CAMERA TRACKING

By

DHUHA BT SHAMSUL IWARDI

FINAL 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

© Copyright 2006
by
Dhuha Bt Shamsul Iwardi, 2006

CERTIFICATION OF APPROVAL

CAMERA TRACKING

by

Dhuha Bt Shamsul Iwardi

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 2006

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.

Dhuha Bt Shamsul Iwardi

ABSTRACT

The Camera Tracking project is a system that captures images and tracks objects. The capturing of the image is done through a web-camera connected to a computer that runs the program of image capture and tracking. This project is applicable in various fields. It offers a low cost small surveillance system. The objective of the project is to develop a system that uses a web-camera that can also act as a surveillance camera that helps to monitor a desired object. The project deals with development of algorithm through the MATLAB software. The basic principle of the project is the phase correlation principle where the phase shift in the frequency domain is equivalent to the shift of the object in the spatial domain. The shift of the object is represented as a peak location at the phase correlation map. The two dimensional discrete Fourier Transform is used to decrease the number of computation in the digital signal analysis. Other image processing routine includes the Fast Fourier Transform and also Inverse Fast Fourier Transform.

ACKNOWLEDGEMENTS

Firstly I would like to praise Allah the Almighty, which have helped and guided me in completing my Final Year Project. With the completion of the Camera Tracking, I would like to express my word of appreciation. To my project supervisor, Mr. Patrick Sebastian I would like to show my greatest gratitude. With all the guidance, advices and cooperation he gave to me, it had helped me a lot in completing the project.

Heartiest appreciation goes to my family, especially my mother, Pn. Zuraimi Ismail and my grandfather, Tn. Hj. Ismail Jamil. I am deeply grateful for all the love, support and patience given. This token of appreciation also goes to Mr. Ku Amir Ferdaus Ku Azizan who has always encouraged me in completing this thesis. Thank you for being there when I needed you.

Last but not least, to all my friends who have walked through my 5 years of student life here in Universiti Teknologi PETRONAS, I am really honored for those great inspirations, love and support for completing my studies. The love and friendship offered is something that shall always be cherished and remembered.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND OF STUDIES

This project tackles the idea of having a system that performs tracking of objects and colors through a web-camera and the MATLAB software.

This project is a development of a system that performs image capture and tracking. Real-time visual image will be captured from a web camera and through Digital Image Processing Toolbox in MATLAB, the object from the image will be tracked as the object moves within the camera's view.

This Camera Tracking project comprises of 2 parts. The first part is the initial software (program) development which will be done in the first semester and the second part is the enhancement of the program. This project will go through a procedure that involves programming and hardware connection through Universal Serial Port (USB) of a computer with the web-camera.

Theoretical knowledge learnt previously in engineering field such as MATLAB Programming, Digital Signal Processing, and Image Processing are applied dynamically throughout the semesters.

1.2 PROBLEM STATEMENT

The wide usage of web camera with internet technology has proven to help connect people from one place and another. The existence of the web camera help human to communicate with one another. Interaction using web camera has benefited people all in all walks of life especially for families, friends as well as business activities.

The application of the project will help to enhance the value of communication through web camera as it can be used for face detection. This is due to the camera used for the purpose of web conferencing does not focus or placed on a correct view. As a result the other party may not be able to correctly see and view the image of the other person.

Moreover, this web-camera can be used as a simple surveillance camera to track an object that might be stolen. The practical approach is simple and low in cost as it does not involve with extra circuitry and wiring. This system also can be implemented in manufacturing industries where detection of components or objects from the production line can be done. The absence of the specific components or device can be monitored and tracked with integrating this basic system with the main control system of a factory.

To solve these shortcomings, the implementation of a moving web camera with additional image tracking features is proposed for future development. The application of a moving web-camera utilizes image processing techniques to have a web-camera placed at a specific location and tracks specific objects as they move within the viewing field of the camera.

1.2.1 Problem Identification

The camera tracking system performs object tracking by using a cross-hair embedded in the camera viewing window that can be monitored through the computer. This technology can be applied in various imaging situations. The principle of the object tracking in image processing is the process of comparing and correlating image captured by the web-camera with the reference image.

1.3 OBJECTIVES & SCOPE OF STUDY

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

- to be able to produce a MATLAB program that capture real-time images from web-camera
- 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.

The Camera Tracking project covers 2 main phases which are the image capturing algorithm development and object tracking algorithm development.

CHAPTER 2
LITERATURE REVIEW

2.1 PATTERN RECOGNITION

Pattern recognition deals with the recognition of objects in images, and is applicable to any other kind of data as well. A recognition system must contain some memory of the objects that it is to be recognized. Classification is the process of grouping objects together into classes according to their perceived likeness or similarities [1]. Pattern classification involves taking the feature extracted from the image and using them to classify image objects automatically. This is done by developing classification algorithms that use the feature information. The distance or similarity measures are used for comparing different objects and their feature vectors.

Based on Ram Rajgopal [1], correlation is the process of moving the reference image around the whole image area and computes the desired area's value. This idea is good but it is time consuming.

The normalized cross correlation is a technique for finding patterns in an image as long as the patterns in the image is not scaled or rotated. Typically, it can detect patterns of the same size up to a rotation of 5° to 10° [2]. If the rotation is larger, the pattern will not be detected. Extending correlation to detect patterns that are invariant to scale changes and rotation is difficult. For matching pattern that is in different scale, the process of scaling or resizing the template must be repeated until it is matched with each other and then correlation operation will be performed.

2.2 IMAGE SEGMENTATION

Image segmentation is a method to distinguish objects from background [1]. It covers both the use of spatial correlation and propagation high level information into segmentation. It derives from the information of the camera captured image and analyzes it. This method extract, categorize, arrange and connect visual features with contextual information to find a coherent segmentation.

For intensity images (i.e., those represented by point-wise intensity levels) three popular approaches are:

- threshold techniques
- edge-based methods
- region-based techniques

2.2.1 Edge Detection Method

Edge-based methods related on contour detection which the weakness in connecting together, broken contour lines resulting in prone to failure in the presence of blurring. There are a few methods of edge detection which are:

- Sobel Method
- Roberts Method
- Prewitt Method
- Canny Method
- Laplacian Method

An edge is a jump in intensity. The cross-section of an edge has the shape of a ramp. An ideal edge is a discontinuity of two section or area (i.e., a ramp with an infinite slope). The first derivative assumes a local maximum at an edge. For a continuous image $f(x, y)$, where x and y are the row and column coordinates, respectively, we typically consider the two directional derivatives $\partial_x f(x, y)$ and $\partial_y f(x, y)$. The main interests in edge detection are two functions that can be expressed in terms of these directional derivatives, the gradient magnitude and orientation. The gradient magnitude is defined as:

$$|\nabla f(x,y)| = \sqrt{(\partial_x f(x,y))^2 + (\partial_y f(x,y))^2} \quad \text{(Equation 2.1)}$$

The gradient orientation is given by:

$$\angle\nabla f(x,y) = ArcTan(\partial_y f(x,y) / \partial_x f(x,y)) \qquad \text{(Equation 2.2)}$$

Local maxima of the gradient magnitude identify edges in $f(x, y)$. When the first derivative achieves a maximum, the second derivative is zero. For this reason, an alternative edge detection strategy is to locate zeros of the second derivatives of *f(x,y)*. The differential operator used in these so-called zero-crossing edge detectors, is the Laplacian.

$$\Delta f(x,y) = \partial_{(x,2)} f(x,y) / \partial_{(y,2)} f(x,y) \qquad \text{(Equation 2.3)}$$

2.3 COLOR RECOGNITION

Colors are represented in 3 channels which are the Red (R), Green (G) and Blue (B) where each channels has a values that varies from 0 to 255. The combination of these three channels builds up a variation of colors.

Color			
RGB Values	RGB= [186 32 37]	RGB= [0 106 170]	RGB =[254 242 0]

Table 1 : Color representation in RGB

In color segmentation, the objective is to separate spatial regions of an image on the basis of similarity within each region and distinction between different regions. Application of color based segmentation ranges from empirical evaluation of various color spaces [7], to clustering in feature space [6].

The difference between color segmentation and color recognition is that the former uses color to separate objects without prior knowledge about the specific surface; the latter attempts to recognize colors of known color characteristics. Although the two problems are, the inverse of each other, results from segmentation can be useful in recognition; Maxwell [5] shows the advantages of using normalized color and separating color from brightness.

2.4 PHASE CORRELATION

The phase correlation method measures the movement between the two fields directly from their phases. The principle underlying the phase correlation motion estimation is described below.

To find a characteristic feature from the first image within the second, the first image, ***a*** is compared with the reference image, ***b*** within a certain range. The resultant correlation surface $c(x,y)$ is:

$$c(x,y) = a(x,y) \otimes b(x,y) = A(x,y) \otimes B(x-\Delta x, y-\Delta y)$$

The frequency domain operation of the correlation process involves multiplication of spatial frequency terms.

$$c(x,y) = a(x,y) \otimes b(x,y) \Leftrightarrow A(u,v) \times B*(u,v)$$

$$\Leftrightarrow A(u,v) \times B*(u,v) e^{-j2\Pi(u\Delta x + v\Delta y)}$$

$$\Leftrightarrow |A(u,v)|^2 \; e^{-j2\Pi(u\Delta x + v\Delta y)}$$

$$s(x-\Delta x, y-\Delta y) \Leftrightarrow S(u,v) e^{-j2\Pi(u\Delta x + v\Delta y)}$$ (Equation 2.4)

The above equation confirms that the correlation surface has the maximum at location *(Δx, Δy)* which is spatial displacement.

In phase correlation, the cross-power spectrum, S*(u,v)* is normalized prior to inverse transformation, which results in only the phase term remaining. The phase term is called the phase array. The correlation surface, $c_p(x, y)$ is obtained after inverse Fourier Transform of the phase array and it consists of weighted delta function at a spatial location *(Δx, Δy)*. In practice, a weighted delta function is represented as a sharp peak.

This is stated as

$$c_p(x,y) = \partial(x-\Delta x, y-\Delta y) \Leftrightarrow \frac{S}{|S|} e^{-j2\Pi(u\Delta x + v\Delta y)}$$ (Equation 2.5)

The phase correlation surface is defined as

$$c_p(x,y) = F^{-1}\{\frac{AB^*}{|AB^*|}\} \quad \text{(Equation 2.6)}$$

Thus, to calculate the phase correlation of the image in respective frames, the Fast Fourier Transform (FFT) is first performed on the two corresponding images. The inverse of the multiplication of the one spectrum is the phase correlation. The spatial displacement (vector) of the captured image (from web-camera) can be calculated by finding the location of the highest peak (pulse) at the phase correlation surface map.

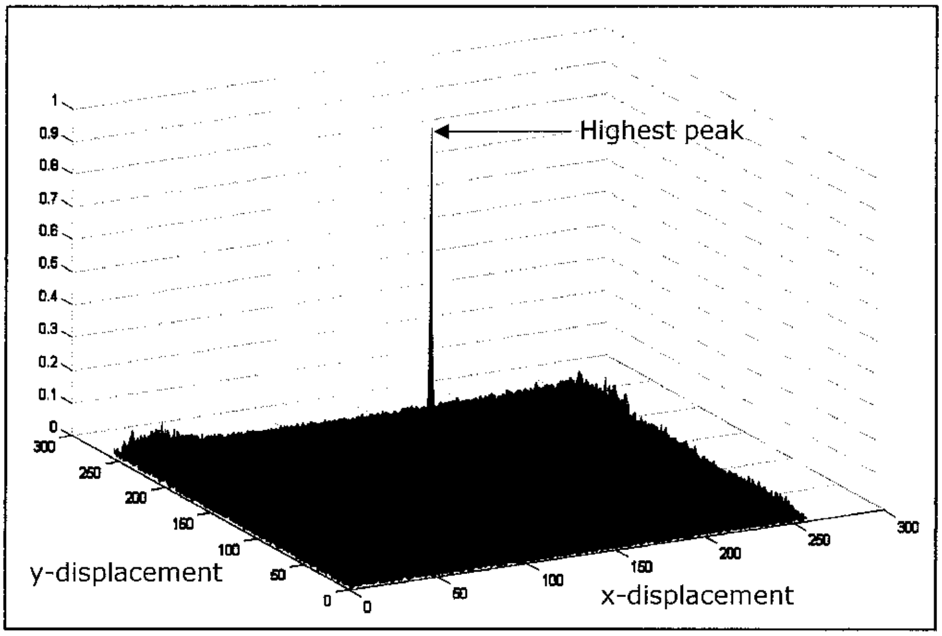


Figure 1 : Phase Correlation of Two Blocks

Figure 1 above shows the surface map of phase correlation where the highest peak is the location where both images are in phase.

Based on Yi Liang, [2], the implementation to the current frame is divided into 16 x 16 blocks in which the phase correlation calculation is performed for each block. In order to correctly estimate the cross correlation of corresponding blocks in respective frames, the blocks are extended. The main reason to this is due to low correlation for a particular motion due to the small overlapping area.

CHAPTER 3
METHODOLOGY

3.1 PROJECT FLOW

This project will go through the steps as in Figure 1 in next page. Basically the project starts with research and studies conducted based on journals, white papers books and articles obtain from both resource center and internet. The familiarization of the MATLAB software is done systematically in order to have easiness 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 consumed 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.

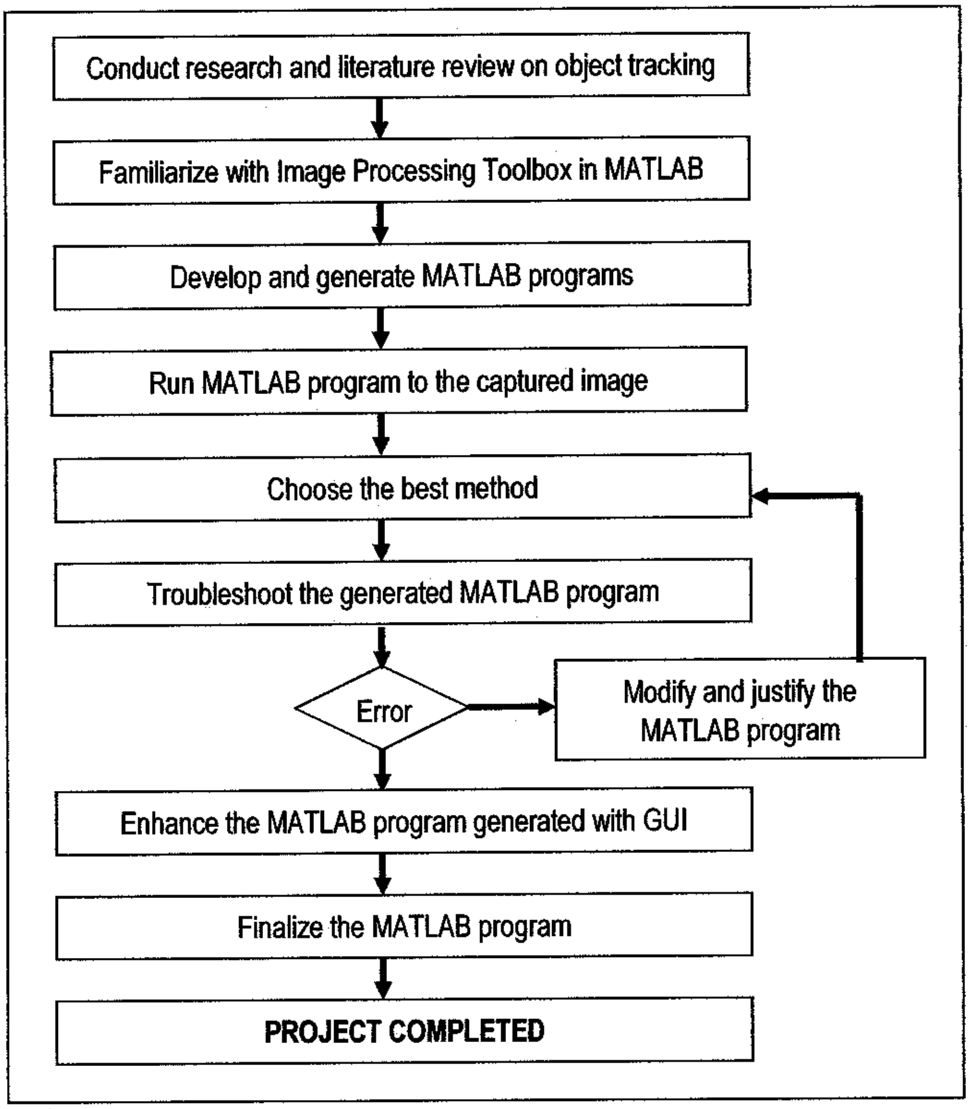


Figure 2 : Flow of Project

3.2 MATLAB PROGRAM FLOW

In this project two methods are approached to perform the object tracking. thus, two different MATLAB algorithms are developed. In this section, the flow of each MATLAB algorithm is discussed.

3.2.1 Phase Correlation Method

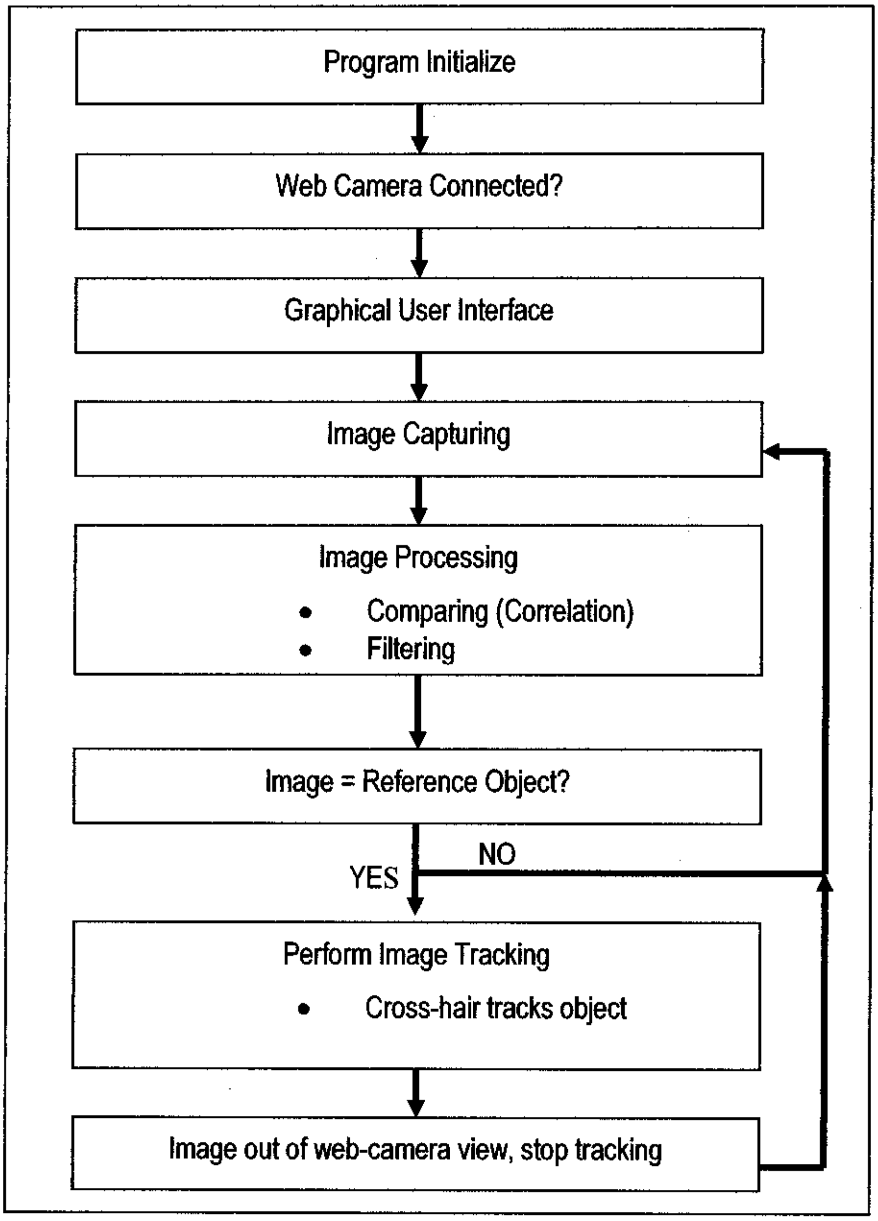


Figure 3 : Flow of MATLAB Algorithm

The flow of the program designed as in Figure 3. Firstly, the program is initialized. It will then check whether the web camera is connected to the computer. Next the image capturing will be executed through the web camera. Once the image is captured, it will be processed by correlating it with the reference image. The program will start tracking by having a cross hair appeared at the center of the object tracked through the web camera window. It will stop tracking when the object is out of the camera's field.

3.2.2 Color Recognition Method

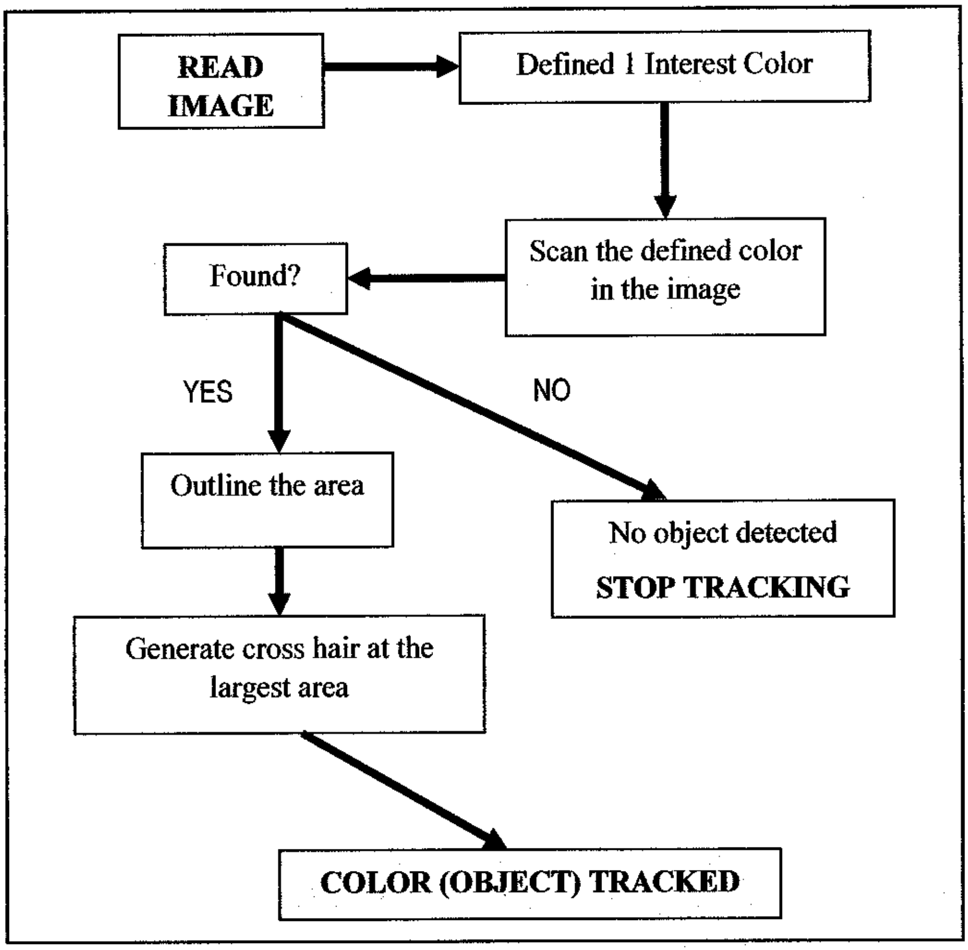


Figure 4 : Color Recognition Program Flow

For this particular method, the color is detected based on the predefined color. At first the image of the web-camera is read. The image is then scanned with the predefined color. Once the color is found in the image, the area of the color will be outlined. The largest outlined area will be located with a cross-hair at the center. This will complete the program by having the cross hair tracking the color (object). If there are no signs of the pre-defined color in the image, the program will end. Figure 4 in previous page shows the program flow of the Color Recognition Method.

3.3 SOFTWARES & TOOLS

The primary tool used for this project is the MATLAB software. This software is selected to model and simulate the algorithms for image capturing, processing and object tracking.

- Web Camera – Creative Vista Plus
- MATLAB 6.5 software - The software is for programs development
- MICROSOFT WORDS – Documentation and reporting purposes
- MICROSOFT VISIO – Application on project managing

CHAPTER 4
RESULTS AND DISCUSSION

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

The reference image used for the project is as in Figure 5 below which is a solid statue of a red cat. The image is resized to be 160 x 120 pixels to meet the resolution of the camera which is also 160 x 120 pixels.

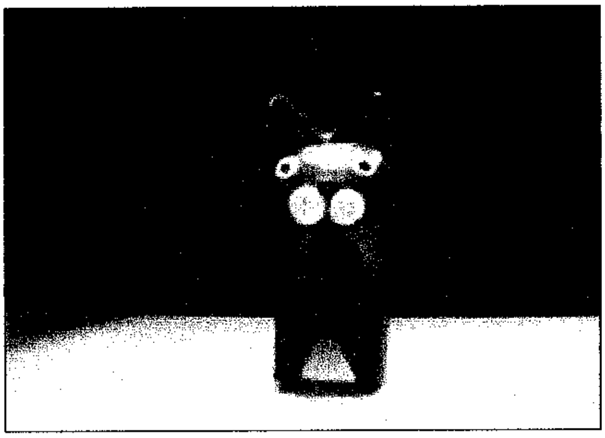


Figure 5 : Reference Image (catred. jpg)

As in the figure above, the location of the red cat is at the center of the image which can be said at pixel location of [80, 60]. The same object is utilized to be move around the web-camera viewing field in order to perform tracking capability.

4.1 PHASE CORRELATION METHOD

4.1.1 The modeling of Phase Correlation Algorithm

There main parts of MATLAB function used in the modeling of the algorithm are:

i. The MATLAB function ***fft2***.

   It is the two dimensional fast Fourier Transform routine which returns two dimensional discrete Fourier Transform computed with a Fast Fourier Transform algorithm.

ii. The MATLAB function ***ifft2***

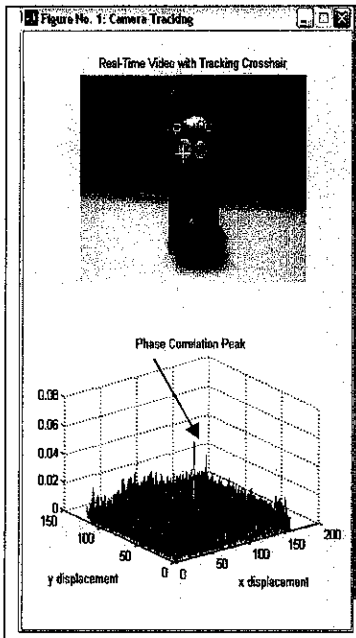
   It is the inverse two dimensional Fast Fourier Transform routine. Its return the two dimensional inverse discrete Fourier Transform computed with the FFT algorithm.

iii. The MATLAB function ***fftshift***

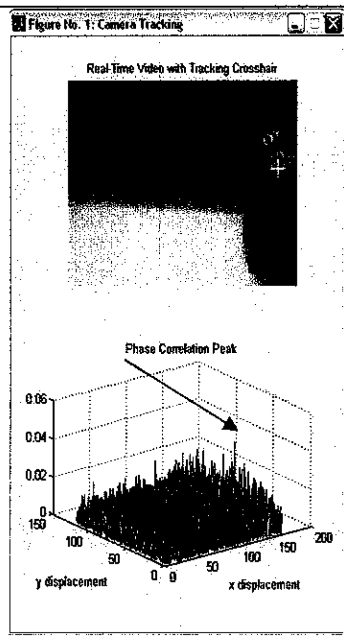
   It is used to rearrange the output by moving the zero frequency components to the center of array

The full designed algorithm of this method can be found in Appendix B.

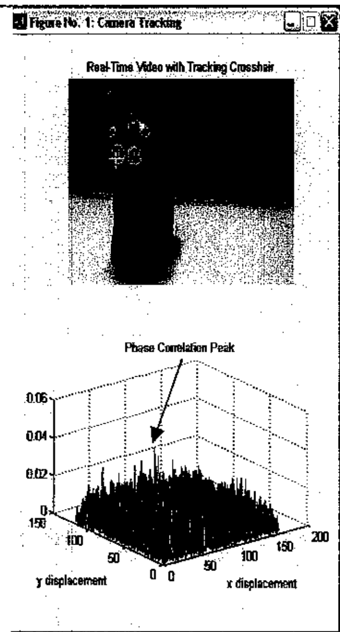
As seen in Table 2 below, the outputs of the algorithm has an embedded cross-hair. The cross-hair will follow the image captured that looks the same as the reference object. Based on the results obtained in the figures, the phase correlation map can be seen below the captured web-camera image. This surface map is for obtaining the peak values of the phase-correlated images. The object was moved left to right and the values of the peak also changed. The highest peak for each object position is pointed by the red arrow. These location of the object is calculated and it's displacement with respect to the reference image can be seen listed in Appendix B.



A: Object at Center



B: Object at Left



C: Object at Right

Table 2 : Output from file track.m

The surface maps in the figures also show that there are a lot of small spikes. These spikes indicate existence of noise while tracking process is done. The noises may occur due to the difference of intensity and brightness as the object is moved around. The brightness of the light plays a major role in order to perform phase correlation of the images. It affects the accuracy of the tracking.

4.2 COLOR RECOGNITION METHOD

Based on the color recognition program flow in Figure 4, red color is chosen to be the detected. The purpose of having red color is due to the object which is used to troubleshoot the programs is red in color. The RGB index of the color is [255 47 0].

The output from this MATLAB coding is shown in Figure 7. The MATLAB program generates one output figure. It consists of the web-camera image before and after it goes through the processing stage.

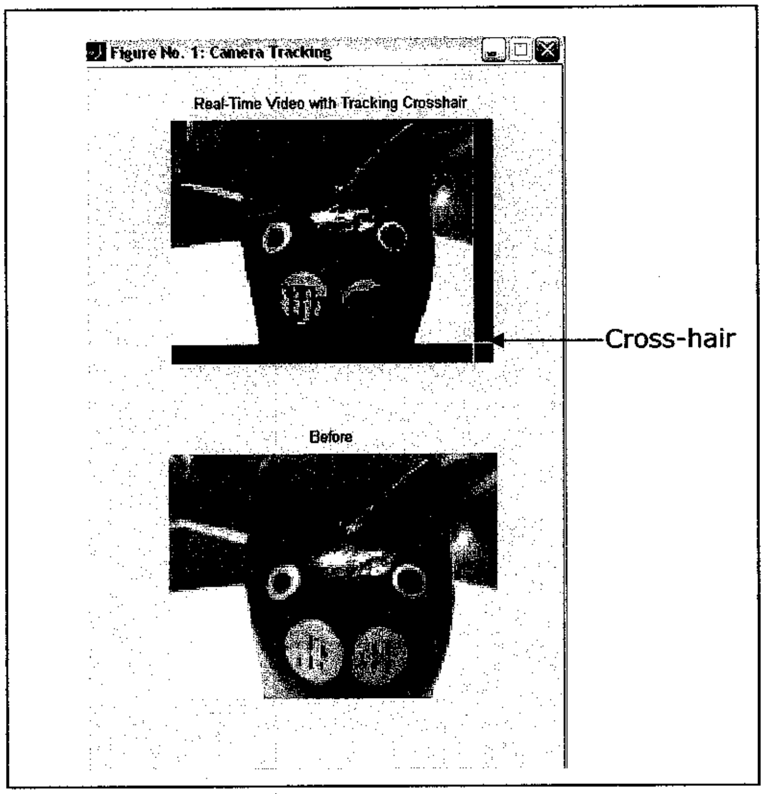


Figure 6 : Output from colorfindcamera.m

This method has an implementation of image segmentation where the edges of the red cat area after the color elimination process can be seen clearly.

CHAPTER 5
RECOMMENDATION

The project does have areas for upgrading. A few changes and modification can be made to have a more robust system. Currently the tracking is done with existence of noise. This can be seen from the surface maps in Table 2. Improving the performance by adding a filter is one of them. The addition of a filter will helped the tracking process to be more precise as it eliminates the noise of the image.

Further enhancement can be done by controlling the brightness level of the image. This can be achieved by performing background extraction. The extraction is done to eliminate the background image and making the camera to only project the object. This method can also minimize the noise in the image.

Currently the tracking system designed to track without moving the web-camera. The tracking is limited only to the camera's viewing field. An addition feature of moving web-camera to track objects can make the system more sophisticated and reliable.

A development of Graphical User Interface for the tracking system can also be made. This step will be taken in order to provide easiness for the user and also for smooth feature. The camera tracking system performance will be greatly improved with the additional features mentioned before.

CHAPTER 6

CONCLUSION

As a conclusion, this project is found to be interesting and beneficial. It gives a lot of experiences and knowledge in problem solving and troubleshooting skills. In addition, this project also gives a real practice on how to conduct an engineering project.

From this project, it is known that there are several mechanisms can be chose to design an object tracking system. After doing the analysis troubleshooting of and research, tracking system using phase correlation has been chosen. In this method, the object is tracked according to the phase of the images. If the two images are in phase, the highest peak will appear which indicates the location of the object in the camera's view.

Phase correlation method is chosen because of its simplicity and practicability for this project. However, there are still drawbacks on this method. Since brightness of the light plays a major role, the accuracy and precision of the tracking somehow affected. This can be seen once the reference image and the real image differ in terms of light intensity.

Nevertheless the tracking is still accomplished as the objectives of the project are met. Further expansion of the project can be made to improve and enhance the project to be more reliable and quality.

REFERENCES

[1] Ram Rajgopal, Pattern Matching Based on a Generalized Transform, Department of Electrical and Computer Engineering, University of Texas

[2] Yi Liang, Phase Correlation of Motion Estimation, Final Project Report, Stanford University

[3] Anzai Y. (1992). Pattern Recognition and Machine Learning. Academic Press

[4] S. Jehan-Besson, M. Barlaud and G. Aubert. (2002). A 3D-Step Algorithm Using Region Based Active Contours for Video Objects Detection. EURASIP Journal on Applied Signal Processingpg 572-581

[5] B.A. Maxwell and S.A. Shafer. (1994) A Framework For Segmentation Using Physical Models Of Image Formation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition

[6] Y. Ohta. (1980). A Region-Oriented Image-Analysis System by Computer. Doctoral Dissertation Kyoto University, Japan.

[7] B.J. Schachter, L.S. Davis and A. Rosenfeld. (1979). Some Experiments in Image Segmentation by Clustering of Local Feature Values, Pattern Recognition. pg 19-28.

[8] Marchand. (1996). Graphics and GUIs with MATLAB. CRC Press.

[9] The Mathworks, (2005) VideoCAPture by Gerox(c) for MATLAB 15 August 2005 from http://www.mathworks.com/

APPENDICES

APPENDIX A

COLOR RECOGNITION METHOD MATLAB CODES

i. colorfindcamera.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FYP2:CAMERA TRACKING
%DHUHA SHAMSUL IWARDI 2860
%COLOR RECOGNITION METHOD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function color_find
%global color_image;
%load color_image;

%load pinref;
success = 0;

% The image size
imrow = 280;
imcol = 240;

framerate = 30 % in frame/s
disptime =5 ;% in seconds
peaksignif = 0.05 ;%fractional difference to consider peak
cursorsz =10; % cross hair size

subplot (211)
set(1,'Name','Camera Tracking');
set(1,'Units', 'normalized'); set(1,'Position',[0.55,0.1,0.4,0.8 ], 'Menubar','none', 'Resize','Off');
figure(1);


image = vcapg;
imshow (image);
title ('Real-Time Video with Tracking Crosshair')

loc = zeros(10,3);

% find img center
```

```
imrow = round(imrow/2);
imcol = round(imcol/2);


   pause(1/framerate) ;

tr=90; %input('Enter the THRESHOLD VALUE 1-99\n')

i=vcapg;

%image(cm);
%title('Color Map of the Selected Color');
figure(2);

[image,index]=find_color(i,[],tr,pixel_color);

pause(1);
rehash;
subplot(3,1,2);
imshow(i);
title('Before')

subplot(3,1,3)
imshow(image);
%plot3(image);
title('After')
  %axis('auto');
   % Give maximum in each column
   %[maxvals_row row] = max(pcout);

   % Give max of the one row
   %[maxval col] = max(maxvals_row);
r=size(image,1);
c=size(image,2);
   %r=row(col);
   %c=col;
    %image(r,c,1) = 255; %r
   %image(r,c,2) = 255; %g
   %image(r,c,3) = 0;   %b

   % Generate the cross hairs
```

```
for iter = 1:cursorsz
   coordr = r-iter;
   coordc = c-iter;

   if all([(coordr >0),(coordc >0)])
      image(r,coordc,1) = 255;   %r
      image(r,coordc,2) = 255;   %g
      image(r,coordc,3) = 0;     %b

      image(coordr,c,1) = 255;
      image(coordr,c,2) = 255;
      image(coordr,c,3) = 0;
   end

   coordr = r+iter;
   coordc = c+iter;

   if all([(coordr >0),(coordc >0)])
      image(coordr,c,1) = 255;
      image(coordr,c,2) = 255;
      image(coordr,c,3) = 0;

      image(r,coordc,1) = 255;
      image(r,coordc,2) = 255;
      image(r,coordc,3) = 0;
   end

end
figure(1)
set(1,'Units', 'normalized'); set(1,'Position',[0.50,0.1,0.4,0.8 ], 'Menubar','none', 'Resize','Off');
subplot(211)
imshow(image);
title ('Real-Time Video with Tracking Crosshair')

end
end
```

ii. function **find_color.m**

```
function [image,index]=find_color(image,map,tr,colors)
%[IMAGE,IND]=FIND_COLOR(A,MAP,THRESHOLD,COLOR VECTOR)

I=make_rgb(image,map);

if isempty(colors)%if colors in the input argument are given as empty to get the color from the mouse
input
disp('Right click on the color you want off the image with the mouse:') ;
disp('Processing');

colors=impixel(I); %this command get the color amp from the mouse click coordinate
%and gives the 1 x 3 array to the colors input.
close
end

[i j]=size(colors);
if i>1,%for single ow procesisng only..nt multiple row..at any time
   colors=colors(:,1);
end
[image,index]= getcolor(I,colors,tr);%tr is the treshold value..required;I is the image,color is the
clicked image

% figure
% imshow(image);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [image,index]=getcolor(I,color,tr)
[i j k]=size(I);
R=color(1,1);
G=color(1,2);
B=color(1,3);
%uint8->the input image..the size is 8 bits unsigned.for double,the size is 64 bit floating point
%give this command to check the amount of memory taken bt uint8 and double---

I=double(I);
% whos
% disp('Check the BYTE occupied by image and I');
mask=( abs(I(:,:,1)-R) <tr ) & ( abs(I(:,:,2)-G) <tr ) &( abs(I(:,:,3)-B) <tr );
%in the above example I(:,:,1),we are gettig the first color component R and subtracting the color
component R
```

```
%by doing so,we may have some negative values..its we are checking that..whether to mask the color
or not
%based in logical function..so after dng for R v,do for G and B respectively..
%( abs(I(:,:,1)-R) <tr ) in this all the values will b either TRUE or FALESE,so either 1 or 0..in the
matrix
I(:,:,1)=I(:,:,1).*(~mask);%multipling each color component R,G and B vth mask(complemented)
I(:,:,2)=I(:,:,2).*(~mask);
I(:,:,3)=I(:,:,3).*(~mask);
image=uint8(I);
index=find(mask==1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function I=make_rgb(image,map)
[i j k]=size(image);
if (~isempty(map)),
   I=ind2rgb(image,map); %if the image is indexed..only..isempty(map)will return
   %1 since for an rgb image ,map is zero,ans ~ will return 0
   I=im2uint8(I);
elseif (isempty(k)),
   I(:,:,2)=I; %for gray scale image only..since k will b zero
   I(:,:,3)=I(:,:,1);
else
I=image;%for rgb image ..it will have all i,j and k
end
```

APPENDIX B

PHASE CORRELATION METHOD MATLAB CODES

i. track.m

```
%=================================================
%FYP1: Camera Tracking
%Dhuha Shamsul Iwardi   2860
%=================================================
%Phase correlation of image captured by web-cam

clc


figure(1);
set(1,'Units', 'normalized'); set(1,'Position',[0.55,0.3,0.4,0.4 ]);


%load pinref;
success = 0;


% The image size
imrow = 280;
imcol = 240;


framerate = 30 % in frame/s
disptime =5 ;% in seconds
peaksignif = 0.05 ;%fractional difference to consider peak
cursorsz =5; % cross hair size
figure(1);
set(1,'Name','Camera Tracking');
set(1,'Units', 'normalized'); set(1,'Position',[0.55,0.3,0.4,0.4 ], 'Menubar','none', 'Resize','Off');


subplot (211);


image = vcapg;
imshow (image);
title ('Real-Time Video with Tracking Crosshair')
A=imread('catred.jpg');
DFTimg1 = fft2(A(:,:,2));
%imshow (A);
```

```
%figure (1);
%set(1,'Position', [0.55,0.1,0.8,1.5],'Menubar','none','Resize','Off');
loc = zeros(10,3);

% find img center
imrow = round(imrow/2);
imcol = round(imcol/2);

for m=1:framerate*disptime

  % Phase correlation of images
  pause(1/framerate) ;

  image = vcapg;

  %image{index} = image{index}(:, :,1);
  %DFTimg1 = DFTimg2; %fft2 (image{refindex});
  DFTimg2 = fft2 (image(:,:,1));



  %normalize
  %phase correlation
  DFTpc = DFTpc ./ abs(DFTpc);

  pcout = fftshift(abs(pcout));

  pcout = fliplr(flipud(pcout));
  figure(1);
  set(1,'Units', 'normalized'); set(1,'Position',[0.50,0.1,0.4,0.8 ], 'Menubar','none', 'Resize','Off');


  subplot(212);
  colormap('default');
  mesh(pcout);
  xlabel('x displacement'); ylabel('y displacement');
  title ('Phase Correlation Peak');

  peak0=0;
  for y1=120
    for x1=160
```

```
    if(pcout(y1,x1)>=peak0)
      peak0=pcout(y1,x1);
      peak0x=x1;
      peak0y=y1;
    end
  end
end

%second highest peak
peak1=0;
for y2=1:120
  for x2=1:160
    if(pcout(y2,x2)>=peak1 & pcout(y2,x2)< peak0)
      peak1=pcout(y2,x2);
      peak1x=x2;
      peak1y=y2;
    end
  end
end

if(peak0x==80 & peak0y==60)
  peakx=peak1x;
  peaky=peak1y;
else
  peakx=peak0x;
  peaky=peak0y;
end

dx1=pcout(peaky-2,peakx-2)+pcout(peaky-2,peakx-1)+pcout(peaky-2,peakx);
dx2=pcout(peaky-1,peakx-2)+pcout(peaky-1,peakx-1)+pcout(peaky-1,peakx);
dx3=pcout(peaky,peakx-2)+pcout(peaky,peakx-1)+pcout(peaky,peakx);

dy1=pcout(peaky-2,peakx-2)+pcout(peaky-1,peakx-2)+pcout(peaky,peakx-2);
dy2=pcout(peaky-2,peakx-1)+pcout(peaky-1,peakx-1)+pcout(peaky,peakx-1);
dy3=pcout(peaky-2,peakx)+pcout(peaky-1,peakx)+pcout(peaky,peakx);

nx1=dx1*(peaky-60);
nx2=dx2*(peaky-59);
nx3=dx3*(peaky-58);
ny1=dy1*(peakx-80);
ny2=dy2*(peakx-79);
```

```
ny3=dy3*(peakx-78);

Ydisp=(nx1+nx2+nx3)/(dy1+dy2+dy3);
Xdisp=(ny1+ny2+ny3)/(dx1+dx2+dx3);

fprintf('\nThe magnitude of the highest peak is %2.3f pixel\n',peak0)
fprintf('The X displacement of the object is %3.2f pixels\n',Xdisp)
fprintf('The Y displacement of the object is %3.2f pixels\n',Ydisp)
% Give maximum in each column
[maxvals_row row] = max(pcout);

% Give max of the one row
[maxval col] = max(maxvals_row);

r=row(col) ;

c= col ;

image(r,c,1) = 255; %r
image(r,c,2) = 255; %g
image(r,c,3) = 0;   %b

% Generate the cross hairs
for iter = 1:cursorsz
  coordr = r-iter;
  coordc = c-iter;

  if all([(coordr >0),(coordc >0)])
    image(r,coordc,1) = 255;   %r
    image(r,coordc,2) = 255;   %g
    image(r,coordc,3) = 0;     %b

    image(coordr,c,1) = 255;
    image(coordr,c,2) = 255;
    image(coordr,c,3) = 0;
  end

  coordr = r+iter;
  coordc = c+iter;

  if all([(coordr >0),(coordc >0)])
```

```
        image(coordr,c,1) = 255;
        image(coordr,c,2) = 255;
        image(coordr,c,3) = 0;

        image(r,coordc,1) = 255;
        image(r,coordc,2) = 255;
        image(r,coordc,3) = 0;
    end

end
figure(1)
set(1,'Units', 'normalized'); set(1,'Position',[0.50,0.1,0.4,0.8 ], 'Menubar','none', 'Resize','Off');
subplot(211)
imshow(image);
title ('Real-Time Video with Tracking Crosshair')

%guidata(handles.output, handles);
% if handles.close > 0
%  m=framerate*disptime
end
```

ii. MATLAB Result for Figure 7A (Object at Center)	
The magnitude of the highest peak is 0.008 pixel The X displacement of the object is 81.13 pixels The Y displacement of the object is 60.88 pixels	The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 81.11 pixels The Y displacement of the object is 60.85 pixels
The magnitude of the highest peak is 0.014 pixel The X displacement of the object is 81.52 pixels The Y displacement of the object is 61.46 pixels	The magnitude of the highest peak is 0.012 pixel The X displacement of the object is 81.26 pixels The Y displacement of the object is 61.13 pixels
The magnitude of the highest peak is 0.012 pixel The X displacement of the object is 81.22 pixels The Y displacement of the object is 61.00 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.03 pixels The Y displacement of the object is 61.36 pixels
The magnitude of the highest peak is 0.012 pixel The X displacement of the object is 81.22 pixels The Y displacement of the object is 61.00 pixels	The magnitude of the highest peak is 0.009 pixel The X displacement of the object is 81.34 pixels The Y displacement of the object is 60.87 pixels
The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.20 pixels The Y displacement of the object is 60.94 pixels	The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.31 pixels The Y displacement of the object is 61.20 pixels
The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.20 pixels The Y displacement of the object is 60.99 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.34 pixels The Y displacement of the object is 60.85 pixels
The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.24 pixels The Y displacement of the object is 61.07 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.34 pixels The Y displacement of the object is 60.85 pixels
The magnitude of the highest peak is 0.015 pixel The X displacement of the object is 81.22 pixels The Y displacement of the object is 61.18 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.33 pixels The Y displacement of the object is 61.25 pixels
The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 81.13 pixels The Y displacement of the object is 60.85 pixels	The magnitude of the highest peak is 0.014 pixel The X displacement of the object is 81.25 pixels The Y displacement of the object is 61.25 pixels
The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 81.13 pixels The Y displacement of the object is 60.85 pixels	The magnitude of the highest peak is 0.015 pixel The X displacement of the object is 81.50 pixels The Y displacement of the object is 61.15 pixels

iii. MATLAB Result for Figure 7B (Object at Left)

The magnitude of the highest peak is 0.003 pixel The X displacement of the object is 80.70 pixels The Y displacement of the object is 61.00 pixels	The magnitude of the highest peak is 0.008 pixel The X displacement of the object is 81.07 pixels The Y displacement of the object is 60.94 pixels
The magnitude of the highest peak is 0.003 pixel The X displacement of the object is 80.70 pixels The Y displacement of the object is 61.00 pixels	The magnitude of the highest peak is 0.005 pixel The X displacement of the object is 80.79 pixels The Y displacement of the object is 61.04 pixels
The magnitude of the highest peak is 0.000 pixel The X displacement of the object is 80.47 pixels The Y displacement of the object is 60.99 pixels	The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 81.25 pixels The Y displacement of the object is 61.16 pixels
The magnitude of the highest peak is 0.000 pixel The X displacement of the object is 80.56 pixels The Y displacement of the object is 60.69 pixels	The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 81.02 pixels The Y displacement of the object is 61.21 pixels
The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 81.06 pixels The Y displacement of the object is 60.96 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.05 pixels The Y displacement of the object is 61.20 pixels
The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 80.66 pixels The Y displacement of the object is 60.93 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.33 pixels The Y displacement of the object is 61.37 pixels
The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 80.66 pixels The Y displacement of the object is 60.93 pixels	The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.18 pixels The Y displacement of the object is 61.22 pixels
The magnitude of the highest peak is 0.004 pixel The X displacement of the object is 80.73 pixels The Y displacement of the object is 61.16 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.27 pixels The Y displacement of the object is 61.20 pixels
The magnitude of the highest peak is 0.008 pixel The X displacement of the object is 80.90 pixels The Y displacement of the object is 60.98 pixels	The magnitude of the highest peak is 0.010 pixel The X displacement of the object is 81.27 pixels The Y displacement of the object is 61.20 pixels
The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 81.01 pixels The Y displacement of the object is 60.55 pixels	The magnitude of the highest peak is 0.013 pixel The X displacement of the object is 81.33 pixels The Y displacement of the object is 61.22 pixels

iv. **MATLAB Result for Figure 7C (Object at Right)**

The magnitude of the highest peak is 0.012 pixel The X displacement of the object is 81.21 pixels The Y displacement of the object is 60.95 pixels	The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 81.07 pixels The Y displacement of the object is 60.88 pixels
The magnitude of the highest peak is 0.013 pixel The X displacement of the object is 81.11 pixels The Y displacement of the object is 61.12 pixels	The magnitude of the highest peak is 0.008 pixel The X displacement of the object is 80.95 pixels The Y displacement of the object is 60.94 pixels
The magnitude of the highest peak is 0.014 pixel The X displacement of the object is 81.30 pixels The Y displacement of the object is 61.01 pixels	The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 81.01 pixels The Y displacement of the object is 60.97 pixels
The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.15 pixels The Y displacement of the object is 61.18 pixels	The magnitude of the highest peak is 0.002 pixel The X displacement of the object is 80.85 pixels The Y displacement of the object is 61.03 pixels
The magnitude of the highest peak is 0.005 pixel The X displacement of the object is 81.02 pixels The Y displacement of the object is 60.88 pixels	The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 80.95 pixels The Y displacement of the object is 60.94 pixels
The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 81.03 pixels The Y displacement of the object is 60.88 pixels	The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.18 pixels The Y displacement of the object is 60.96 pixels
The magnitude of the highest peak is 0.004 pixel The X displacement of the object is 80.86 pixels The Y displacement of the object is 61.12 pixels	The magnitude of the highest peak is 0.011 pixel The X displacement of the object is 81.18 pixels The Y displacement of the object is 60.96 pixels
The magnitude of the highest peak is 0.003 pixel The X displacement of the object is 80.74 pixels The Y displacement of the object is 60.88 pixels	The magnitude of the highest peak is 0.007 pixel The X displacement of the object is 80.98 pixels The Y displacement of the object is 60.91 pixels
The magnitude of the highest peak is 0.006 pixel The X displacement of the object is 80.97 pixels The Y displacement of the object is 60.83 pixels	The magnitude of the highest peak is 0.003 pixel The X displacement of the object is 81.06 pixels The Y displacement of the object is 61.03 pixels
The magnitude of the highest peak is 0.004 pixel The X displacement of the object is 80.92 pixels The Y displacement of the object is 60.95 pixels	The magnitude of the highest peak is 0.002 pixel The X displacement of the object is 81.00 pixels The Y displacement of the object is 60.85 pixels

APPENDIX C

PROJECT GANTT CHART

NO	ACTIVITIES / WEEK	SEMESTER 1 (JULY 2005)														
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
1	**Initialization of Project** • Project Overview • Discussion with Supervisor • Collection of books, articles and relevant information on project															
2	**Preliminary Report**															
3	**Image Capturing** • Study and familiarize with the software image processing functions. • Locate several techniques on object tracking and image capturing. • Develop program for each techniques in image tracking															
4	**Progress Report**															
5	**Troubleshooting** • Troubleshoot the program with a reference object • Determine the changes on program • Rectify the program based on the changes												●			
6	**Logbook**															
7	**Interim Report**															
8	**Oral Presentation**															

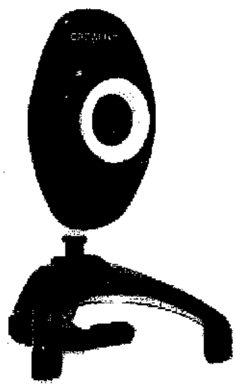
● = Finalize with tracking

NO	ACTIVITIES / WEEK	SEMESTER 2 (JANUARY 2006)														
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
1	**Continuation with project work**															
2	**Logbook**															
3	**Defined the techniques/methods applied to project**															
4	**Development of Algorithms** • Locate several techniques on object tracking and image capturing. • Develop program for each techniques in image tracking															
5	**Progress Report**															
6	**Troubleshooting** • Troubleshoot the program with a reference object • Determine the changes on program • Rectify the program based on the changes															
7	**Planning of Documentation**															
8	**Submission of Final Report (Soft Cover)**															
9	**Oral Presentation**	SCHEDULED FROM 6 JUNE – 9 JUNE														
10	**Submission of Technical Report**	DUE ON 12 JUNE 2006														
11	**Submission of Final Report (Hard Bound)**	DUE ON 23 JUNE 2006														

APPENDIX D

WEB-CAMERA SPECIFICATION

Home > Products > Web Cameras > WebCam



CREATIVE WEBCAM ***Vista Plus***

Add live video to your Instant Messaging!

Check Product Availability

Features	Specifications	Software	Requirements
Package Contents	Email Page	Print Page	

SPECIFICATIONS

Feature	**Description**
Sensor:	CMOS
Highest Still Image Resolution:	352 X 288
Highest Video Resolution:	352 X 288; 30 fps
Focus:	Adjustable manual focus ring
Interface:	USB 1.1 (Compatible with USB 2.0)

* Some of the products or bundled products featured here may not be available in your region or country. Please check with your local region site for product specifications and availability.

2006 Creative Technology Ltd. All rights reserved.
ivacy Policy | Terms of Use

CREATIVE

<IMG BORDER="0" NAME="DCSIMG" WIDTH="1" HEIGHT="1" SRC="http://wa.creative.com/njs.gif?dcsuri=/nojavascript&WT.js=No">