINAL YEAR PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

VISUAL MOUSE

by

Muhammad Mustafa bin Bahrudin

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Mr. Patrick Sebastian
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Muhammad Mustafa bin Bahrudin

ABSTRACT

This Final Year Project is about Visual Mouse (Color tracking). This project will demonstrates how a webcam could be used to track objects in front of it and track hand movements to draw on the computer. It is applicable for paper presentations and lectures. This project will deal with algorithm and tools in MATLAB software as its medium. The basic principle of this project would be distinguishing color and use its properties to move a cursor in MATLAB and draw as well.

ACKNOWLEDGEMENTS

I would like to express my thanks to Mr. Patrick Sebastian, my supervisor, for his support, supervision and thorough teachings. His problem solving skills and supervision are essential for the completion of this project. I have learnt a lot from him and also from his experiences that he shared with me.

Finally I would like to show appreciation to my mother Mariam binti Sarman and little brother Kamil Rashidi bin Bahrudin and my friends for giving me a constant source of strength during my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CRT	Cathode Ray Tube
GUI	Graphical User Interface
LCD	Liquid Crystal Display
RGB	Red Green Blue

CHAPTER 1

INTRODUCTION

1.1. Introduction

This project involves the development of a system that captures movement by matching certain objects and colors through a webcam and analyzes using MATLAB software. The system will capture live images and track object movements and will output by moving a cursor on the computer.

This project consists of software development for tracking purposes and producing a drawing which includes erasing, dragging and zooming function.

1.2. Problem statement

Develop a system that track an object with a specific color range and produce a drawing from the movement of the object.

The application of the project will help to enhance the value of presentation through web camera. Often lecturers face difficulties to control lecture slides while on the center stage and trying to write formulas or draw diagrams using the computer.

To solve these shortcomings, the implementation of object tracking using a web camera is beneficial. The application utilizes image processing techniques to have a web-camera placed at a specific location and tracks specific objects or hand movement to control a cursor.

1.3. Objective and Scope of Study

The primary objective of this project is to enhance the utility of the existence web camera for object tracking as well. Hence, the project targets are as follows:

- to be able to generate a MATLAB program that performs object tracking

- to produce a working model of the of the project

The input of the system generated will be the live visual image from any web camera that is connected to a computer which will run the MATLAB program. This program will consist of a few stages to identify, process, compare, filter and track the image or object.

CHAPTER 2

LITERATURE REVIEW AND THEORY

2.1 Segmentation

In computer vision, **segmentation** refers to the process of partitioning a digital image into multiple regions (sets of pixels). The goal of segmentation is to simplify and change the representation of an image into something that is more meaningful and easier to analyze. [1] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).[1]

Some of the practical applications of image segmentation are:

- Medical Imaging
- Locate objects in satellite images (roads, forests, etc.)
- Face recognition systems
- Automatic traffic controlling systems
- Machine vision

Several general-purpose algorithms and techniques have been developed for image segmentation. Since there is no general solution to the image segmentation problem, these techniques often have to be combined with other techniques in order to effectively solve an image segmentation problem.

2.2 RGB color model

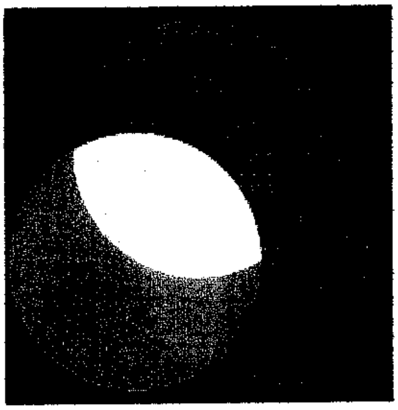


Figure 1: A representation of additive color mixing

In CRT based (analog electronics) television three color electron guns are used to stimulate such an arrangement of phosphorescent coatings of the glass, the resultant reemission of photons providing the image seen by the eye.

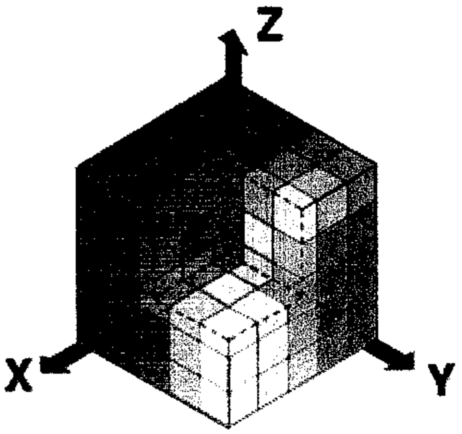


Figure 2: The RGB color model mapped to a cube (with cut-away shown)

The **RGB color model** or **RBG color standard** (often spelled **RBG** in historical engineering literature) is an additive model in which red, green, and blue (often used in additive light models) are combined in various ways to reproduce other colors. The name of the model and the abbreviation 'RGB' come from the three fundamental colors, red, green, and blue and the discovery of cathode ray tubes which could display color instead of a monochrome phosphoresence (including grey scaling) such as black and white film and television imaging.

RGBA is also used, to mean Red, Green, Blue, and Alpha. It is not a different color model, but a different representation where Alpha is used for transparency.

However, do not confuse the colors with the primary pigments which are red, blue and yellow also known as primary colors in the art world. The latter depends on reflection and absorption of photon while the former, RGB, depends on photon emission produced from compound excitation by impact with an electron beam.

The RGB color model itself does not define what is meant by 'red', 'green' and 'blue' and so the results of mixing them are not specified as exact.

When the exact spectral make-up of the red, green and blue primaries are defined, the color model would then become what is known in science and engineering as an absolute color space, such as sRGB or Adobe RGB.

2.2.1 Biological relevance of primary colors

The choice of 'primary' colors was derived from human anatomy of the eye because they are stimuli that maximised the difference between the physiological responses of the cone cells of the human retina to light falling within the small part of the electromagnetic continuum which is capable of stimulating these cells, and hence is what we call visible. Spectroscopically, Red and Blue are located at opposite ends of the visible light spectrum — usually measured as an electomagnetic wave, in or by frequencies (hertz or cycles per second), or wavelength's (usually measured in a sub-multiple of the metre, as in the nanometer — abbreviated as **nm** as is cited below.)

The normal three kinds of light-sensitive photoreceptor cells in the human eye (cone cells) respond most to yellowish-green (long wavelength or L), bluish-green (medium or M) and bluish-violet (short or S) light (peak wavelengths of 564 nm, 534 nm and 420 nm respectively). The difference in the signals received from the three kinds allows the brain to differentiate a wide gamut of different colors, while being most sensitive (overall) to green light and to differences between shades of green.

As an example, suppose that light in the yellow range of wavelengths (approximately 577 nm to 597 nm) enters the eye and strikes the retina. Light of these wavelengths would activate both the medium and long wavelength cones of the retina, but not equally – the long-wavelength cells will respond more (fire more frequently).

The difference in the response can be detected by the brain and associated with the concept that the light is 'yellow'. In this sense, the yellow appearance of objects is simply the result of yellow light from the object entering our eye and stimulating the relevant kinds of cones simultaneously but to different degrees.

To generate optimal color ranges for species other than humans, other primary colors would have to be used. For species with four different color receptors, such as many birds, one would use four primary colors; for species with just two kinds of receptors, such as most mammals, one would use two primaries.

Even for humans, use of the three 'primary' colors is not the most efficient. In theory, three kinds of emitters that matched the response curves of the cones should produce results closer to 'real life' with less wasted energy.

2.2.2 RGB and displays

One common application of the RGB color model is the display of colors on a cathode ray tube, liquid crystal display or plasma display, such as a television or a computer's monitor. Each pixel on the screen can be denoted in the computer or interface hardware (for example, a 'graphics card') as values for red, green and blue. These values are converted into intensities which are then used for display.

By using an appropriate combination of red, green and blue intensities, many colors can be represented. Typical display adapters in 2007 used up to 24 bits of information for each pixel. This is usually apportioned with 8 bits each for red, green and blue, giving a range of 256 possible values, or intensities, for each hue. With this system, 16,777,216 (256^3 or 2^{24}) discrete combinations of hue and intensity can be specified. It is claimed that the human eye can distinguish as many as 10 million discrete hues (this number varies from person to person depending upon the condition of the eye and the age of the person). However, at the resolution of current screens and at a standard viewing distance people cannot distinguish more than a few hundred hues.

2.2.2.1 Video electronics

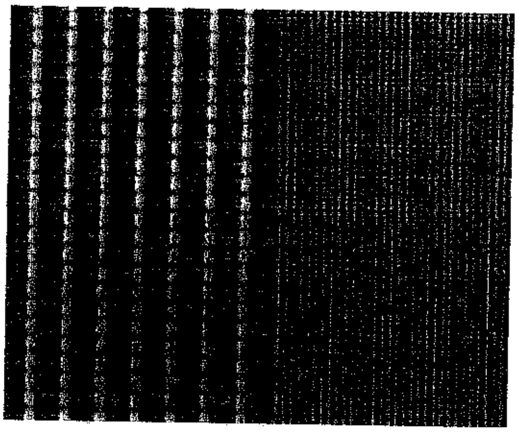


Figure 3: RGB pixels in an LCD TV (on the right - an orange and a blue color, on the left - a close up of pixels)

RGB is a type of component video signal used in the video electronics industry is carried on three separate cables/pins. Extra cables are sometimes needed to carry synchronizing signals. RGB signal formats are often based on modified versions of the RS-170 and RS-343 standards for monochrome video. This type of video signal is widely used in Europe since it is the best quality signal that can be carried on the standard SCART connector. Outside Europe, RGB is not very popular as a video signal format – S-Video takes that spot in most non-European regions. However, almost all computer monitors around the world use RGB.

2.2.2.2 Non-linearity

The intensity of the color output on computer display devices is normally not directly proportional to the R, G and B values. That is, even though a value of 0.5 is very close to halfway between 0 and 1.0 (full intensity), the light intensity of a computer display device when displaying (0.5, 0.5, 0.5) is normally (on a standard 2.5 gamma CRT / LCD) only 18% of that when displaying (1.0, 1.0, 1.0), instead of at 50%.

2.2.2.3 Professional color calibration

Proper reproduction of colors in professional environments requires extensive color calibration of all the devices involved in the production process. This results in several transparent conversions between device-dependent color spaces during a typical production cycle in order to ensure color consistency throughout the process. Along with the creative processing, all such interventions on digital images inherently damage it by reducing its gamut. Therefore the denser the gamut of the original digitized image, the more processing it can support without visible degradation. Professional devices and software tools allow for 48 bpp (bits per pixel) images to be manipulated (16 bits per channel) in order to increase the density of the gamut.

2.2.3 Representations

2.2.3.1 Numeric representations

A color in the RGB color model can be described by indicating how much of each of the red, green and blue color is included. Each can vary between the minimum (no color) and maximum (full intensity). If all the colors are at minimum the result is black. If all the colors at maximum, the result is white. A confusing aspect of the RGB color model is that these colors may be written in several different ways.

- Color science talks about colors in the range 0.0 (minimum) to 1.0 (maximum). Most color formulae take these values. For instance, full intensity red is 1.0, 0.0, 0.0.

- The color values may be written as percentages, from 0% (minimum) to 100% (maximum). To convert from the range 0.0 to 1.0, see percentage. Full intensity red is 100%, 0%, 0%.

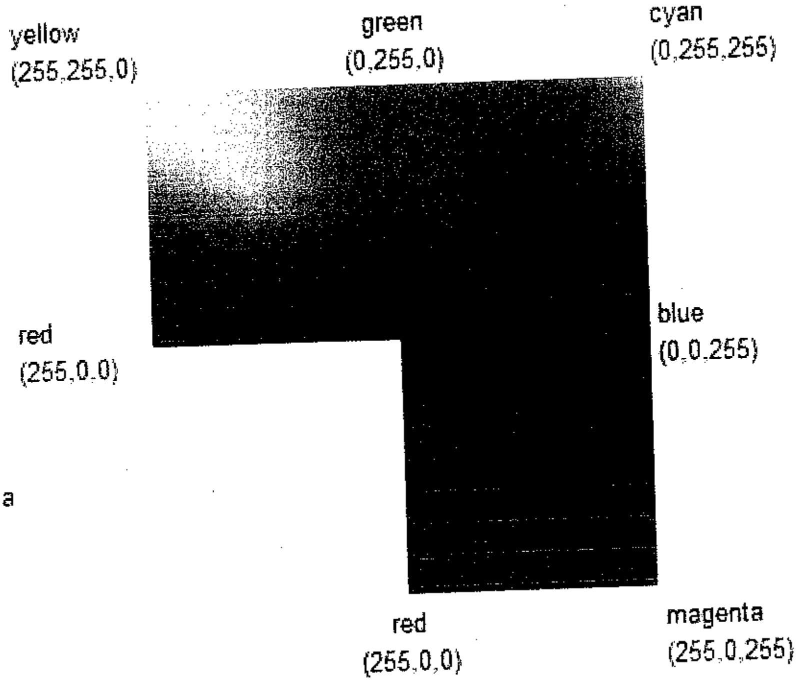
- The color values may be written as numbers in the range 0 to 255, simply by multiplying the range 0.0 to 1.0 by 255. This is commonly found in computer science, where programmers have found it convenient to store each color value in one 8-bit byte. This convention has become so widespread that many writers now consider the range 0 to 255 authoritative and do not give a context for their values. Full intensity red is 255,0,0.

- The same range, 0 to 255, is sometimes written in hexadecimal, sometimes with a prefix (e.g. #). Because hexadecimal numbers in this range can be written with a fixed two digit format, the full intensity red #ff, #00, #00 might be contracted to #ff0000.

2.2.3.2 24-bit representation

When written, RGB values in 24 bits per pixel (bpp), also known as Truecolor, are commonly specified using three integers between 0 and 255, each representing red, green and blue intensities, in that order. For example:

- (0, 0, 0) is black
- (255, 255, 255) is white
- (255, 0, 0) is red
- (0, 255, 0) is green
- (0, 0, 255) is blue
- (255, 255, 0) is yellow
- (0, 255, 255) is cyan
- (255, 0, 255) is magenta



The above definition uses a convention known as *full-range* RGB. This convention is so often used that some people have come to view it as universal. This can be confusing because color values are also often considered to be in the range 0.0 through 1.0, rather than 0 to 255 (the latter range is used when colors are encoded in eight bits, which encoding permits 256 different values (sometimes written using two hexadecimal characters)). If in doubt, it is best to describe the range over which a color is specified.

Full-range RGB can represent up to two hundred and fifty-five shades of a given hue. (Only pure reds, greens, blues or greys have this full range of shades.)

Typically, RGB for digital video is not full range. Instead, video RGB uses a convention with scaling and offsets such that (16, 16, 16) is black, (235, 235, 235) is white, etc. For example, these scalings and offsets are used for the digital RGB definition in CCIR 601.

CHAPTER 3

METHODOLOGY

The project starts with research and studies conducted based on journals, white papers books and articles. The familiarization of the MATLAB software is done systematically in order to have ease in generating the program for the system.

The project algorithm is initiated once the theories from the literature review conducted earlier are understood. The development of the MATLAB programs requires study on ideas and generating the program involves supervision. This stage consumes time as it is an ongoing process of trial and troubleshooting in order to choose and obtain the best method for the project. Once the desired results are achieved, a comparison of the tracking mode will be implemented. It is then finalized and further enhancement is implemented.

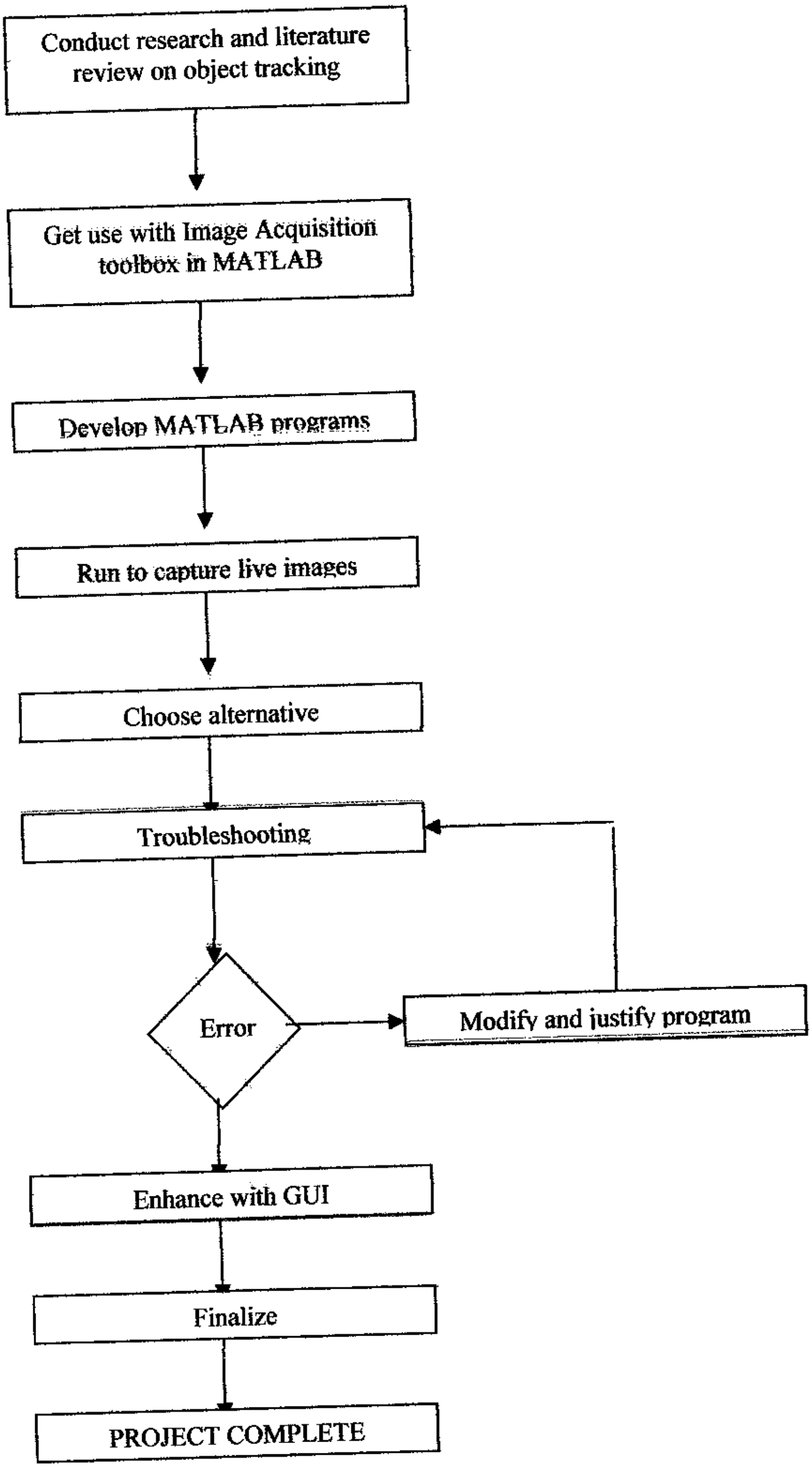


Figure 4: Project Flow

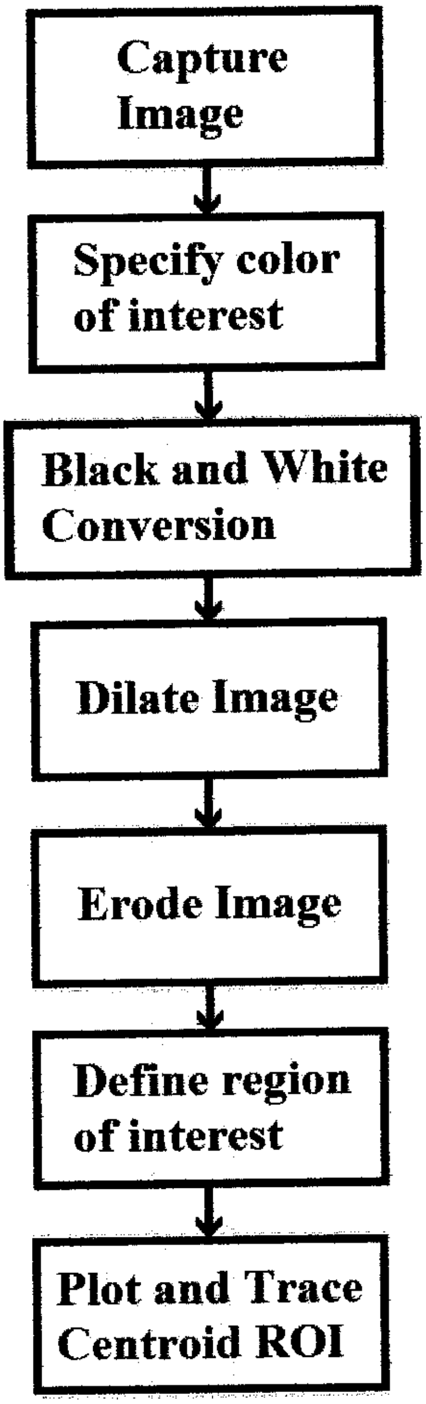


Figure 5: Program Flow

CHAPTER 5

RESULTS

In this chapter, the results and findings on each tracking method applied in this project will be discussed.

5.1 Masking

The reference image used for the project is as in Figure 6 below which is a blue-green glove.

Figure 6: Blue-Green glove

The next step is to find the base color of the glove and use it to omit the background and put a tracking marker on the glove. The nearest color was of four colors: Azure, Cerulean Blue, Cobalt Blue and Denim [3] as referred to.

Using the 'pixval' function in MATLAB, the value of each color components was found.

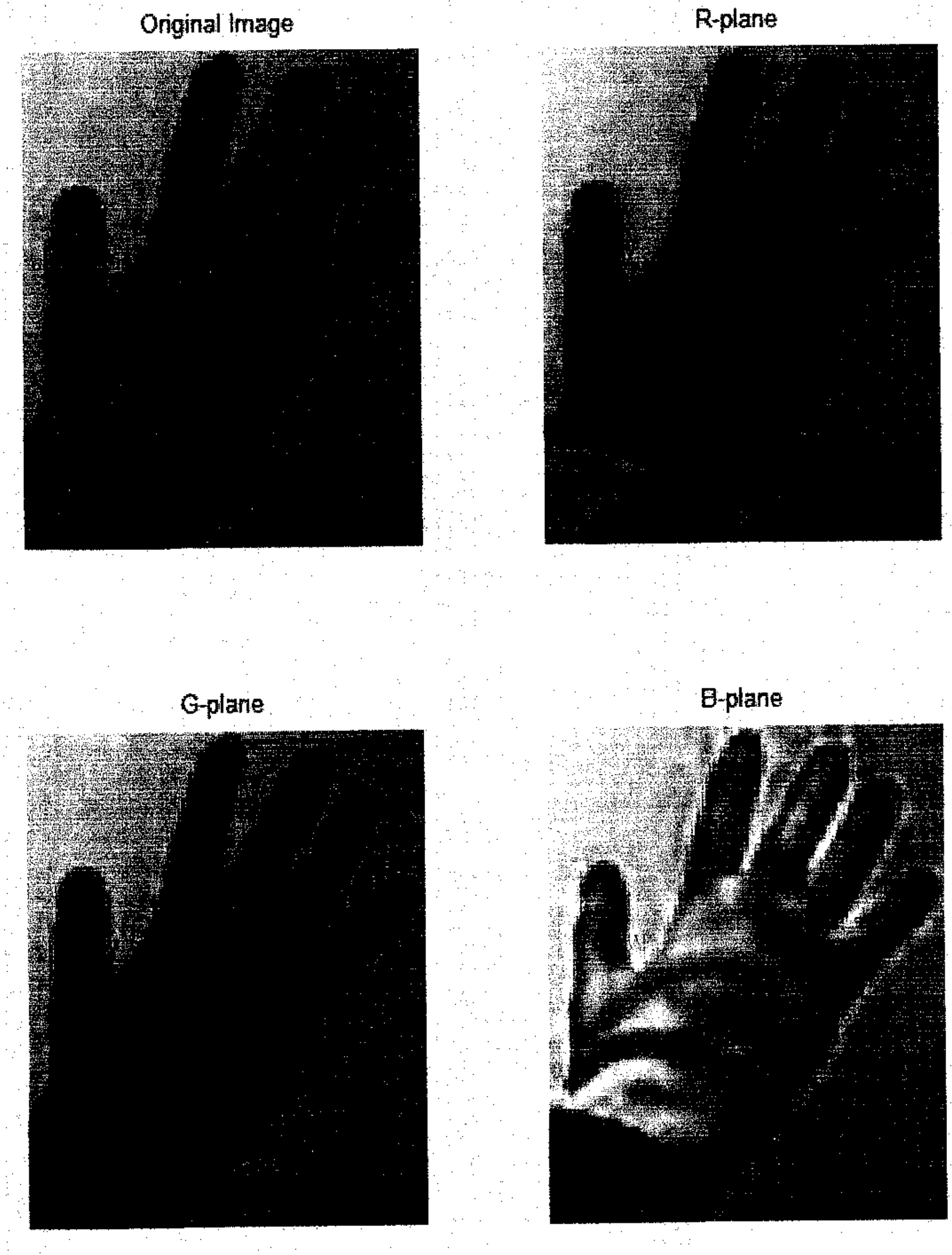


Figure 7: Reference image break into RGB components

From the gray images, the highest value and the lowest value of the glove were taken to make the mask.

Minimum	Component	Maximum
0	Red	5
17	Green	144
68	Blue	248

Table 1: Glove minimum and maximum gray levels according to RGB components

5.2 Result of masking

Figure 8: The produced mask

The produced mask will then be pre-processed to produce a marker. Referring to the program flow, the following are the steps taken.

1. Capture image:

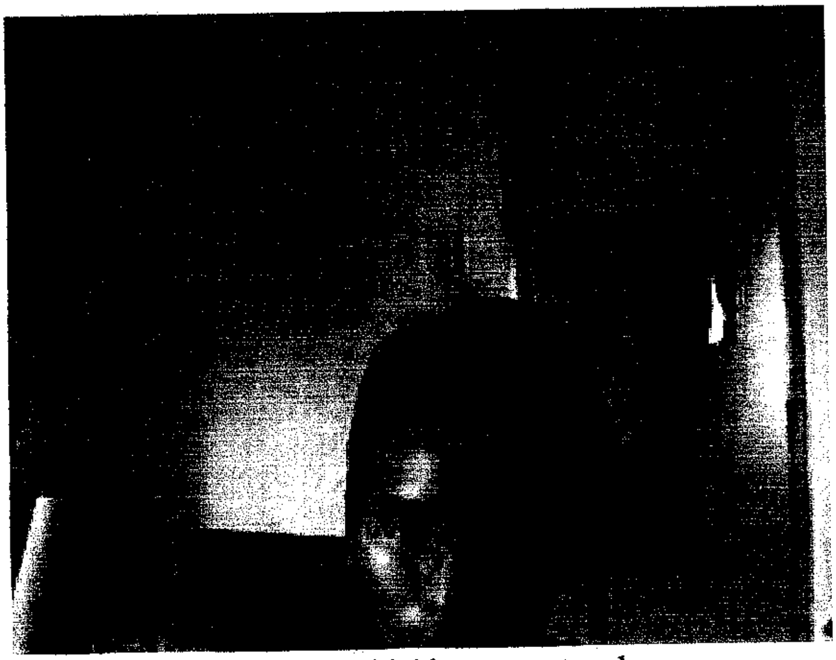


Figure 9: Initial image captured

Initial image is captured before running image processing using MATLAB and stored under a label.

2. Specify color of interest:

Figure 10: Blue color plane

As the color of interest is the color of the glove, the blue color plane is selected where the brightest is the maximum intensity of blue in the corresponding color plane.

3. Black and White conversion:

Figure 11: Black and White converted image

The image is then converted to black and white which is equivalent with binary image for further processing.

4. Eroding or Dilating:

Figure 12: First Erosion

Erosion process is applied to the black and white image to distinguish the glove.

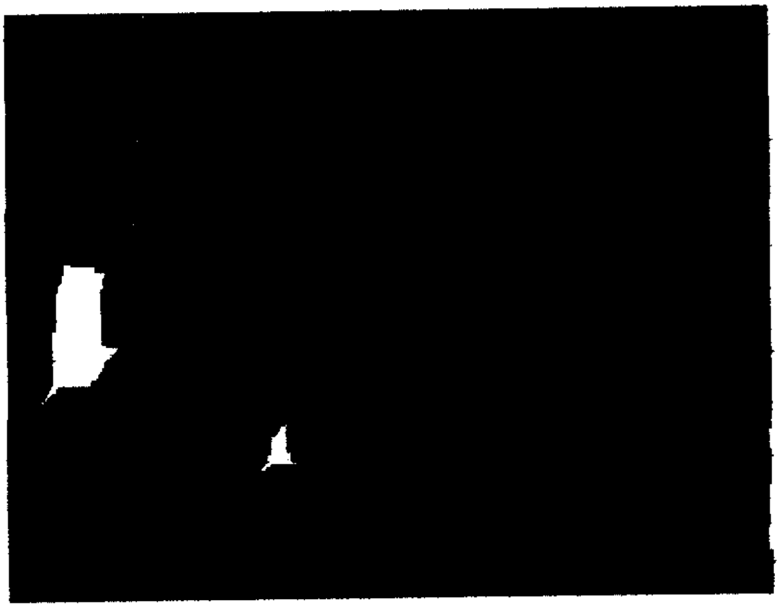


Figure 13: Second Erosion

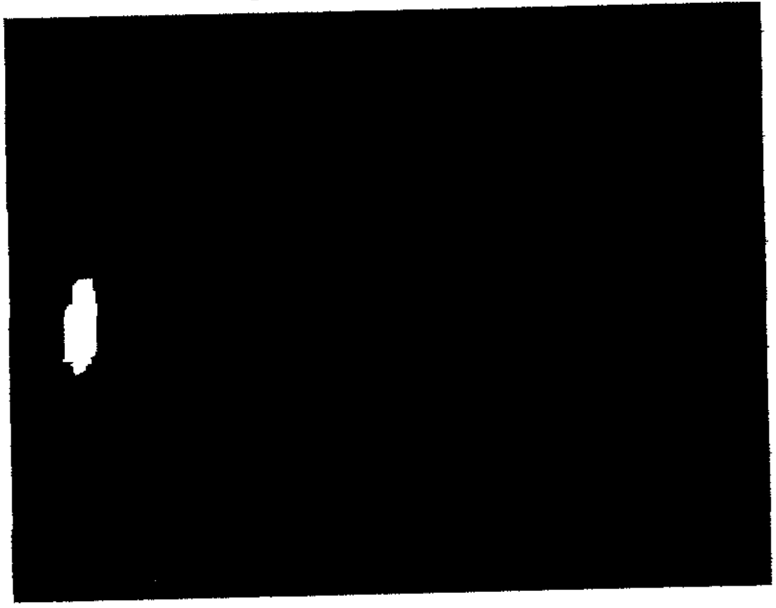


Figure 14: Third Erosion

5. Define the region of interest:

The region of interest is defined as the largest white region on the eroded image. Using 'regionprops' function, the position of the extracted object (color of the glove) is mark with a red star. This uses the centroid of the object left (assuming all unneeded features are omitted).

Figure 13: Marked object representing the glove position

6. Plot and trace:

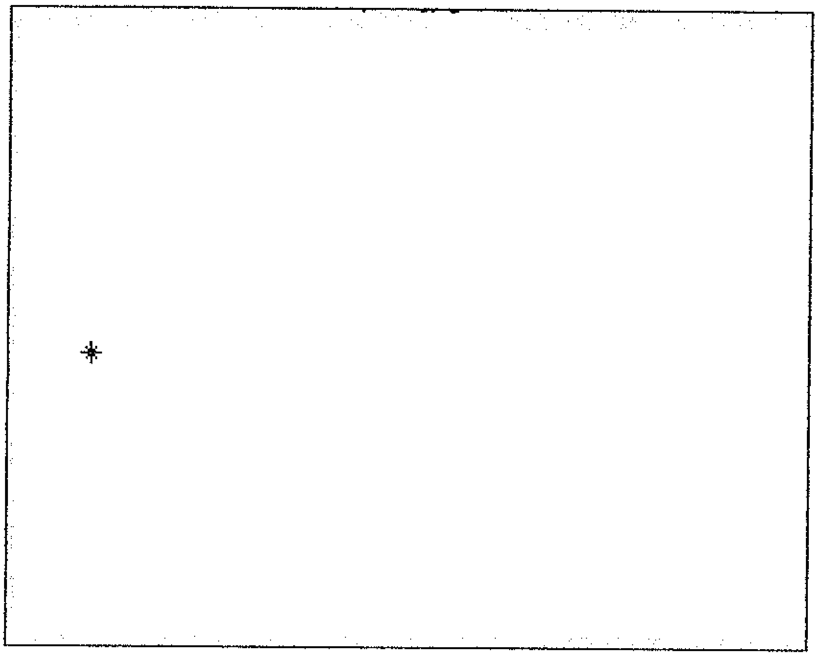


Figure 15: Marker on a white background

The marker is taken out from the image and plotted on a white background for drawing purpose which uses a trace function in MATLAB.

CHAPTER 4

CONCLUSION AND DISCUSSION

The project is interesting in the sense it teaches how to segment objects of a certain color from the background and use it to produce a drawing application. It provides massive of experience and knowledge in image processing.

There are some other alternative to approach tracking an object such as using correlation where a reference image is kept and then compared with captured image. the outcome will be a peak which represents the location of the object in the image.

In this project, color is used because of its simplicity and practicality. Different colors can be used for tracking. However, ample amount of light is needed for capturing images to distinguish the color of interest.

REFERENCES

[1] Linda G. Shapiro and George C. Stockman (2001): "Computer Vision", pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[2] Fundamental requirements for picture presentation, Cowlishaw, M. F., *Proc. Society for Information Display*, Volume 26, No. 2 (1985)

[3] Dhuha bt Shamsul Iwardi 2006, Camera tracking, Universiti Teknologi PETRONAS.

[4] The Mathworks, Image Processing Toolbox.

[3] Frery, A. C.; Melo, C. A. S. & Fernandes, R. C. Web-Based Interactive Dynamics for Color Models Learning. *Color Research and Application*, 2000, 25, 435-441. DOI 10.1002/1520-6378(200012)25:6<435::AID-COL8>3.0.CO;2-J

[4] Sanjit K. Mitra, "Digital Signal Processing - A Computer Based Approach", Mc Graw Hill, Third Edition, 2006

[5] J.P Lewis, "Fast Normalized Cross-Correlation, In Vision Interface", 1995.

APPENDICES

Masking code:

```
figure;
vcapg2(0);


while(1)
    capture = vcapg2(0);
    pause(0.05);
    R = capture(:,:,1);
    G = capture(:,:,2);
    B = capture(:,:,3);

    mask = ( (R<5) & (17<G<144) & (68<B<248));

    subplot(1,2,1), imshow(capture), title('Original
Image');
    subplot(1,2,2), imshow(mask), title('Mask');

end
```

```
figure;
vcapg2(0);

while(1)
    capture = vcapg2(0);
    pause(0.05);
    G = capture(:,:,2);

    mask = G>150;
    %[y,x] = find(mask==1);
    %x=round(mean(x));
    %y=round(mean(y));

    subplot(1,2,1),imshow(capture),title('Original Image');
    subplot(1,2,2),imshow(mask), title('Mask');

 end
```

Tracking glove code

```
I = imread('blue2.jpg');
I2 = I(:,:,3);
I3 = im2bw(I2,0.7);
se1 = strel('line',10,45)       % line, length 10, angle 45
degrees
se2 = strel('square',11)       % 11-by-11 square
I4 = imerode(I3,se1);
I5 = imerode(I4,se2);
[L,num] = bwlabel(I5);
stats_cent = regionprops(L,'centroid');
cent = [stats_cent(1).Centroid];
imshow(I)
hold on; plot(cent(1),cent(2),'r*')
```

Latest coding (works to tracking extent):

```
figure;
vcapg2(0);


while(1)
    capture = vcapg2(0);
    pause(0.05);
    G = capture(:,:,3);


    I3 = im2bw(G,0.7);
    se1 = strel('line',10,45)      % line, length 10, angle
 45 degrees
    se2 = strel('square',11)      % 11-by-11 square
    I4 = imerode(I3,se1);
    I5 = imerode(I4,se2);
    [L,num] = bwlabel(I5);
    stats_cent = regionprops(L,'centroid');
    cent = [stats_cent(1).Centroid];


    %[y,x] = find(mask==1);
    %x=round(mean(x));
    %y=round(mean(y));


    subplot(1,2,1),imshow(capture),title('Original Image');
    subplot(1,2,2),imshow('plot.jpg'),
    hold on; plot(cent(1),cent(2),'r*')
    hold off;
    title('Marker');


  end
```