

# **TRACKING OF RIGID OBJECTS**

by

**YEO HANN**

Interim report submitted in partial fulfillment of  
the requirements for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

© Copyright 2010

by

Yeo Hann, 2010

# **CERTIFICATION OF APPROVAL**

## **TRACKING OF RIGID OBJECTS**

by

Yeo Hann

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

Approved:

---

Mr. Patrick Sebastian  
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

June 2010

## **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.

---

Yeo Hann

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my family for their moral support and guidance in helping me undertake challenges in life encountered.

I would like to express my appreciation and gratitude my supervisor, **Mr. Patrick Sebastian** for the valuable knowledge and experience he has taught me. He has been a great role model and inspiration in completion of this project.

Next, I would like to thank the lecturers who have taught me in other fields of study. The core foundation and basics of electrical and electronic engineering have been sowed in me through their sharing and passing down of knowledge.

Last but not least, I would like to express thanks to all friends and individuals who have helped me directly or indirectly in this project. The kind considerations of sharing expertise and skills have benefited me both in this project as well as life's learning cycle.

## **ABSTRACT**

This report mainly focuses on the basic theories and approaches used in background subtraction. An insight to background subtraction and its basic theory are discussed. Some problems related to previous background subtraction methods and limitations are also discussed. Theories are also linked with the algorithms that will be used. Also included is a literature review on previous work related journals. Discussion on the models and approaches used are included. Implementations of algorithms on top of the approaches are also defined with further clarity. Introductions to the methodology used are discussed. Discussions of expected results are determined together with objectives and goals for further implementations of results obtained.

## LIST OF FIGURES

Figure 1: Process of Background Subtraction.....	5
Figure 2: Sample Results Using Proposed Method.....	9
Figure 3: Kalman Filter Iteration Process.....	12
Figure 4: RGB Colour Cube.....	14
Figure 5: CMYK Colour Model.....	14
Figure 6: Sample image used by MATLAB Image Processing Toolbox .....	15
Figure 7: Image of Coins After Binary Thresholding .....	16
Figure 8: Process Flow of Object Tracking .....	17
Figure 9: Flowchart for Colour Recognition Process .....	18
Figure 10: Flowchart for Shape Recognition Process .....	19
Figure 11: Initial Starting of Video (Background Model).....	21
Figure 12: Results of Background Subtraction on Figure 11.....	21
Figure 13: MATLAB Performing Background Subtraction .....	22
Figure 14: Original Input Image for Kalman Filtering .....	23
Figure 15: Measured Values of Ball Movement .....	23
Figure 16: Kalman Filter Performing Prediction and Correction .....	24
Figure 17: Graph of Prediction and Measured Values .....	24
Figure 18: Original Image of Red Target .....	25
Figure 19: Identifying Pixels Belonging to the Red Band .....	26

Figure 20: Histogram of red band .....	26
Figure 21: Obtaining Red Objects Mask .....	27
Figure 22: Masked of Only Red Objects .....	28
Figure 23: Example of Multiple Shapes with Different Colours .....	28
Figure 24: Mask of Only Red Objects with Multiple Shapes (RGB Model) .....	29
Figure 25: Mask of Only Yellow Objects with Multiple Shapes (HSV Model) .....	29
Figure 26: A Multi Coloured Ball Example .....	30
Figure 27: Red Band Mask With Low Threshold .....	31
Figure 28: Red Band Mask With High Threshold .....	31
Figure 29: Binary Image of Original after Threshold .....	32
Figure 30: Sample Image for Multi Colour .....	32
Figure 31: Color Recognition for RGB (Using RGB Model) .....	32
Figure 32: Color Recognition for Unspecified Colours (Using HSV Model) .....	34
Figure 33: Masked Out Detectable Shapes .....	35
Figure 34: Output of Colour Recognition .....	35
Figure 35: Pseudo Coloured Objects from Shape Recognition .....	36
Figure 36: Output Showing Green Triangle Tracked .....	39

## **LIST OF TABLES**

Table 1: Tabulated Results for Colour Recognition .....	33
Table 2: Tabulated Results for Shape Recognition .....	36
Table 3: Table of Successful Detection of Shapes and Colours .....	40



## TABLE OF CONTENTS

<b>ABSTRACT</b> .....	v
<b>LIST OF FIGURES</b> .....	vi
<b>LIST OF TABLES</b> .....	viii
<b>CHAPTER 1 : INTRODUCTION</b> .....	1
1.1 Background of Study.....	1
1.2 Problem Statement.....	2
1.3 Objectives.....	3
1.4 Scope of Study.....	3
<b>CHAPTER 2 : LITERATURE REVIEW</b> .....	5
2.1 Background subtraction basics.....	5
2.2 Background Subtraction Theories.....	6
2.3 Kalman Filter Theory .....	10
2.4 Colour Space .....	13
2.5 Shape Recognition .....	15
<b>CHAPTER 3 : METHODOLOGY</b> .....	17
3.1 Procedure Identification .....	17
3.1.1 Overall Process .....	17
3.1.2 Colour Recognition .....	18
3.1.3 Shape Recognition .....	19
3.2 Tools .....	20

<b>CHAPTER 4 : RESULTS AND DISCUSSION</b> .....	21
4.1 Background Subtraction Results .....	21
4.2 Kalman Filtering Results .....	23
4.3 Colour Recognition Results .....	25
4.4 Shape Recognition Results .....	34
4.5 Combination of Algorithms .....	37
4.6 Discussion .....	39
<b>CHAPTER 5 : CONCLUSION AND RECOMMENDATIONS</b> .....	41
5.1 Conclusion .....	41
5.2 Recommendations .....	42
<b>REFERENCES</b> .....	43

# CHAPTER 1

## INTRODUCTION

### 1.1 Background of Study

The importance of object tracking has been playing an important role in security surveillance and research purposes. In surveillance, much focus has been put into object tracking or foreign object identification. Motion tracking provides important information in determining any foreign or unwanted objects within a defined area.

However the use of surveillance is not only applied for security purposes but also in research. Researchers are able to use tracking to focus on test variables of interest. In short tracking of objects is not only in terms of detecting motion or an incoming foreign object but also being able to block out unwanted and unimportant information.

Background subtraction is a method used in tracking by comparing of 2 sets of image slices namely foreground and background. The background model will be determined by the algorithm to obtain the most stable image slice that does not have any movement or foreign objects introduced. Thus during comparison, the background subtraction algorithm takes an image slice from the video input stream and compares it with the background model. Any difference in the two slices will produce a foreground model which is generally moved targets.

Being able to detect rigid (does not change shape) moving targets solves the first half of the issue. The following methods that will be used are to segregate which movements in the foreground image is useful and valuable information. Certain information may not be of great use for example shadow movement or reflectance. On the contrary, information such as target shape and characteristic would provide starting points to information segregation.

## **1.2 Problem Statement**

The monitoring of behavior or activities involving changing information is an important aspect regarding image processing and video surveillance [1]. The ability to determine any undesired foreign objects within a specified area of interest has much importance especially in terms of security and observation purposes.

Determining foreign and changed information is the basic logical approach. However there is certain change detection that is undesirable. Take an example in a traffic tracking system where the object of focus is a moving car. There will be several moving objects such as leaves and shadow. The ability to filter out this information is essential to produce a genuine result of motion detection and object tracking.

Current surveillance systems are able to record but are not able to determine which portion is redundant. By having an intelligent algorithm, only specific tracking will be done and information can be filtered.

### **1.3 Objectives**

The main objective is to develop a working algorithm that is able to track a moving object in a video. As more objects are introduced, other applications and techniques are used to process the information extracted from the motion detection. The main components can be subdivided into 2 parts. First, an algorithm is developed in order to track an object where essential information required in determining a moving object is obtained.

To use colour recognition approaches as a selected target feature. Targets that are used will be having different colouring thus increases the capability of the algorithm to distinguish one object from another.

To perform shape recognition as a secondary characteristic distinguisher. By having additional feature recognition, the algorithm will be more intelligent in terms of variability of a target. Having colour and shape will complement each other to produce a more specific target.

To fuse all methods used to become a single algorithm that will have the intelligence to work on its own without much user intervention.

### **1.4 Scope of Study**

The scope of study falls under Intelligent Signal & Imaging cluster. Computer Vision and Image Processing has an important role in doing pre-processing towards the input video sequence and also post-processing on motion detection outputs. Functionalities of MATLAB are greatly used in generating an algorithm for a per-pixel based approach on a given video input sequence. Mathematical understandings such as Gaussian distribution (probability) and basic statistical approaches (standard deviation and variance) are used widely.

Colour recognition dwells deep in types of colour space that is available in the current industry. RGB is one of the most common examples. The combination of Red, Green, and Blue colours allows many electronic displays to produce a wide range of colours. Colour space information is important in deciding the feasibility of colour detection. Shape recognition will deal mostly with pixel area together with shape characteristics such as centroid, edges, perimeter and etc.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Background subtraction basics

Background subtraction is an algorithm used much related to surveillance and monitoring. The algorithm involves detecting motion or significant differences inside of a video frame. What the algorithm does is it compares a background model with a foreground which in this case is an input stream using background differencing. With the difference of the two, any object that is not in the background can be detected thus creating an output mask.

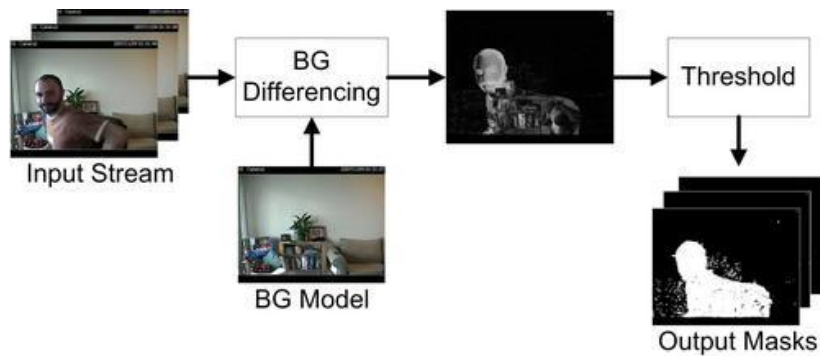


Figure 1: Process of Background Subtraction

There are mainly 3 approaches to background subtraction. First uses a statistical property of a background observed over a period of time to construct the background model. From this background model, a decision can be made which pixels in the foreground (input stream) do not fall into the background model. There is a base assumption in this approach that is the background remains unchanged in geometry, reflectance, and illumination.

Another approach is using image motion only assuming that a background is stationary. The background can also be at most very slowly changing but a target must be moving. No background model is required for this approach. Nevertheless, if the target stops moving there is no data that can be interpreted.

The last approach is based upon geometry. This approach uses an algorithm that performs real-time depth segmentation. The assumption is that the geometry of the background cannot vary. Requirements for this method however is robust and also require high computational power.

## **2.2 Background Subtraction Theories**

In this section, a more detailed description of how background subtraction models the image sequences is explained. Each pixel is modeled as a mixture of Gaussians. The persistence and variance of each of the Gaussians of the mixture will be used to determine which Gaussians belong to the background model. Pixel values that do not fit the background distributions will be considered as foreground until a Gaussian includes it with enough consistent evidence.

The background method uses two parameters which are  $\alpha$ , the learning constant and  $T$ , proportion of data that must be accounted for by the background. Consider values of a particular pixel over time as pixel process. Pixel process will be a time series of pixel values. What is known about a particular pixel at any given time will be its previous values.



$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \quad (2.2 - 1)$$

$I$  is the image sequence

$\{x_0, y_0\}$  are the pixel location in the image

The history of each pixel is modeled by a mixture of  $K$  Gaussian distributions. The probability of observing the current pixel value will be

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2.2 - 2)$$

$K$  is the number of distributions

$\omega_{i,t}$  is an estimate weight of the data is accounted by the  $i^{th}$  Gaussian in the mixture at time  $t$

$\mu_{i,t}$  is the mean value of the  $i^{th}$  Gaussian in the mixture at time  $t$

$\Sigma_{i,t}$  is the covariance matrix of the  $i^{th}$  Gaussian in the mixture at time  $t$

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \quad (2.2 - 3)$$

The equation above is the Gaussian probability density function.

For computational reasons, the covariance matrix is assumed to be  $\Sigma_{i,t} = \sigma_k^2 \mathbf{I}$

Every new pixel value  $X_t$  is checked against the existing  $K$  Gaussian distributions until a match is found. A match is defined to be within 2.5 standard deviations of a distribution. This is an example of per pixel/per distribution thresholding. When none of the  $K$  distributions match, the least probable distribution is replaced with the current value as its mean value (initially high variance and low prior weight).

$$\omega_{k,t} = (1 - \alpha) \omega_{k,t-1} + \alpha(M_{k,t}) \quad (2.2 - 4)$$

Prior weights of K distributions at time  $t$  are adjusted as above.  $M_{k,t}$  will be 1 for matched distributions and 0 for others. Weights are normalized after this approximation. When there is a match, the new observations are updated to be

$$\mu_t = (1 - \rho) \mu_{t-1} + \rho X_t \quad (2.2 - 5)$$

$$\sigma_t^2 = (1 - \rho) \sigma_{t-1}^2 + \rho (X_t - \mu_t)^T (X_t - \mu_t) \quad (2.2 - 6)$$

Where  $\rho = \alpha \eta(X_t | \mu_k, \sigma_k)$

An advantage of this method of updating is when allowed to become part of the background; the existing model of the background is not destroyed. The original background remains until it becomes the  $K^{\text{th}}$  most probable.

Up to now, the equations have only described the on-line mixture model. Having the mixture model, the study now proceeds to the background model estimation. Generally, the background model should have the most supporting evidence and the least variance. To model the background, a method is needed for deciding which portion of the mixture model best represents the background model.

First, the Gaussians are ordered by  $\omega/\sigma$ . From the equation we see that as the Gaussian has a higher weight, the value increase. Having a lower variance also increases the value.

$$B = \operatorname{argmin}_b (\sum_{k=1}^b \omega_k > T) \quad (2.2 - 7)$$

Where  $T$  is the minimum portion of data should be accounted for background.

Now having the on-line mixture model and background model, a method is needed to identify foreground pixels in each new frame while updating each pixel's description. The foreground pixels can be segmented into regions by a connected components algorithm.

The discussed algorithm method is a basic approach to background subtraction which is sufficient to determine object movements in a particular position. It is able to note any sort of movement and track its line of path by adding colorization to the information obtain.

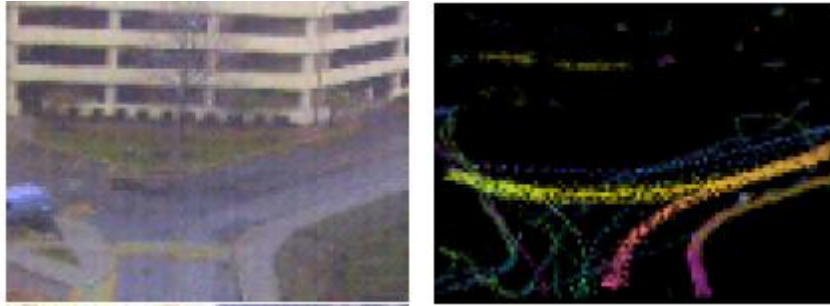


Figure 2: Sample Results Using Proposed Method

Image results for the proposed method. The background model is taken to be the image frame on the left. The image on the right is a display of all movements detected. The information is added with colour for each specific line of path. A mixture of two colours represents two objects moving in the same path.

The background subtraction approach can be divided into 3 main subtasks.

- 1) Obtaining the on-line mixture model
- 2) Building the background model
- 3) Determining the foreground using connected components algorithm

### 2.3 Kalman Filter Theory

Kalman filter uses measurements observed over time which may contain noise or random variations which evolves into inaccuracies for post-processing methods. The values that are estimated tend to be closer to the true values of measurements and calculated values. It uses a set of mathematical equations that works recursively in minimizing the mean of the squared error. The inputs to the filter are past, present, and also future states.

$$\widehat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \widehat{X}_{k-1} \quad (2.3 - 1)$$

$\widehat{X}_k$  = Current Estimation

$K_k$  = Kalman Gain

$Z_k$  = Measure Value

$\widehat{X}_{k-1}$  = Previous Estimation

The  $k$ 's are states. The purpose of the equation is to find  $\widehat{X}_k$  which is the estimate of signal  $X$  at each consequent  $k$ .  $Z_k$  is the measurement value which is not 100% certain of. The unknown component in the equation above is the Kalman Gain  $K_k$ . For simplicity purposes, the gain is assume to be 0.5 which will produce an averaging filter.

In implementing this filter, the conditions of Kalman Filtering must fit to a problem. Given that the 2 Kalman filter equations are

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2.3 - 2)$$

$$z_k = Hx_k + v_k \quad (2.3 - 3)$$

Signal values may be evaluated using linear stochastic equation (2.3 – 2). Any  $x_k$  is a linear combination of its previous value plus a control signal. Most of the time there is no control signal. Equation (2.3 – 3) shows that any measurement value is the linear combination of signal value and measurement noise. Noise in this case is taken to be Gaussian. When a system fits the model, the standard deviation of noise functions  $w_{k-1}$  and  $v_k$  are left to be estimated.

Initial values and parameters for the filter is important to get better estimates. The better the noise parameter estimation, the better the estimation. There are 2 main sets of equations which are Time Update used for prediction and Measurement Update used in correction. The equations are applied at k-th state. Time Update equations are as below:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (2.3 - 4)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.3 - 5)$$

Measurement updates equations:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (2.3 - 6)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2.3 - 7)$$

$$P_k = (I - K_k H)P_k^- \quad (2.3 - 8)$$

Matrices **A**, **B** and **H** are known from the initial modeling previously. What is left is to determine **R** and **Q**. **R** can be obtained from the noise in the environment the system is in.

The previous section deals with modeling a system of interest to a Kalman filter environment. This section is the equations used in iterations in Time and Measurement updates. Kalman filter works in a cycle where it performs a time update by first projecting the state ahead and then project the error covariance ahead. In the measurement update, the Kalman Gain is computed followed by updating the estimate using  $z_k$ . The error covariance is then updated producing outputs of k+1.

With the gathered information, the iteration can be done. Previous estimates will be the input for current state. The iteration works mainly on two cycles that is Time Update and Measurement Update.

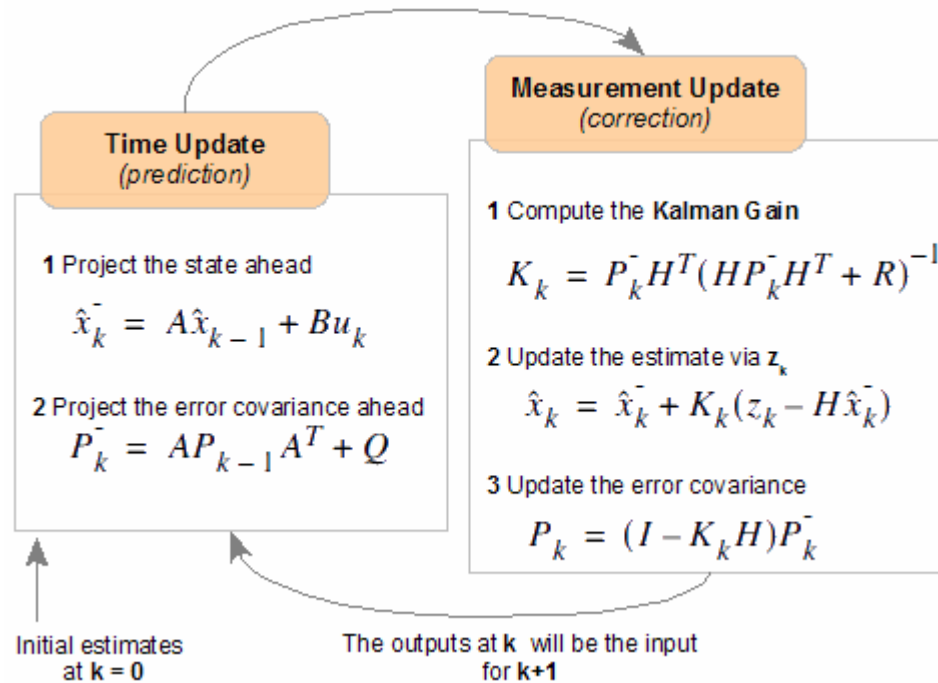


Figure 3: Kalman Filter Iteration Process

$\hat{x}_k^-$  is the rough estimation that is done at the first change of state (time update).  $P_k^-$  is also the prior estimation of error covariance. These prior values will be used in the measurement update iteration. In the measurement update iteration,  $\hat{x}_k$  is calculated and producing an estimate of  $x$  at time  $k$ .  $P_k$  is used for future estimates as well as  $\hat{x}_k$ .

## 2.4 Colour Space

A colour space has a range of colours that an electronic device can display or see [8]. This space is defined to provide devices such as computers for example, to have vision. Due to hardware limitation, there are only finite numbers of colours that machines can see or display thus requiring a convention for determining each unique colour produced.

Colour space provides a system of tabulating numerically each and every finite colour. A simple example can be taken where in a single axis that has a range predefined; a single tone of primary spectral component can vary from different shades. This is a simple form of 1-D colour space for a primary spectral component.

There are mainly 2 types of colour space that is related to each other, the RGB Colour Space which has Red, Green and Blue as its basic colour. Each colour will have its own axis line. In RGB Colour Space, a 3-D axis can be obtained producing a cube like feature. The intersection of maximum points of all 3 colours will produce white.

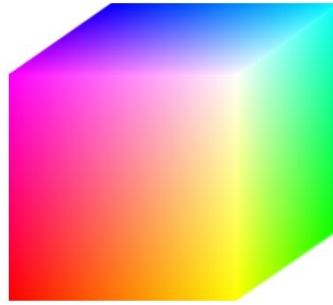


Figure 4: RGB Colour Cube

Figure 4 shows an example of the RGB Colour Cube. Its primary spectral components would be red which is located at the y-axis of the cube; green which is located on the x-axis; and blue on the z-axis. This model is based on a Cartesian coordinate system. For every plot of coordinate, there will be an assigned colour to it. This is how the RGB Colour Space works in facilitating the colours in a standardize method.

Another colour space that is used in the machine vision system is CMY Colour Model. The base colours used in this model are Cyan, Magenta and Yellow. These are the secondary colours of light or also known better for primary colours of pigments [8]. The main principle in this model is based on subtraction of the primary spectral component.

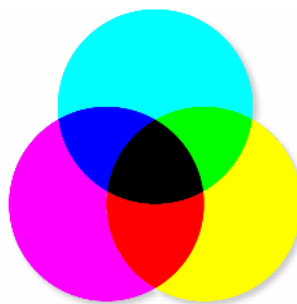


Figure 5: CMYK Colour Model

A major difference between RGB and CMYK is the combination of all 3 primary components producing white for RGB and black for CMYK.



## 2.5 Shape recognition

In this context, shape recognition is mainly for a specific number of shapes. The few shapes of interest would be of circular, rectangular or square, and triangle. The goal of an algorithm developed is to be able to differentiate the shapes from one another. There are two main methods that will be discussed in brief understanding in this section.

One way to determine objects is using blobs. A background having a constant colour is more suited for this method. White background for example is an ideal situation to provide the relative constant measure. When an object is placed on the background, the object is assumed to be a foreground. Thus foreground usually represents foreign objects introduced to the constant background environment.

Thresholding will be used to isolate the objects of interest. Differing the level of threshold can allow certain foreign objects with different characteristics to be isolated as well. A simple example for thresholding is binary thresholding.



Figure 6: Sample image used by MATLAB Image Processing Toolbox

Figure 6 shows a picture of coins placed in a constant coloured background. The image is a converted grayscale image. Using binary thresholding, a certain threshold is set that grayscale levels above the threshold are set to 1 and below set to 0. If the background level is known, the threshold value can be set to be just above the background value to eliminate the background and isolating the objects.

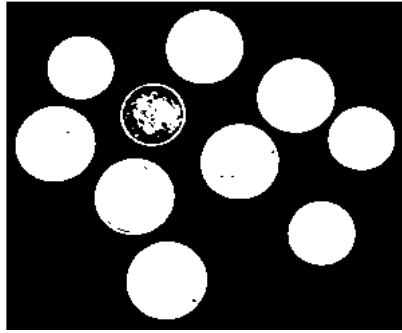


Figure 7: Image of Coins after Binary Thresholding

Figure 7 shows the resultant from using binary thresholding. The white areas are the extracted foreground. There are some areas within the coin that so happen to have the same grayscale level with the background thus is considered to be a background. Simple methods can be used to fill up the spaces within the foreground.

The produced results are called blobs. These blobs are the area of interest for further calculations to determine what shapes is being dealt with. Since the shapes being dealt with are simple basic shapes, the characteristics are easily spotted. For example a square will have four equal sides, circle having a constant radius and a triangle having 3 pointed corners. Calculations can be done on the blobs to produce a simple algorithm.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Procedure Identification

##### 3.1.1 Overall Process

Background subtraction provides the information of where and which pixels have detected motion.

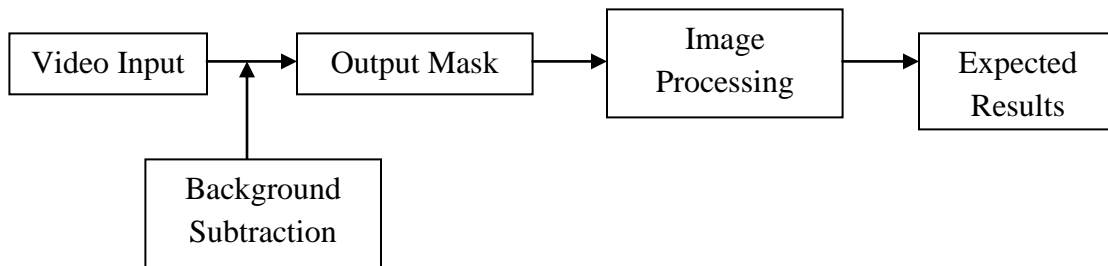


Figure 8: Process Flow of Object Tracking

To start off, a video input was taken having a blue and red coloured ball roll onto a white surface. A total of 3 different types of video inputs are obtained that are

- i) Blue ball rolling on a white surface
- ii) Red ball rolling on a white surface
- iii) Blue and Red ball rolling on a white surface

The main reason for using a white surface is to allow the video stream to have the most stable input it can have. It also helps eliminate unwanted noise and avoids introducing additional variables to the video input. A stable background model can also be established that the background subtraction algorithm is able to work with.

The result of background subtraction is the output masks of the motion detected is obtained. The masks provide the information needed in filtering out the stationary objects which had not changed. In the shape recognition stage, many different methods will be used to determine what are the objects being used.

### 3.1.2 Colour Recognition

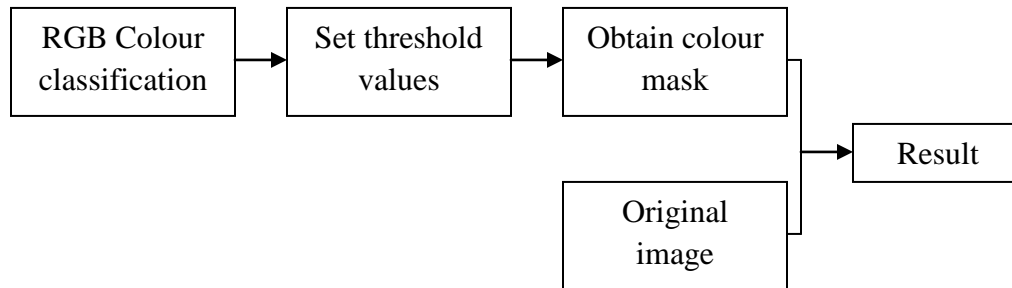


Figure 9: Flowchart for Colour Recognition Process

The first step in colour recognition involves classifying the image into its three primary spectral components, red, green and blue. To obtain this result, an algorithm that counts the number of pixels belonging to a particular spectral component is used. Determining whether a pixel falls within a spectral component requires a threshold value. This allows pixels that have the colour of interest be selected.

Obtaining a colour mask for targeted pixels only requires setting determined pixels belonging to a spectral component to be a binary image. By multiplying the colour mask with the original image, a resultant image can be obtained only showing pixels of a particular spectral component.

Another approach uses HSV colour space. This method emphasizes on tracking of yellow targets. The same steps are taken as the RGB space however with minute adjustments. The algorithm first changes the image into hue, saturation and value then setting a threshold to obtain a binary image. Thus at this stage has 3 binary images of hue, saturation and value. Taking the intersection of all three images will produce a yellow mask.

### 3.1.3 Shape Recognition

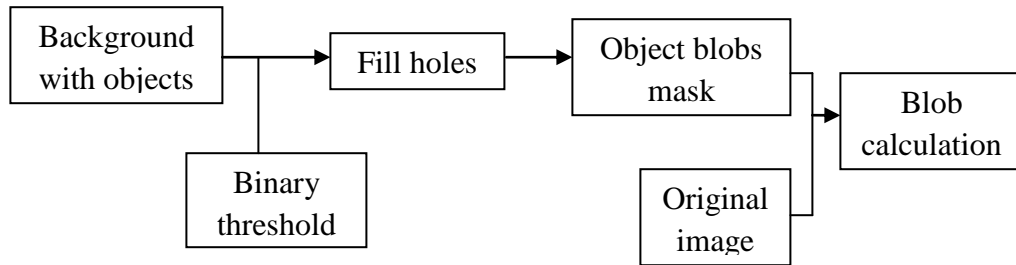


Figure 10: Flowchart for Shape Recognition Process

A still shot of a constant coloured or value background is obtained for a starting point. Using grayscale levels, the histogram of the background can be found and mean values are used to determine roughly the grayscale value of the background. When new objects are introduced, the histogram of the image will change thus producing values that lie outside of the background. Using a threshold value, there can be a cut of point for values that are within the range of the mean of the background.

A problem may occur when thresholding is used to eliminate the background to obtain only the objects blobs. The main problem is when the object has either partially or fully same values with the background. For the former, a simple processing method which is hole filling can be used. A hole is when a background region is surrounded by a connected border of foreground pixels [10]. This allows blobs of a constant region to be obtained.

Multiplying the masked blobs with the original image will produce an image showing only the objects introduced to the background. The main importance in this blob acquiring is to calculate the characteristics of the blobs to determine the shape of an object.

### **3.2 Tools**

The main tool used in all the calculations involved is MATLAB. MATLAB provides the platform for complex computations and performs iterative computations very well. Since all algorithms used are based on computations at every pixel, MATLAB is able to provide the computation power.

M-files which are scripts that MATLAB uses to execute the programmed algorithms are widely used. In programming an algorithm for both background subtraction and Kalman filtering, a function of desired operation is programmed and called to be used. MATLAB programming language has similarities to C++ programming however with differences that are specific as MATLAB reserves certain functions in their libraries.

Additional add-ons for MATLAB such as Image Processing Toolbox is also an added advantage. This toolbox provides predefined sets of instructions to aid image processing algorithms. Certain built-in functions are easier to be called from the toolbox than to write a new algorithm.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Background Subtraction Results

Initial testing of the algorithm has been done and the results are shown as below. The input video is a RGB video input of a simulation of air flow through a 3D room.

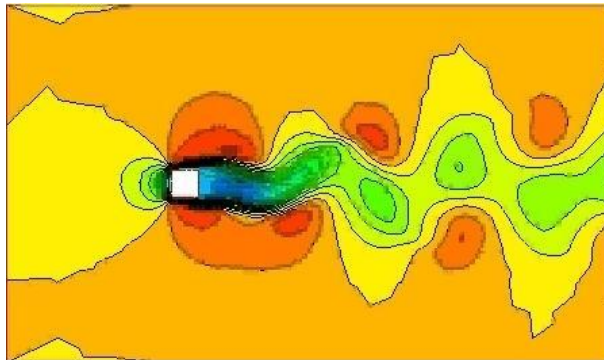


Figure 11: Initial Starting of Video (Background Model)

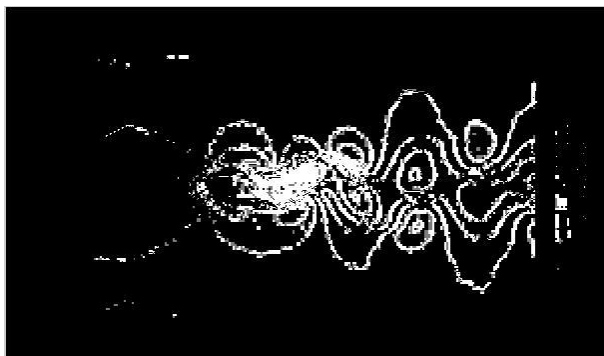


Figure 12: Results of Background Subtraction on Figure 11

Figure 12 shows a mask of the moved areas of Figure 11. The white areas are the areas that were computed to have changed position. The black areas on the other hand show no changes as we can on the upper and lower side of the figure there are plenty of untouched areas thus the output remains black.

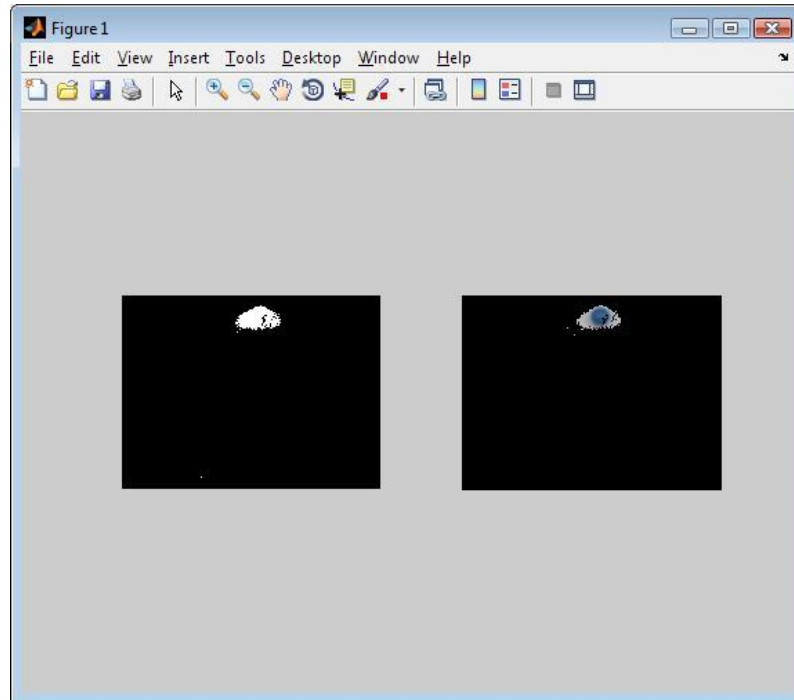


Figure 13: MATLAB Performing Background Subtraction

Figure 13 shows the output mask (left) and the result of multiplying the mask with original input. Since the blue ball is the only moving object in the entire frame at the particular time, only its portion of pixels will be masked. The mask is a little bigger than the object itself due to noise factors such as shadow. The ball which is moving is also casting a shadow on the background sheet thus the algorithm also masks out the moving shadow. This can be easily overcome by performing by setting a threshold of the shadow value to be passed.



## 4.2 Kalman Filtering Results



Figure 14: Original Input Image for Kalman Filtering



Figure 15: Measured Values of Ball Movement

Figure 14 shows the original input stream for a red ball falling from the top to bottom. Using the Kalman Filtering to perform its estimation and correction, Figure 15 displays the green ring of the measured value.

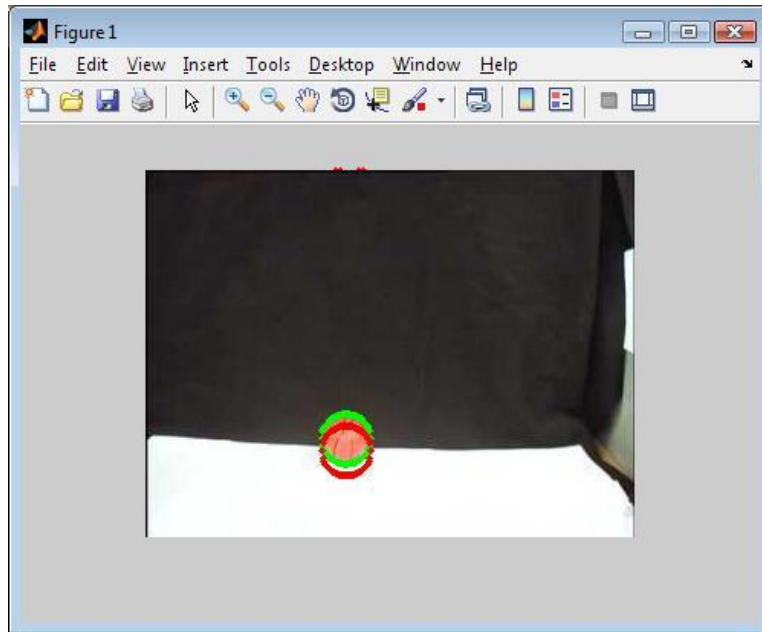


Figure 16: Kalman Filter Performing Prediction and Correction

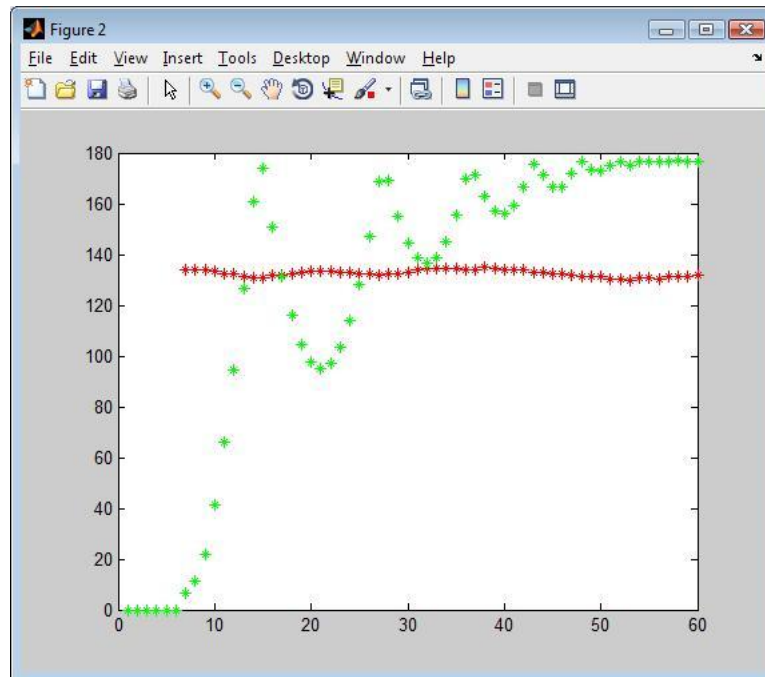


Figure 17: Graph of Prediction and Measured Values

The green ring in Figure 16 shows the measured values of the ball movement. Red ring shows the estimation/prediction values of ball movement simultaneously with measured values for comparison. Figure 17 displays the graph of predicted and measured values. There is a difference between the two values however the output from the filter produces a very good mask of the object. This mask would be useful for shape determination as it able to be as tight as possible with the object. Take for example for a circle, the masks would not have any sharp edges thus we can deduce the object is of a circular shape. A square shape would have four equivalent length edges and a triangular shape would have only three sides.

### 4.3 Colour Recognition Results



Figure 18: Original Image of Red Target

Figure 18 shows the original image used in extracting one of the single spectral components. The spectral component that will be used in this section is the red band. The aim is to be able to mask out only red coloured pixels in the original picture.

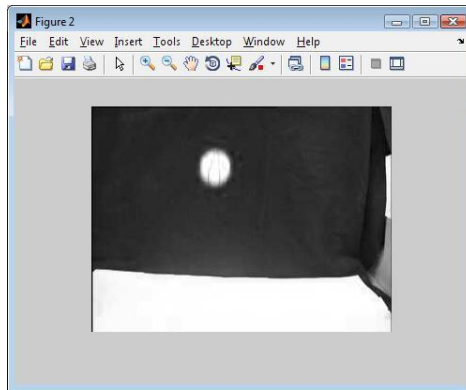


Figure 19: Identifying Pixels Belonging to the Red Band

Figure 19 is the result of the first step in colour recognition which is identifying pixels belonging to the red band.

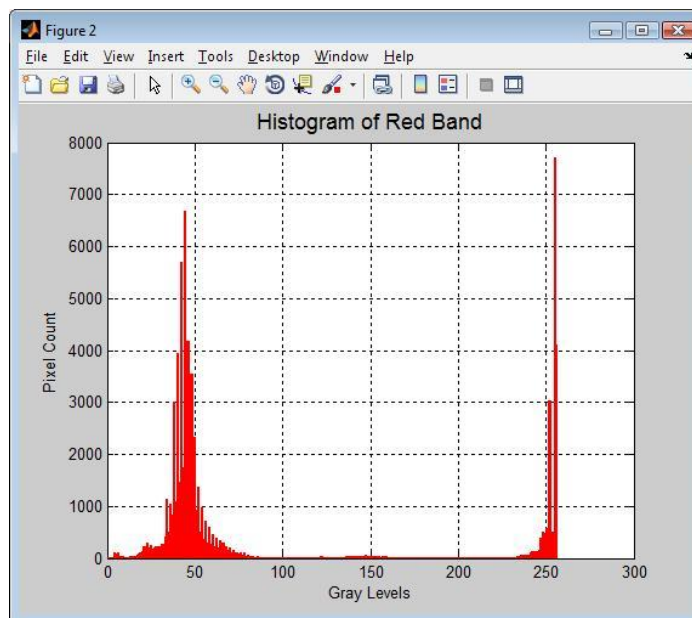


Figure 20: Histogram of Red Band

Figure 20 shows the pixel count of areas that belong to the red band. From analyzing the histogram, we can set a threshold to only select true red pixels. Since the histogram of the image is of the red band, the higher the value on the histogram are the values belonging to the red band. The low threshold values set are 85 and high to be 255.

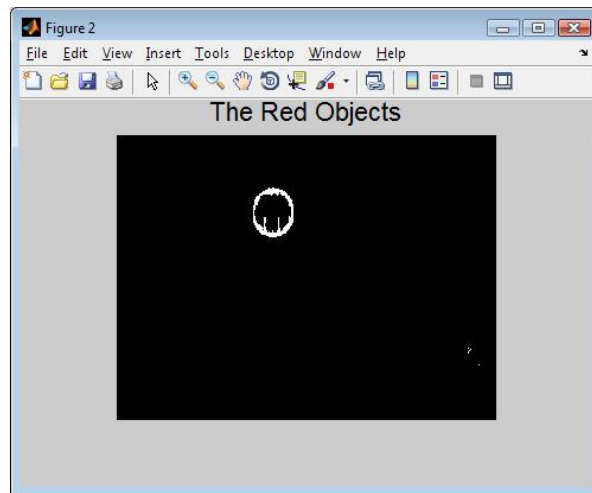


Figure 21: Obtaining Red Objects Mask

Figure 21 is after the thresholding process which will produce a binary result image. From Figure 21, some basic image processing functions will be used to smoothen the mask and to obtain a solid blob without any holes. Functions such as closing and opening are used to smoothen the edges and hole filling to produce a single white blob. Having a smooth and solid blob, the mask can be multiplied with the original image to get the object components from the original image.

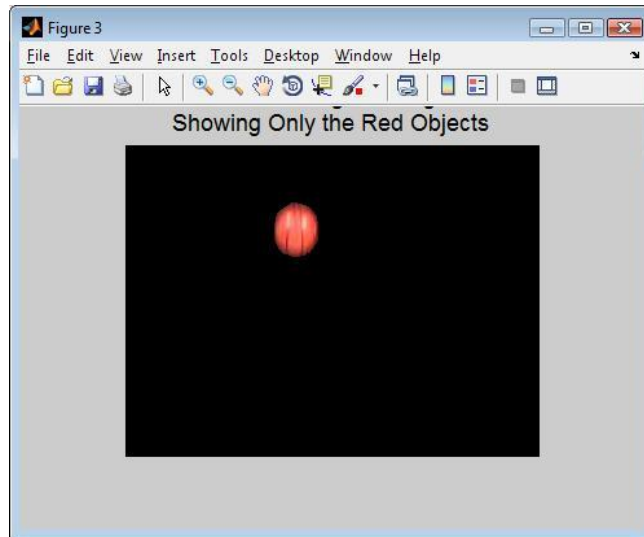


Figure 22: Mask of Only Red Objects

Figure 22 shows the results after multiplying the red object mask with the original image. Here we can see the object although has some white and black features it still belongs to the object.

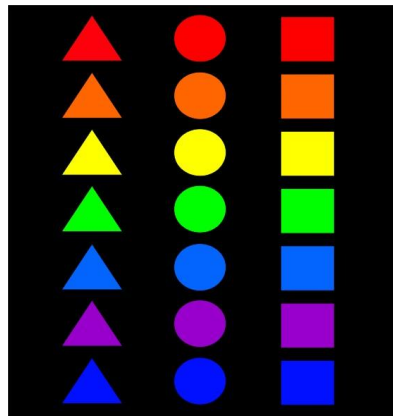


Figure 23: Example of Multiple Shapes with Different Colours

In Figure 23, shows another image of multiple objects with different colours. The algorithm is expected to work out which colours belong to which band of Red, Green or Blue. The output result would be showing only the red objects. This process of differentiating colours is also referred to as masking. Figure 24 shows the results of masking out red targets of the algorithm.

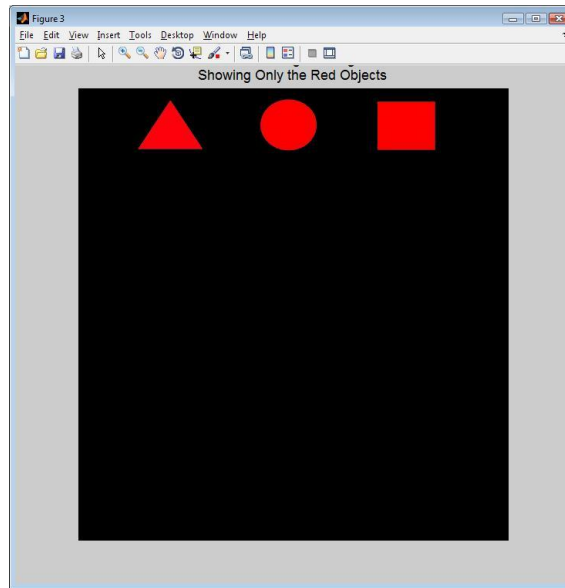


Figure 24: Mask of Only Red Objects with Multiple Shapes (RGB Model)

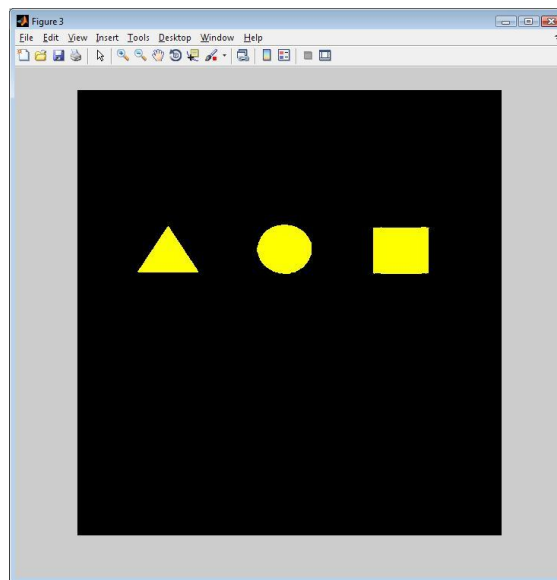


Figure 25: Mask of Only Yellow Objects with Multiple Shapes (HSV Model)

Figure 25 shows the output mask of the algorithm working on HSV colour space as introduced in section 3.1.2. The goal of using HSV colour space is to track colours other than the basic colour model of RGB.



Figure 26: A Multi Coloured Ball Example

Figure 26 shows the original image used as an input. This image is a good example to determine the role of threshold values. Since there are a few colours involved, the threshold value can be adjusted to give different precision on the pixels belonging to the desired colour bands.

Referring to Figure 20 as for the image histogram; when a low threshold (towards the left of the histogram) is used, more red band components are allowed into the output mask. A high threshold (towards the right of the histogram) will produce a more restricted output that only consists of higher values of the red band. Thus when performing simple masking between only two colours, a low threshold can be used since there are major differences in the two colours. However, when a situation that has multiple colours of almost the same shade a higher threshold can be used to provide more precision. A higher threshold value will only allow components that are close to actual band values to be in the output mask.





Figure 27: Red Band Mask with Low Threshold

From Figure 27, with low threshold values there are components belonging to other colours that pass as a red component. The threshold value set here is 15.



Figure 28: Red Band Mask with High Threshold

Figure 28 shows the components that are passed by the high threshold value. There are certainly less pixels compared to the Figure because of the high restriction settings of the threshold value. The threshold value set is 200.

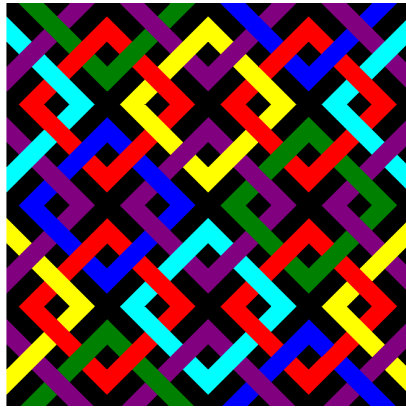


Figure 29: Sample Image for Multi Colour

Another image is taken as an example (Figure 29) which has a multi coloured texture. In Figure 30 and 31, shows the success of the algorithm determining the RGB and secondary colours using the base of RGB and HSV colour space.

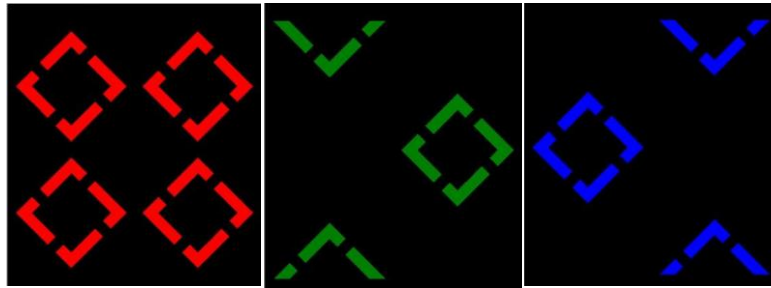


Figure 30: Result of Color Recognition for RGB (Using RGB Model)

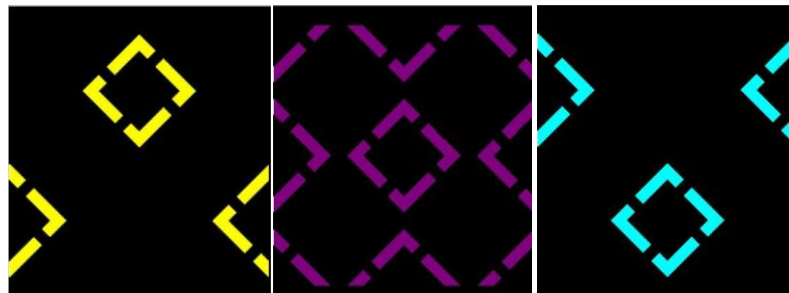


Figure 31: Result of Color Recognition for Unspecified Colours (Using HSV Model)

Colours Introduced	Success
Red	✓
Green	✓
Blue	✓
Yellow	✓
Purple	✓
Light Blue	✓
Pink	✓
Light Purple	✓

Table 1: Tabulated Results for Colour Recognition

#### 4.4 Shape Recognition Results

In the first procedure, the original image is checked with a threshold value to produce a binary image. The threshold value of 100 gray level is selected.

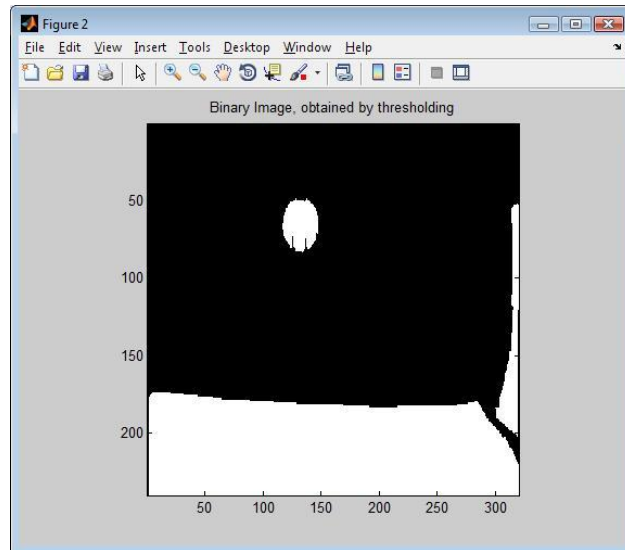


Figure 32: Binary Image of Original after Threshold

Figure 32 depicts a binary image of blobs present in the image. Since the situation is known to be using a circular object, smoothing techniques and hole filling are used in order to produce complete and solid blobs.

The blobs are in three different positions. This is an issue that can be solved with some experimental calculation of determining the maximum size of an object that is to be used. Maintaining the same camera angle and position, the object size in pixel dimensions is assumed to be relatively constant. Knowing the maximum pixel size of objects allows removing of blobs bigger than stated. This will produce blobs only related to the objects of interest and not constant regions in the images.

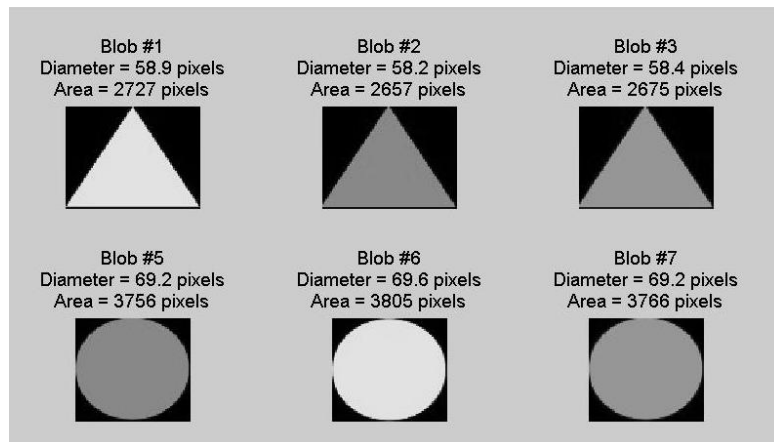


Figure 33: Masked Out Detectable Shapes

Figure 33 are the results of shape recognition using the previous algorithm. The original image is of Figure 23. Here the algorithm is able to count the diameter as well as the total area included. Take for example in this case that a circle and a triangle has been introduced. The algorithm can be told to ignore blobs of areas greater than 2700 pixels in order to only mask out triangles. The same can be done for masking out only a circle that is simply adjusting the range of having 3700 – 3850 pixels.

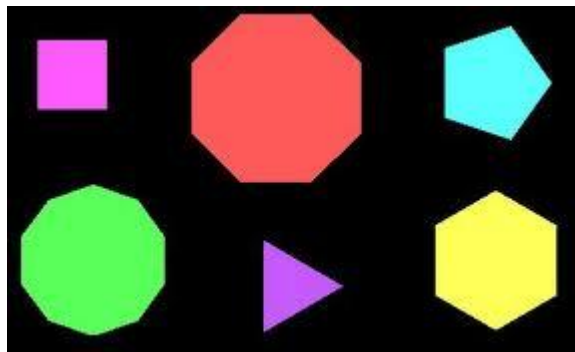


Figure 34: Shapes not Specified for Target Tracking

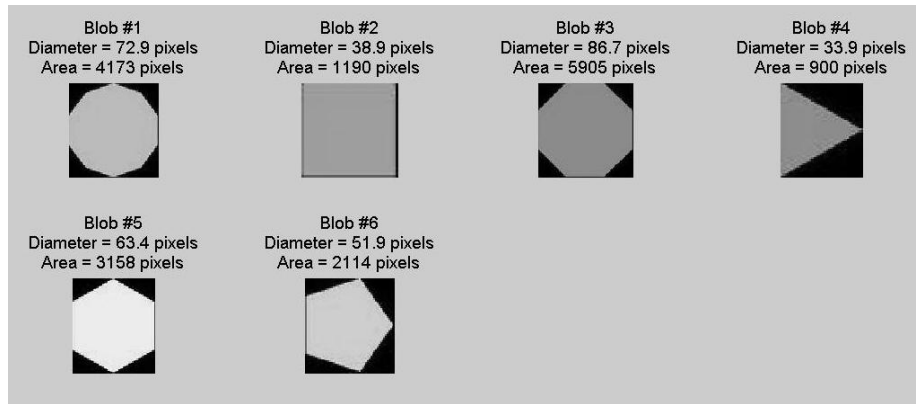


Figure 35: Shapes Recognized by Algorithm from Figure 34

Figure 35 shows the shapes able to be detected from the algorithm. Figure 34 introduces six shapes some specified earlier in discussion and some newly introduced. In Figure 35, blobs 1, 3, 5 and 6 are considered different from circular, triangle and square shapes. However the algorithm is able to realize the presence of blobs of different shapes within the area.

Common Shapes	Success	Uncommon Shapes	Success
Circle	✓	5 Sides	✓
Square	✓	6 Sides	✓
Triangle	✓	7 Sides	✓

Table 2: Tabulated Results for Shape Recognition

## 4.5 Combination of Algorithms

A fundamental assumption is that output masks from the background subtraction are still shot images. Thus it is possible to substitute the output mask of background subtraction with an example image such as Figure 23. The main reason for the substitution is to enable a wider variety of colours and shapes introduced to test the shape and colour recognition process. In the same time output masks from the background subtraction may have defects such as blurring due to lack of camera resolution since the target of interest is moving. In the earlier sections, background subtraction is proven possible and working to provide a sequence of output masks thus giving the viewer the perception that it is a video.

In this section, the main process to be tested would be shape and colour recognition working in serial sequence as one algorithm. First the sample image will be filtered with the colour recognition algorithm providing objects of the same colour but differing in shape. This output will then be passed to shape recognition algorithm which will determine which shapes are of interest. As a start, a green triangular shape is the target of interest.

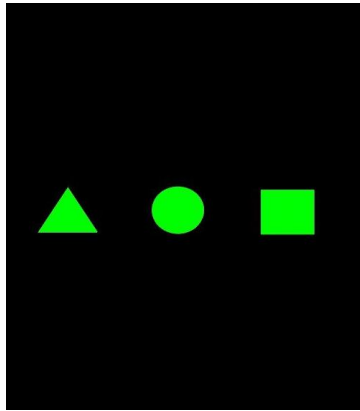


Figure 34: Output of Colour Recognition

Figure 34 shows the output obtained from the colour recognition algorithm. This output will be passed over to the shape recognition algorithm to determine which shape the algorithm will choose. Knowing from previous trials, a triangle has roughly about 2650 to 2750 pixel areas. The shape recognition algorithm is set to only pass objects that have a range of 2650 to 2750 pixel areas.

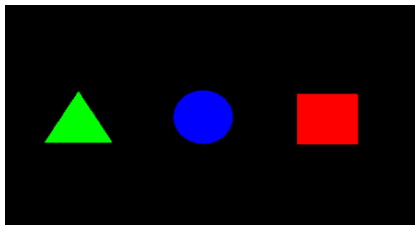


Figure 35: Pseudo Coloured Objects from Shape Recognition

In Figure 35, the algorithm clearly detects 3 different objects that are within the area. The colours that are present here are being pseudo coloured by the algorithm to differentiate that there are currently three objects in study. Knowing the three objects, the algorithm proceeds by filtering any pseudo coloured objects that do not have a pixel area ranging from 2650 to 2750 pixels.



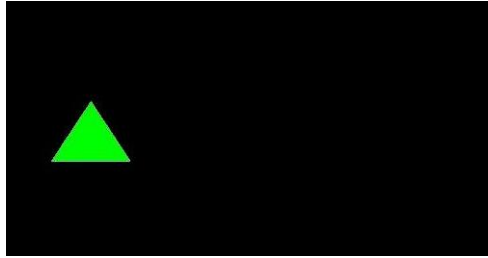


Figure 36: Output Showing Green Triangle Tracked

Figure 36 shows the tracked target of a green triangle present. From Figure 23 having many shapes and colours, the algorithm has segregated the different characteristics and features. All the different shapes or colours can be tracked by user specification. Using Figure 23 as a reference, a total of 21 objects can be tracked by the algorithm.

#### **4.6 Discussion**

Background subtraction is used in tracking moving objects. Any pixels that have changed more than the specified threshold will be categorized as a foreground image. This foreground image will be the output masks as shown in example of Figure 13. Background subtraction working by itself would not prove to be much of a beneficial algorithm due to the fact that in a video stream, many factors can produce a false movement. It can be said that background subtraction is the main algorithm which has to be performed in object tracking.

The determining of information of targeted objects is done separately. The basic approaches in determining shape and colour are done in separate occasions. Up to now, there are 3 approaches in the discussion. First the background subtraction model, colour recognition and shape recognition model.

Background subtraction model provides output masks that give only moving objects. It requires further processing if a user specification is intended. With shape and color recognition introduced, more variables that are not related to the specification can be eliminated.

With current models introduced, the fusion of all 3 models is required. There are a few different methods that have been introduced. Serial processing is where the video stream is loaded and processed by all 3 models frame by frame in a sequence. This approach is simple but may be time consuming and requires computational power. However the results are easy and simple to deal with.

Another approach would be parallel processing where a single frame is fed into 3 different models at the same time. The results from the separate models will then be evaluated using another algorithm.

Successful Colours Detected	Successful Shapes Detected
Red, Dark Green, Light Green, Dark Blue, Light Blue, Yellow, Purple	Circle, Triangle, Square, Rectangle

Table 3: Tabulated Success of Combination of Shapes and Colours

Table 1 shows the total number of colours detected and shapes detected. This far, the algorithm easily is able to determine the difference between these colours and shapes. The shapes those are able to detect do not depend on a specific target. Since the algorithm is working on a blob based approach, any particular shape or target can be introduced. However the target introduced cannot change its shape throughout the entire study time. Shapes recorded in Table 1 are of shapes that are comprehensive to human language. A user only needs to know the pixel area of a particular target of interest and making minute adjustments to the algorithm to segregate it.

## CHAPTER 5

### CONCLUSION AND RECCOMENDATIONS

#### 5.1 Conclusion

Up to this stage, many algorithms have been introduced. The direction of the algorithm up to this point is based on a modular approach where single characteristics of a target are determined one by one. The few important characteristics have been identified which are target movement, shape and colour. The algorithm is showing the intelligence that it requires by successfully segregating unwanted information and producing relatively sharp output masks.

Image Processing Toolbox (IPT) that is provided in MATLAB provides a good stepping stone in terms of programming. Basic and fundamental functions are in built to the IPT helps shorten and simplify coding methods. Processing time can also be shortened with a shorter code. This would be a major concern when dealing with a live input video stream.

The basic algorithm is completed in modules up to this stage. Fusion of the three modules is done producing a single standalone algorithm. The basics of the algorithm have been established however many improvements can still be done in terms of producing an output mask that is accurate and efficient. The algorithms used are also able to be minutely adjusted according to how sensitive it can be towards the target of interest.

## **5.2 Recommendations**

Improving the video quality will provide a stable environment to the background model. Thus having a high quality video input by using a High Definition video camera will help reduce the errors. Having higher definition of the video inputs will increase the number of frames for background subtraction. Therefore the more inputs will avoid any gaps in between each frame.

Having a higher performance computer would drastically improve the processing time of the algorithm. Since there is an objective to have real time video input, a higher and faster computing machine is required. Computations performed by the algorithm are repetitive and great in number due to per pixel computations.

## REFERENCES

- [1] Surveillance, Retrieved from <http://www.wikipedia.com>
- [2] Z. Zivkovic, F. Heijden, “Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction.” Faculty of Science of University of Amsterdam, July 2004.
- [3] Parks, D. H. and Fels, S, “Evaluation of Background Subtraction Algorithms with Post-processing,” September 2008
- [4] J. Song, “Background Subtraction,” The Connexions Project. December 2008.
- [5] Chris Stauffer, W.E.L. Grimson - Adaptive background mixture models for real-time tracking (Kalman Filter)
- [6] M. Tkalcic, Jurij F. Tasic, “Colour Space – perceptual, historical and application background,” University of Ljubljana, 2003.
- [7] R. Gonzalez, R. Woods, “Digital Image Processing Third Edition,” Person Education, 2010. pp.424-428.
- [8] Z. Zivkovic, “Improved Adaptive Gaussian Mixture Model for Background Subtraction.” Intelligent and Autonomous Systems Group, University of Amsterdam, 2004.