

**VIRTUAL MOUSE**

By

**WAN CHUNG SON**

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

**BACHELOR OF INFORMATION SYSTEMS (HONS)**

**INFORMATION SYSTEMS ENGINEERING**

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2016

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**VIRTUAL MOUSE**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

## **ACKNOWLEDGEMENTS**

I would like to express my deepest appreciation to my supervisors, Dr. Ng Hui Fuang who has given me this opportunity to engage in an image processing project, as well as provided me feedbacks and suggestion to make this project possible. Without his guidance this project would not have been possible.

I would like to thank a friend of mine, Yixian Quah, who have provided me suggestions on the materials needed to improve the effectiveness and the efficiency of the project. In addition, I would like to give my thanks to my family and friends who have supported and motivated me throughout the course of developing the project.

## **ABSTRACT**

This project promotes an approach for the Human Computer Interaction (HCI) where cursor movement can be controlled using a real-time camera, it is an alternative to the current methods including manual input of buttons or changing the positions of a physical computer mouse. Instead, it utilizes a camera and computer vision technology to control various mouse events and is capable of performing every task that the physical computer mouse can.

The Virtual Mouse colour recognition program will constantly acquiring real-time images where the images will undergone a series of filtration and conversion. Whenever the process is complete, the program will apply the image processing technique to obtain the coordinates of the targeted colours position from the converted frames. After that, it will proceed to compare the existing colours within the frames with a list of colour combinations, where different combinations consists of different mouse functions. If the current colours combination found a match, the program will execute the mouse function, which will be translated into an actual mouse function to the users' machine.

# TABLE OF CONTENT

<b>DECLARATION OF ORIGINALITY</b>	i
<b>ACKNOWLEDGEMENTS</b>	ii
<b>ABSTRACT</b>	iii
<b>TABLE OF CONTENT</b>	iv
<b>LIST OF FIGURES</b>	v
<b>LIST OF TABLES</b>	vi
<b>LIST OF ABBREVIATION</b>	vii

## **CHAPTER 1 PROJECT BACKGROUND**

1.1	Introduction	1
1.2	Review of the Physical Mouse	2
	1.2.1 Mechanical Mouse	2
	1.2.2 Optical and Laser Mouse	3
1.3	Problem Statement	4
1.4	Motivation of Virtual Mouse	5
	1.6.1 Convenient	5
	1.6.2 Cost Effective	5
1.5	Project Scope	6
1.6	Project Objective	6
1.7	Impact, Significance and Contribution	7

## **CHAPTER 2 LITERATURE REVIEW**

2.1	Visual Panel	8
2.2	Mouse Simulation Using Two Coloured Tapes	9
2.3	Visual Mouse Using a Webcam	10
2.4	Portable Vision Based Human Computer Interaction (HCI)	11
2.5	Conclusion	12

## **CHAPTER 3 SYSTEM DESIGN**

3.1	System Design	13
	3.1.1 Calibration Phase	14
	3.1.2 Recognition Phase	17
3.2	Application Layout	21
3.3	Implementation Issues and Challenges	25

## **CHAPTER 4 METHOD AND TECHNOLOGIES INVOLVED**

4.1	Methodology	27
4.2	Hardware and Software Requirement	30
	4.2.1 Hardware Requirement	30
	4.3.2 Software Requirement	31
4.3	Verification Plan	31

<b>CHAPTER 5 IMPLEMENTATION AND TESTING</b>	
5.1 Overview	35
5.2 Performance in Various Environments	35
<b>CHAPTER 6 CONCLUSION</b>	
6.1 Overview	39
6.2 Limitation	40
6.3 Future Works	40
<b>BIBLIOGRAPHY</b>	41
<b>SIMILARITY REPORT</b>	42

## LIST OF FIGURES

<b>Figure 1.1:</b>	Mechanical mouse, with top cover removed .....	2
<b>Figure 1.2:</b>	Optical Mouse, with top cover removed .....	4
<b>Figure 2.1:</b>	The system overview of Visual Panel .....	8
<b>Figure 2.2:</b>	The system architecture of the mouse-simulation .....	9
<b>Figure 2.3:</b>	Yellow colour tape for cursor movement .....	10
<b>Figure 2.4:</b>	Input image using one and two pointers .....	10
<b>Figure 2.5:</b>	The Flow Chart of Portable Vision-Based Human Computer Interaction .....	11
<b>Figure 3.1:</b>	Virtual Mouse Colour Recognition Block Diagram .....	13
<b>Figure 3.2:</b>	The comparison between un-filtered and filtered frame .....	14
<b>Figure 3.3:</b>	The comparison between RGB and HSV frame .....	15
<b>Figure 3.4:</b>	The distribution of three-sigma rule .....	16
<b>Figure 3.5:</b>	User's colour input and the result of the binary threshold after standard deviation calculation .....	16
<b>Figure 3.6:</b>	Indications of the original captured frame and the converted Binary Threshold .....	18
<b>Figure 3.7:</b>	The main menu of the application .....	21
<b>Figure 3.8:</b>	The settings menu of the application .....	21
<b>Figure 3.9:</b>	The overall layout of the application .....	23
<b>Figure 3.10:</b>	The general representation of feedback window .....	24
<b>Figure 3.11:</b>	The general representation of Adaptive Threshold .....	24
<b>Figure 4.1:</b>	Agile method overview .....	27

## LIST OF TABLES

<b>Table 1.1:</b>	Advantage and disadvantage of the Mechanical Mouse .....	3
<b>Table 1.2:</b>	Advantage and disadvantage of the Optical and Laser Mouse ...	4
<b>Table 3.1:</b>	The overall colour combinations for specific mouse functions ..	19
<b>Table 3.2:</b>	The overall settings of the Virtual Mouse Colour Recognition program .....	21
<b>Table 3.3:</b>	The dynamic track-bars setting for HSV values modification ...	21
<b>Table 4.1:</b>	Verification Plan P1 .....	31
<b>Table 4.2:</b>	Verification Plan P2 .....	32
<b>Table 4.3:</b>	Verification Plan P3 .....	33
<b>Table 4.4:</b>	Verification Plan P4 .....	34
<b>Table 5.1:</b>	Different brightness testing results .....	35
<b>Table 5.2:</b>	Colour conflicts testing results .....	37
<b>Table 5.3:</b>	Tilt and Rotation testing results .....	37
<b>Table 5.4:</b>	Distance testing results .....	37



# LIST OF ABBREVIATIONS

<i>GUI</i>	Graphical User Interface
<i>LED</i>	Light Emitting Diodes
<i>HCI</i>	Human Computer Interaction
<i>MHI</i>	Motion History Images
<i>IDE</i>	Integrated Development Environment
<i>OpenCV</i>	Open Source Computer Vision
<i>HSV</i>	Hue, Saturation, and Values
<i>RGB</i>	Red, Green, and Blue

# 1. PROJECT BACKGROUND

## 1.1. Introduction

A mouse, in computing terms is a pointing device that detects two-dimensional movements relative to a surface. This movement is converted into the movement of a pointer on a display that allows to control the Graphical User Interface (GUI) on a computer platform. There are a lot of different types of mouse that have already existed in the modern days technology, there's the mechanical mouse that determines the movements by a hard rubber ball that rolls around as the mouse is moved. Years later, the optical mouse was introduced that replace the hard rubber ball to a LED sensor to detects table top movement and then sends off the information to the computer for processing. On the year 2004, the laser mouse was then introduced to improve the accuracy movement with the slightest hand movement, it overcome the limitations of the optical mouse which is the difficulties to track high-gloss surfaces. However, no matter how accurate can it be, there are still limitations exist within the mouse itself in both physical and technical terms. For example, a computer mouse is a consumable hardware device as it requires replacement in the long run, either the mouse buttons were degraded that causes inappropriate clicks, or the whole mouse was no longer detected by the computer itself.

Despite the limitations, the computer technology still continues to grow, so does the importance of the human computer interactions. Ever since the introduction of a mobile device that can be interact with touch screen technology, the world is starting to demand the same technology to be applied on every technological devices, this includes the desktop system. However, even though the touch screen technology for the desktop system is already exist, the price can be very steep.

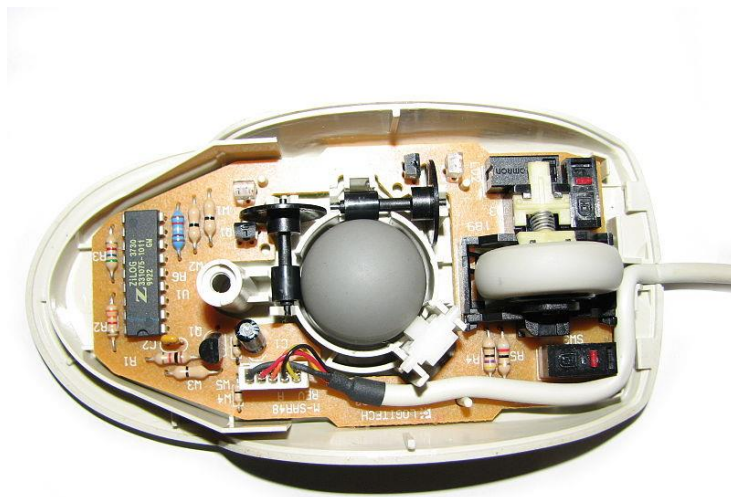
Therefore, a virtual human computer interaction device that replaces the physical mouse or keyboard by using a webcam or any other image capturing devices can be an alternative way for the touch screen. This device which is the webcam will be constantly utilized by a software that monitors the gestures given by the user in order to process it and translate to motion of a pointes, as similar to a physical mouse.

## 1.2. Review Of The Physical Mouse

It is known that there are various types of physical computer mouse in the modern technology, the following will discuss about the types and differences about the physical mouse.

### 1.2.1. Mechanical Mouse

Known as the trackball mouse that is commonly used in the 1990s, the ball within the mouse are supported by two rotating rollers in order to detect the movement made by the ball itself. One roller detects the forward/backward motion while the other detects the left/right motion. The ball within the mouse are steel made that was covered with a layer of hard rubber, so that the detection are more precise. The common functions included are the left/right buttons and a scroll-wheel. However, due to the constant friction made between the mouse ball and the rollers itself, the mouse are prone to degradation, as overtime usage may cause the rollers to degrade, thus causing it to unable to detect the motion properly, rendering it useless. Furthermore, the switches in the mouse buttons are no different as well, as long term usage may cause the mechanics within to be loosed and will no longer detect any mouse clicks till it was disassembled and repaired.



**Figure 1.1** Mechanical mouse, with top cover removed

The following table describes the advantages and disadvantages of the Mechanical Mouse.

Advantage	Disadvantage
<ul style="list-style-type: none"> <li>• Allows the users to control the computer system by moving the mouse.</li> <li>• Provides precise mouse tracking movements</li> </ul>	<ul style="list-style-type: none"> <li>• Prone to degradation of the mouse rollers and button switches, causing to be faulty.</li> <li>• Requires a flat surface to operate.</li> </ul>

**Table 1.1:** Advantage and disadvantage of the Mechanical Mouse

### 1.2.2. Optical And Laser Mouse

A mouse that commonly used in these days, the motions of optical mouse rely on the Light Emitting Diodes (LEDs) to detect movements relative to the underlying surface, while the laser mouse is an optical mouse that uses coherent laser lights. Comparing to its predecessor, which is the mechanical mouse, the optical mouse no longer rely on the rollers to determine its movement, instead it uses an imaging array of photodiodes. The purpose of implementing this is to eliminate the limitations of degradation that plagues the current predecessor, giving it more durability while offers better resolution and precision. However, there's still some downside, even-though the optical mouse are functional on most opaque diffuse surface, it's unable to detect motions on the polished surface. Furthermore, long term usage without a proper cleaning or maintenance may leads to dust particles trap between the LEDs, which will cause both optical and laser mouse having surface detection difficulties. Other than that, it's still prone to degradation of the button switches, which again will cause the mouse to function improperly unless it was disassembled and repaired.



**Figure 1.2** Optical Mouse, with top cover removed

The following table describes the advantages and disadvantages of the Optical and Laser Mouse.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Allows better precision with lesser hand movements.</li> <li>• Longer life-span.</li> </ul>	<ul style="list-style-type: none"> <li>• Prone to button switches degradation.</li> <li>• Does not function properly while on a polished surface.</li> </ul>

**Table 1.2:** Advantage and disadvantage of the Optical and Laser Mouse

### 1.3. Problem Statement

It's no surprised that every technological devices have its own limitations, especially when it comes to computer devices. After the review of various type of the physical mouse, the problems are identified and generalized. The following describes the general problem that the current physical mouse suffers:

- Physical mouse is subjected to mechanical wear and tear.
- Physical mouse requires special hardware and surface to operate.
- Physical mouse is not easily adaptable to different environments and its performance varies depending on the environment.
- Mouse has limited functions even in present operational environments.
- All wired mouse and wireless mouse have its own lifespan.

## **1.4. Motivation of Virtual Mouse**

It is fair to say that the Virtual Mouse will soon to be substituting the traditional physical mouse in the near future, as people are aiming towards the lifestyle where that every technological devices can be controlled and interacted remotely without using any peripheral devices such as the remote, keyboards, etc. it doesn't just provides convenience, but it's cost effective as well.

### **1.4.1. Convenient**

It is known in order to interact with the computer system, users are required to use an actual physical mouse, which also requires a certain area of surface to operate, not to mention that it suffers from cable length limitations. Virtual Mouse requires none of it, as it only a webcam to allow image capturing of user's hand position in order to determine the position of the pointers that the user want it to be. For example, the user will be able to remotely control and interact the computer system by just facing the webcam or any other image capturing devices and moving your fingers, thus eliminating the need to manually move the physical mouse, while able to interact with the computer system from few feet away.

### **1.4.2. Cost Effective**

A quality physical mouse is normally cost from the range of 30 ringgit to a hefty 400 ringgit, depending on their functionality and features. Since the Virtual Mouse requires only a webcam, a physical mouse are no longer required, thus eliminating the need to purchase one, as a single webcam is sufficient enough to allow users to interact with the computer system through it, while some other portable computer system such as the laptop, are already supplied with a built-in webcam, could simply utilize the Virtual Mouse software without having any concerns about purchasing any external peripheral devices.

### **1.5. Project Scope**

Virtual Mouse that will soon to be introduced to replace the physical computer mouse to promote convenience while still able to accurately interact and control the computer system. To do that, the software requires to be fast enough to capture and process every image, in order to successfully track the user's gesture. Therefore, this project will develop a software application with the aid of the latest software coding technique and the open-source computer vision library also known as the OpenCV. The scope of the project is as below:

- Real time application.
- User friendly application.
- Removes the requirement of having a physical mouse.

The process of the application can be started when the user's gesture was captured in real time by the webcam, which the captured image will be processed for segmentation to identify which pixels values equals to the values of the defined colour. After the segmentation is completed, the overall image will be converted to Binary Image where the identified pixels will show as white, while the rest are black. The position of the white segment in the image will be recorded and set as the position of the mouse pointer, thus resulting in simulating the mouse pointer without using a physical computer mouse. The software application is compatible with the Windows platform. The functionality of the software will be coded with C++ programming language code with the integration of an external library that does the image processing known as the OpenCV.

### **1.6. Project Objective**

The purpose of this project is to develop a Virtual Mouse application that targets a few aspects of significant development. For starters, this project aims to eliminate the needs of having a physical mouse while able to interact with the computer system through webcam by using various image processing techniques. Other than that, this project aims to develop a Virtual Mouse application that can be operational on all kind of surfaces and environment.

The following describes the overall objectives of this project:

- To design to operate with the help of a webcam.  
The Virtual Mouse application will be operational with the help of a webcam, as the webcam are responsible to capture the images in real time. The application would not work if there are no webcam detected.
- To design a virtual input that can operate on all surface.  
The Virtual Mouse application will be operational on all surface and indoor environment, as long the users are facing the webcam while doing the motion gesture.
- To program the camera to continuously capturing the images, which the images will be analysed, by using various image processing techniques.  
As stated above, the Virtual Mouse application will be continuously capturing the images in real time, where the images will be undergo a series of process, this includes HSV conversion, Binary Image conversion, salt and pepper noise filtering, and more.
- To convert hand gesture/motion into mouse input that will be set to a particular screen position.  
The Virtual Mouse application will be programmed to detect the position of the defined colours where it will be set as the position of the mouse pointers. Furthermore, a combination of different colours may result in triggering different types of mouse events, such as the right/left clicks, scroll up/down, and more.

### **1.7. Impact, Significance and Contribution**

The Virtual Mouse application is expected to replace the current methods of utilizing a physical computer mouse where the mouse inputs and positions are done manually. This application offers a more effortless way to interact with the computer system, where every task can be done by gestures. Furthermore, the Virtual Mouse application could assist the motor-impaired users where he/she could interact with the computer system by just showing the correct combination of colours to the webcam.

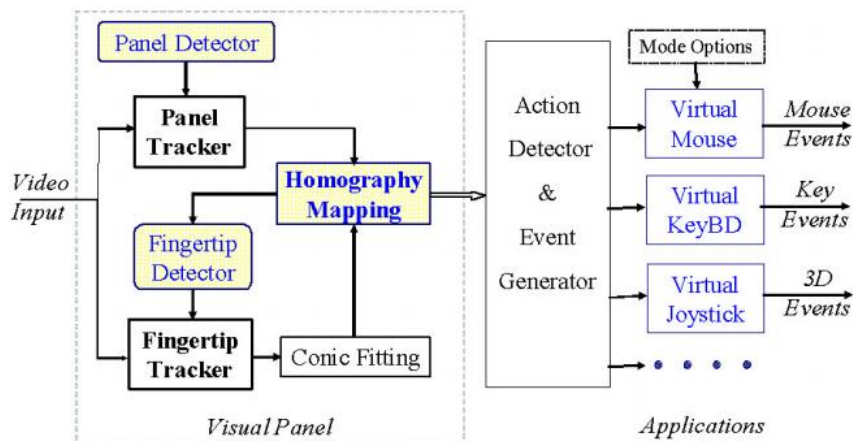


## 2. LITERATURE REVIEW

As modern technology of human computer interactions become important in our everyday lives, varieties of mouse with all kind of shapes and sizes were invented, from a casual office mouse to a hard-core gaming mouse. However, there are some limitations to these hardware as they are not as environmental friendly as it seems. For example, the physical mouse requires a flat surface to operate, not to mention that it requires a certain area to fully utilize the functions offered. Furthermore, some of these hardware are completely useless when it comes to interact with the computers remotely due to the cable lengths limitations, rendering it inaccessible.

### 2.1. Visual Panel

To overcome the stated problems, Zhengyou et al. (2001), proposed an interface system named Visual Panel that utilize arbitrary quadrangle-shaped planar object as a panel to allow the user to use any tip-pointer tools to interact with the computer. The interaction movements will be captured, analysed and implement the positions of the tip-pointer, resulting accurate and robust interaction with the computer. The overall system consists of panel tracker, tip-pointer tracker, holography, calculation and update, and action detector and event generator as it can simulate both mouse and keyboard.

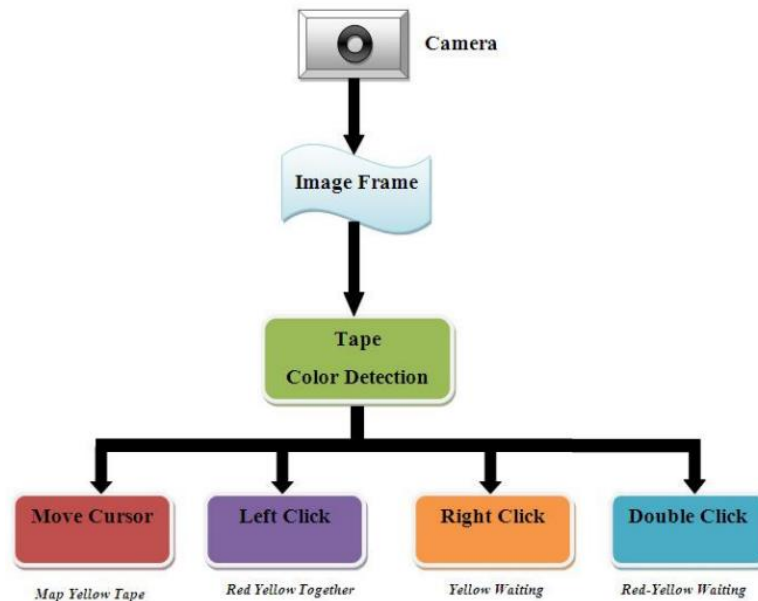


**Figure 2.1:** The system overview of Visual Panel (Zhengyou, Ying and Shafer, 2001)

However, although the proposed system solved the issues of cable length limitations, it still requires a certain area and material to operate. Zhengyou et al., have mentioned that the system can accept any panel as long as it is quadrangle-shaped, meaning any other shape besides stated shape are not allowed.

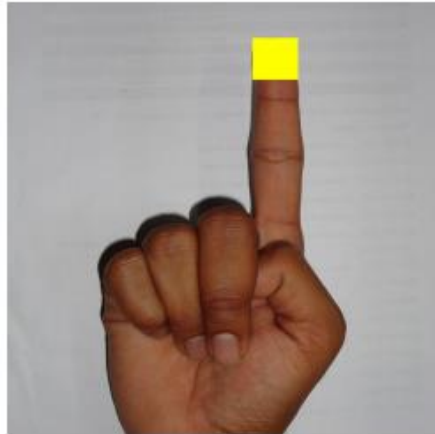
## 2.2. Mouse Simulation Using Two Coloured Tapes

Kamran Niyazi et al. (2012), mentioned that to solve the stated problem, ubiquitous computing method is required. Thus, colour tracking mouse simulation was proposed. The said system tracks two colour tapes on the user fingers by utilizing the computer vision technology. One of the tapes will be used for controlling the movement of the cursor while the other will act as an agent to trigger the click events of the mouse.



**Figure 2.2:** The system architecture of the mouse-simulation (Niyazi, 2012)

To detect the colours, the system are first required to process the captured image by separating the hand pixels from the non-hand pixels, which can be done by background subtraction scheme that segments the hands movement information from the non-changing background scene. In order to implement this, the system requires to capture a pair of images to represent the static workplace from the camera view. When subtraction process is complete, the system will undergo another process that separates the RGB pixels to calculate the probability and differentiate the RGB values to determine which part are the skin and which are not. After the process is completed, it will start detecting the defined colour in the image, the image RGB pixels will be converted into HSV colour plane in order to eliminate the variation in shades of similar colour. The resulting image will be converted to Binary Image and will undergo a filtering process to reduce the noise within the image.

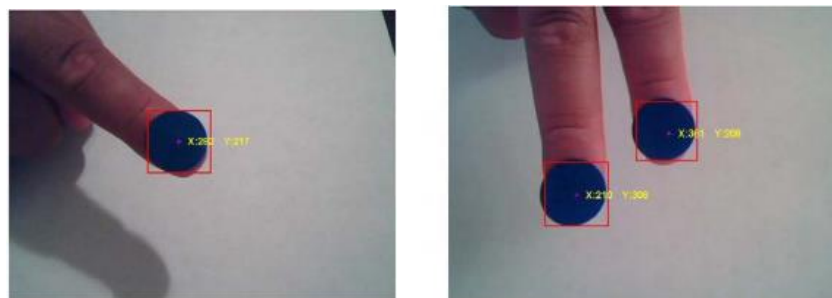


**Figure 2.3:** Yellow colour tape for cursor movement (Niyazi, 2012)

Even though the proposed system solved most of the stated issues, but there are limited functions offered by the proposed system as it merely able to perform common functions, such as: cursor movements, left/right click, and double clicks. While other functions, such as the middle click and mouse scroll were ignored.

### **2.3. Virtual Mouse Using a Webcam**

Another colour detection method proposed by Kazim Sekeroglu (2010), the system requires three fingers with three colour pointers to simulate the click events. The proposed system are capable of detecting the pointers by referring the defined colour information, track the motion of the pointers, move the cursor according to the position of the pointer, and simulate the single and double left or/and right click event of the mouse.

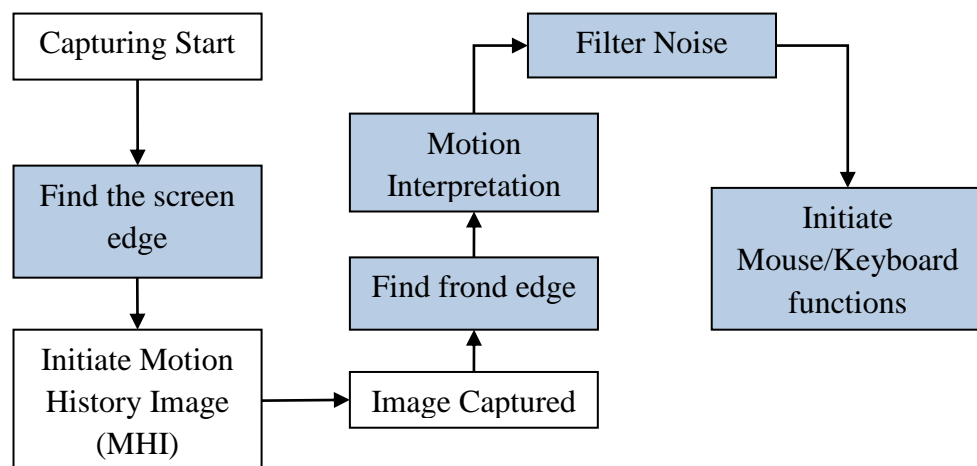


**Figure 2.4:** Input image using one and two pointers (Sekeroglu, 2010)

To detect the colours, they have utilized the MATLAB's built in "*imsubtract*" function, with the combination of the noise filtering by using median filter, which are effective in filtering out or at least reduce the "salt and pepper" noise. The captured image will be converted to Binary Scale Image by using MATLAB's built in "*im2bw*" function to differentiate the possible values for each pixel. When the conversion is done, the captured image will undergo another filtering process by using "*bwareaopen*" to remove the small areas in order to get an accurate number of the object detected in the image.

#### 2.4. Portable Vision-Based Human Computer Interaction(HCI)

Another "Ubiquitous Computing" approach proposed by Chu-Feng Lien (2015), requires only finger-tips to control the mouse cursor and click events. The proposed system doesn't requires hand-gestures nor colour tracking in order to interact with the system, instead it utilize a feature name Motion History Images(MHI) , a method that used to identify movements with a row of images in time.



**Figure 2.5:** The Flow Chart of Portable Vision-Based Human Computer Interaction (Chu-Feng, 2008)

Even though the proposed system possess good accuracy in a well-controlled environment, it does has its own limitations. The proposed system are not capable to detect fast moving movements as the frame-rates are not able to keep up, thus leading to increase of error rate. Furthermore, due to the mouse click events occurred when the finger hold on a certain positions, this may lead to user constant finger movements to prevent false alarm, which may result inconvenience.

## **2.5. Conclusion**

There are abundance of methods for computer interaction besides the traditional physical mouse interaction. With the evolutions of modern technology and programming, so does the Human Computer Interaction (HCI) methods, as it allows unlimited ways to access the computers. This approach allows the developers to design specific/unique system that suit the needs of the users, from gesture movement tracking to coloured tracking, it's no surprise that in near future, physical mouse will no longer be needed and be replaced by video cameras that tracks gestures.

### 3. SYSTEM DESIGN

#### 3.1. System Design

During the process of colour recognition, it contains 2 major phases which are the calibration phase and recognition phase. The purpose of the calibration phase is to allow the system to recognize the Hue Saturation Values of the colours chosen by the users, where it will store the values and settings into text documents, which will be used later on during the recognition phase. While on the recognition phase, the system will start to capture frames and search for colour input with based on the values that are recorded during the calibration phase. The phases of the virtual mouse is as shown in figure below.

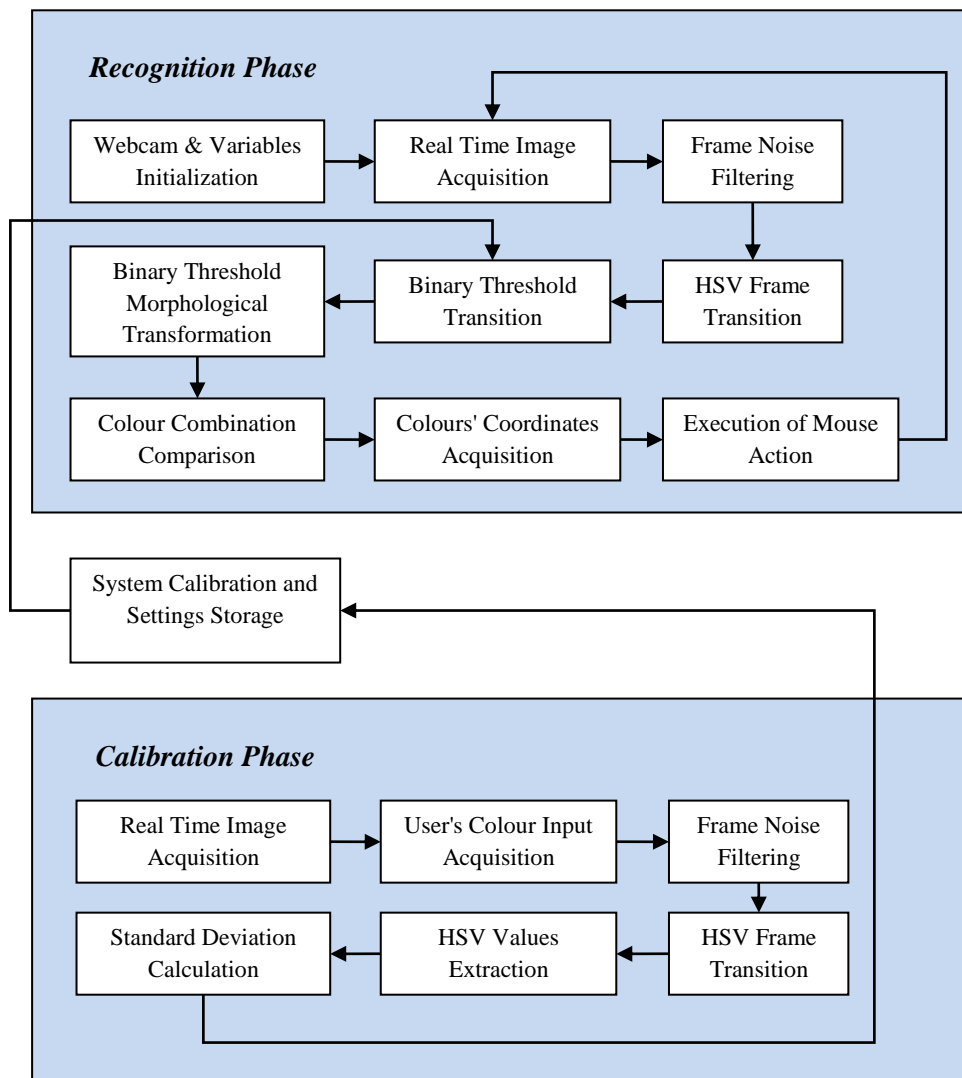


Figure 3.1: Virtual Mouse Block Diagram

### 3.1.1. Calibration Phase

#### a) Real Time Image Acquisition

The program will start of by capturing real-time images via a webcam where it will await for users' colour input. The size of the acquired image will be compressed to a reasonable size to reduce the processing loads of processing the pixels within the captured frame.

#### b) User's Colour Input Acquisition

The program acquires the frames that consist of input colours submitted by the users, the captured frame will be sent for process where it will undergone a series of transition and calculation to acquire the calibrated HSV values.

#### c) Frame Noise Filtering

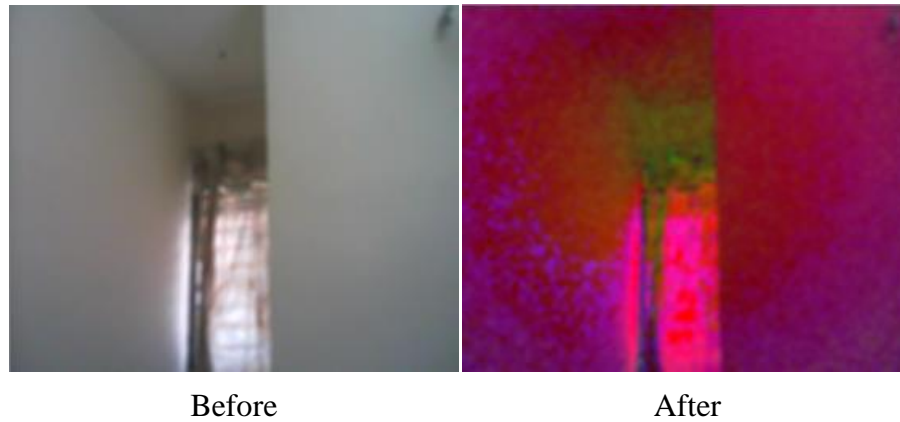
Every captured frame consists of noises that will affect the performance and the accuracy of the program, therefore the frame require to be noise free. To do that, filters need to be applied on the captured frames to cancel out the unwanted noise. For the current project, Gaussian filter will be used, which is a common smoothing method to eliminate noise in a frame. This can be done by using **GaussianBlur**(InputArray **src**, OutputArray **dst**, Size **ksize**, double **sigmaX**, double **sigmaY=0**, int**borderType** = **BORDER\_DEFAULT** ).



**Figure 3.2:** The comparison between un-filtered and filtered frame

d) HSV Frame Transition

The captured frame require to be converted from a BGR format to a HSV format. Which can be done by using `cvtColor(src, dst, CV_BGR2HSV)`.



**Figure 3.3:** The comparison between RGB and HSV frame

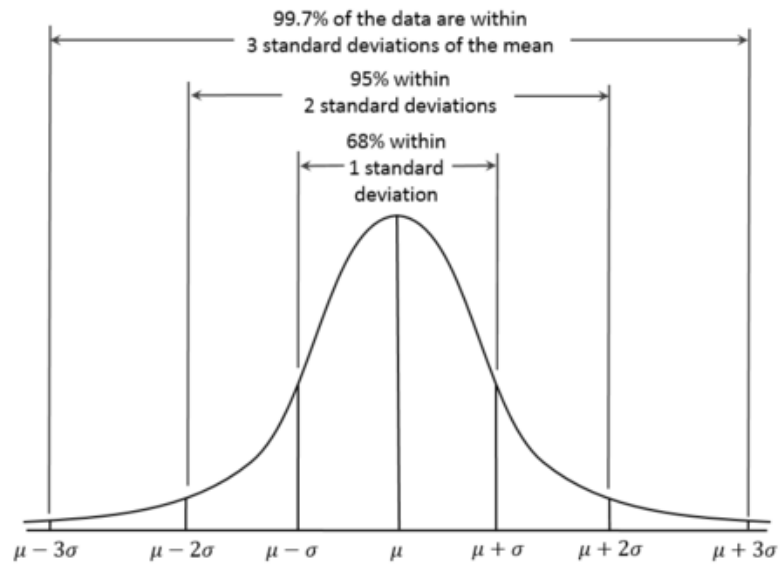
e) HSV Values Extraction

In order to acquire the HSV values, the converted frame require to be split into 3 single different planes, to do that the frame needs to be divided from a multi-channel array into a single channel array, which can be done by using `split(const Mat& src, Mat* mvbegin)`.

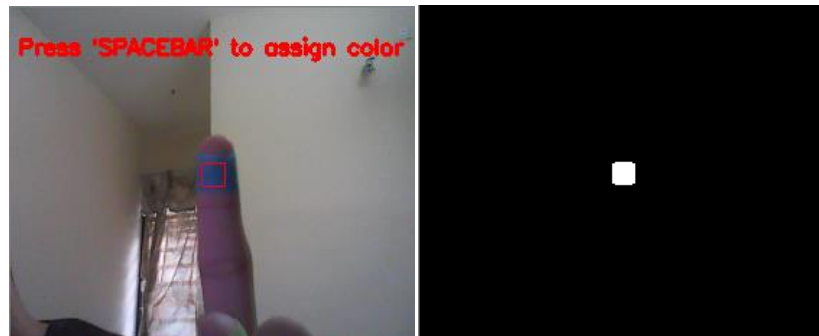
f) Standard Deviation Calculation

To obtain the maximum and the minimum of the HSV values, it requires to gone through the Standard Deviation calculation, a measurement used to quantify the amount of variation / dispersion among other HSV values. Furthermore, to obtain an accurate range of values, three-sigma rule are required in the calculation, so that chances of the captured values have a very high possibility to fall within the three-sigma intervals.





**Figure 3.4:** The distribution of three-sigma rule



**Figure 3.5:** User's colour input and the result of the binary threshold after standard deviation calculation

### 3.1.2. Recognition Phase

#### a) Webcam & Variables Initialization

On the early stage of the recognition phase, the program will initialize the required variables which will be used to hold different types of frames and values where each are will be used to carry out certain task. Furthermore, this is the part where the program collects the calibrated HSV values and settings where it will be used later during the transitions of Binary Threshold.

#### b) Real Time Image Acquisition

The real time image is captured by using the webcam by using `(cv::VideoCapture cap(0);)`, where every image captured are stored into a frame variable (`cv::Mat`), which will be flipped and compressed to a reasonable size to reduce process load.

#### c) Frame Noise Filtering

Similar to the noise filtering during the calibration phase, Gaussian filters will be applied to reduce the existing noise of the captured frames. This can be done by using **GaussianBlur**(InputArray **src**, OutputArray **dst**, Size **ksize**, double **sigmaX**, double **sigmaY=0**, **intborderType=BORDER\_DEFAULT** ).

#### d) HSV Frame Transition

The captured frame require to be converted from a BGR format to a HSV format. Which can be done by using `cvtColor(src, dst, CV_BGR2HSV)`.

#### e) Binary Threshold Transition

The converted HSV frame will undergone a range check to check if the HSV values of the converted frame lies between the values of the HSV variables gathered during the calibration phase. The result of the range check will convert the frame into a Binary Threshold, where a part of the frame will set to 255 (1 bit) if the said frame lies within the specified HSV values, the frame will set to 0 (0 bit) if otherwise.

f) Binary Threshold Morphological Transformation

After the binary threshold is obtained, the frame will undergo a process called Morphological Transformation, which is a structuring operation to eliminate any holes and small object lurking around the foreground. The transformation consist of two morphological operators, known as Erosion and Dilation. The Erosion operator are responsible for eroding the boundaries of the foreground object, decreasing the region of the binary threshold, which is useful for removing small noises. As for Dilation, it is an opposite of erosion, it increases the region of the binary threshold, allowing eroded object to return to its original form. For the current project, both operators were used for morphological Opening and Closing, where Opening consists of combination of erosion followed by dilation, which is very useful in removing noise, whereas Closing is the opposite of Opening, which is useful in closing small holes inside the foreground object.



**Figure 3.6:** Indications of the original captured frame and the converted Binary Threshold

g) Colour Combination Comparison

After obtaining results from Morphological Transformation process, the program will calculate the remaining number of objects by highlighting it as blobs, this process requires cvBlob library, which is an add-on to OpenCV. The results of the calculation will then send for comparison to determine the mouse functions based on the colour combinations found within the captured frames.

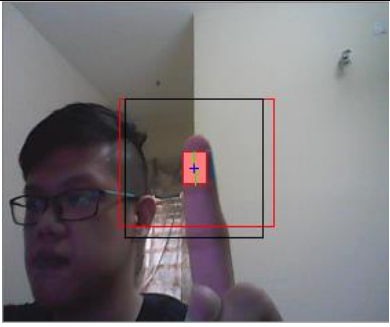
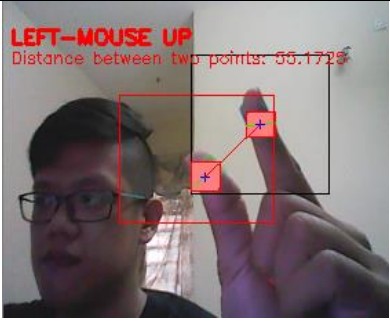
#### h) Colours' Coordinates Acquisition

For every object within the binary threshold, the program will highlight the overall shape of the object (`cvRenderBlobs(const IplImage *imgLabel, CvBlobs &blobs, IplImage *imgSource, IplImage *imgDest, unsigned short mode=0x000f, double alpha=1.);`), where it will calculate the area of the shape and the coordinates of midpoint of the shapes. The coordinates will be saved and used later in either setting cursor positions, or to calculate the distance between two points to execute various mouse functions based on the result collected.

#### i) Execution of Mouse Action

The program will executes the mouse actions based on the colours combinations exist in the processed frame. The mouse actions will perform according to the coordinates provided by the program, and the program will continue on acquire and process the next real-time image until the users exit from the program.

The following describes the colour combinations to trigger and execute the mouse functions:

Sample Layout	Mouse Functions	Colour Combinations
	Move Cursor	First Colour (e.g. Blue)
	Left Click Up	First Colour + Second Colour (Un-pinch Gesture) (e.g. Blue + Green)

	Left Click Down	First Colour + Second Colour (Pinch Gesture) (e.g. Blue + Green)
	Right Click	First Colour + Third Colour (e.g. Blue + Red)
	Enable/Disable Scroll Mode	First Colour + Second Colour + Third Colour (e.g. Blue + Green + Red)
	Scroll Down	(During Scroll Mode) First Colour + Second Colour (Un-pinch Gesture) (e.g. Blue + Green)
	Scroll Up	(During Scroll Mode) First Colour + Second Colour (Pinch Gesture) (e.g. Blue + Green)

**Table 3.1:** The overall colour combinations for specific mouse functions

### 3.2. Application Layout

The application will start of by displaying a console window; users are required to select the options available in the main menu as different options leads to different functions of the program. If the users somehow select an incorrect options (e.g. fourth option), the result will be the display of error message and re-direct users back to the main menu.

```
-----
Welcome to Wan Chung Son Virtual Mouse Application
-----
| 1 | START Virtual Mouse
| 2 | Color Calibrations
| 3 | Settings
-----
choice:
```

Figure 3.7: The main menu of the application

The second option allows the user to choose and calibrate the desired colours in order to achieve optimal accuracy and performance during the recognition phase. Other than that, the third option allows the user to modify the settings of the program, this includes the height and width of the feedback windows, choices of webcams, and more.

```
-----
Virtual Mouse Application Settings
-----
| 1 | Height of windows: 230
| 2 | Width of windows: 280
| 3 | Webcam choice: 0
| 4 | Height of captured margin: 100
| 5 | Width of captured margin: 100
| 6 | Adaptive Threshold: 0
| 7 | Reset to Default
| 8 | Return to main menu
-----
choice: _
```

Figure 3.8: The settings menu of the application

The table below describes each and every setting within the settings menu:

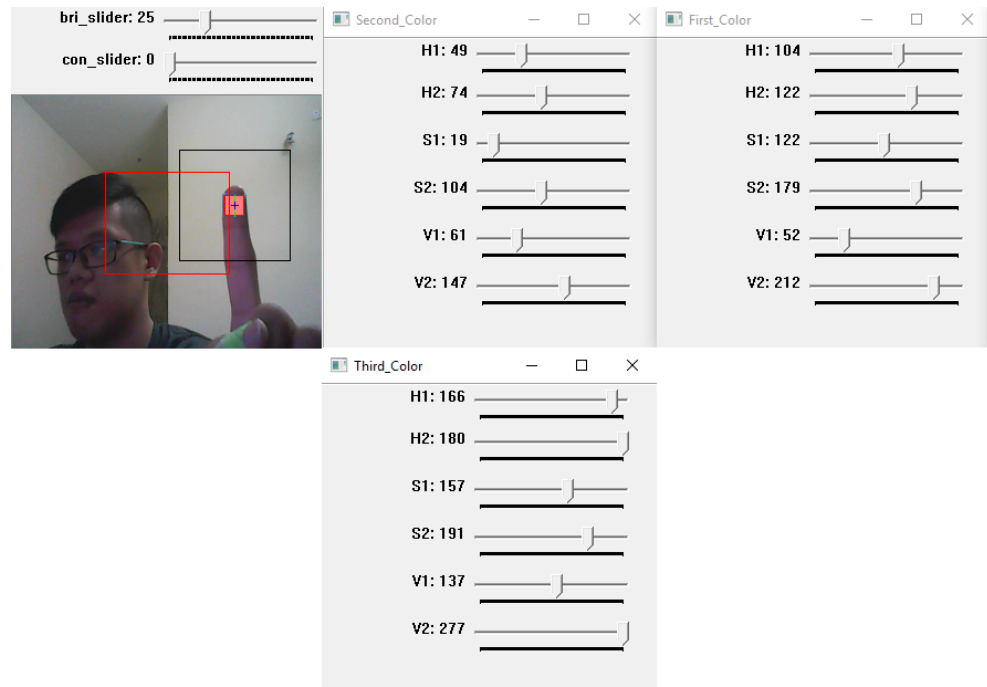
No.	Settings Name	Purpose
1.	Height of Windows	Alters the height of the feedback windows.
2.	Width of Windows	Alters the width of the feedback windows.
3.	Webcam choice	Alters the choices of webcams.
4.	Height of captured margin	Alters the height of the captured margin.
5.	Width of captured margin	Alters the width of the captured margin.
6.	Adaptive Threshold	Enable or disable the adaptive threshold function.
7.	Reset to Default	Reset the settings to its original state.
8.	Return to main menu	Save current settings and return to the main menu.

**Table 3.2:** The overall settings of the Virtual Mouse Colour Recognition program

Whenever the first option have been chosen, the application will start to start up several processes and initialized the required variables, which will then start to display several windows that indicate the binary threshold and the HSV track-bars of individual colours, and a main window that displays the real-time captured frame. The provided HSV track-bars are adjustable, as users are allowed to apply slight adjustment to improve the accuracy of colour recognition. However, to properly adjust the track-bars, users are required to have a basic understanding of HSV colour model, where un-experience adjustment may lead the whole recognition phase to be haywire. The following describes the purpose of each track-bar:

No.	Track-bar name	Purpose
1.	H1	The upper boundaries of Hue plane.
2.	H2	The lower boundaries of Hue plane.
3.	S1	The upper boundaries of Saturation plane.
4.	S2	The lower boundaries of Saturation plane.
5.	V1	The upper boundaries of Value plane.
6.	V2	The lower boundaries of Value plane.

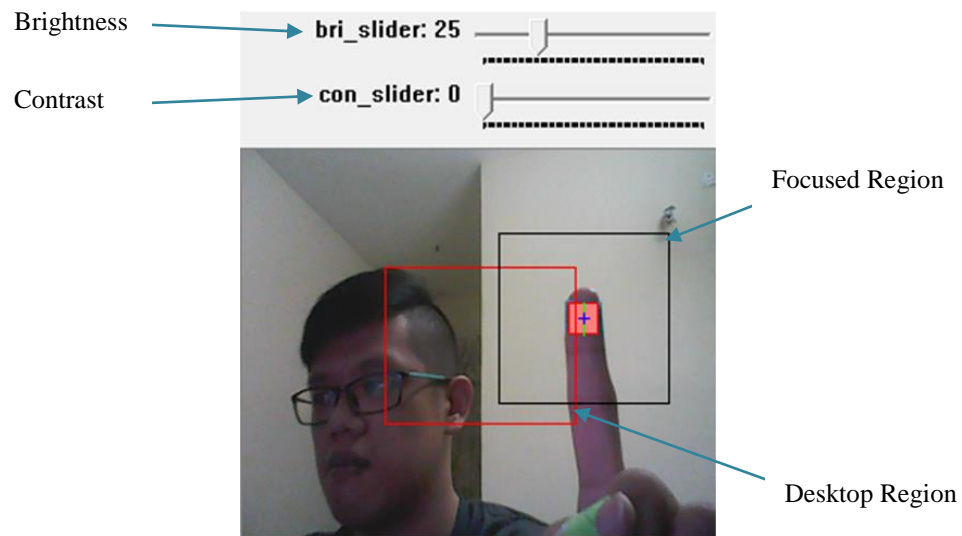
**Table 3.3:** The dynamic track-bars setting for HSV values modification



**Figure 3.9:** The overall layout of the application

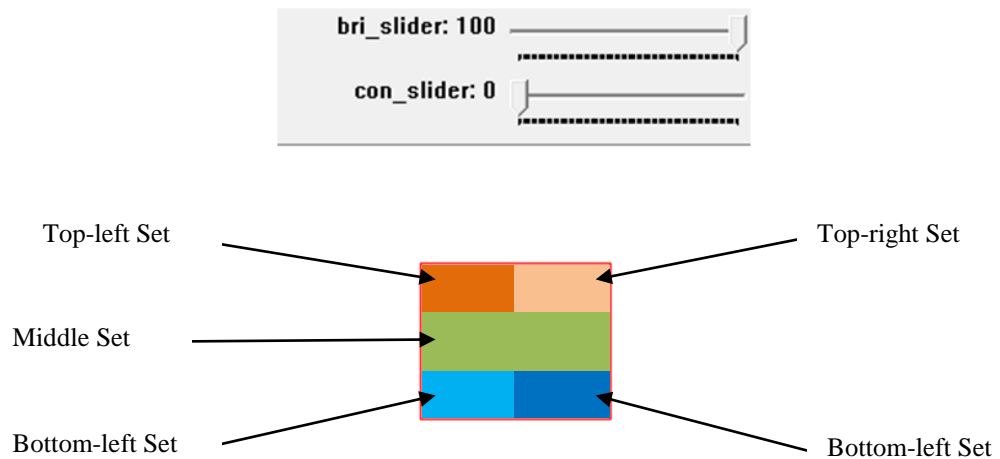
The feedback window named "Live" that displays the real-time frame consists of two different region that are outlined with red and black individually. The region that were outlined in red represents the size of the physical monitor that the users currently using, as the coordinates set within that region reflects an actual representation to the physical monitor. Other than that, the region that were outlined in black represents the focused region, where the other two colours (e.g. Green and Red) can only be detected within that region. The implementation of focused region is to eliminate colour conflicts occurs outside of the focused region, not to mention that the entire mouse functions can be executed with merely three fingers, thus eliminating the need of process the entire frame for all three different colours. Furthermore, the feedback window also includes two track-bars that are named "bri\_slider" and "con\_slider" where these two track-bars are responsible of adjusting the brightness and the contrast of the captured frame, the HSV values of the captured frame will be altered accordingly as well. The following describes the general representation of each element within the feedback window.





**Figure 3.10:** The general representation of feedback window

Besides that, there are an additional option where the users are able to enable adaptive threshold which can be found in the settings menu. The purpose of the adaptive threshold is to allow the program to constantly alters the variables set that holds the HSV values, where the variables set alteration are based on the position of the targeted colours currently at, which as shown as below:



**Figure 3.11:** The general representation of Adaptive Threshold

### 3.3. Implementation Issues and Challenges

Throughout the development of the application, there are several implementation issues occurred. The following describes the issues and challenges that will likely to be encountered throughout the development phase:

- **The interruptions of salt and pepper noises within the captured frames.**

Salt and pepper noises occurred when the captured frame contains required HSV values that are too small, but still underwent a series of process even though it's not large enough to be considered an input. To overcome this issue, the unwanted HSV pixels within the frame must first be filtered off, this includes the area of the pixels that are too large and small. With this method, the likelihood of interruptions of similar pixels will reduce greatly.

- **Performance degradation due to high process load for low-tier system.**

Since the application is required to undergo several of process to filter, process and execute the mouse functions in real time, the application can be CPU intensive for most of the low-tier system. If the size of the captured frames is too large, the time-taken for the application to process the entire frame are increase drastically. Therefore, to overcome this issue, the application is required to process only the essential part of the frames, and reduces the redundant filtering process that could potentially slow the application down.

- **The difficulties of calibrating the brightness and the contrast of the frames to get the required HSV values.**

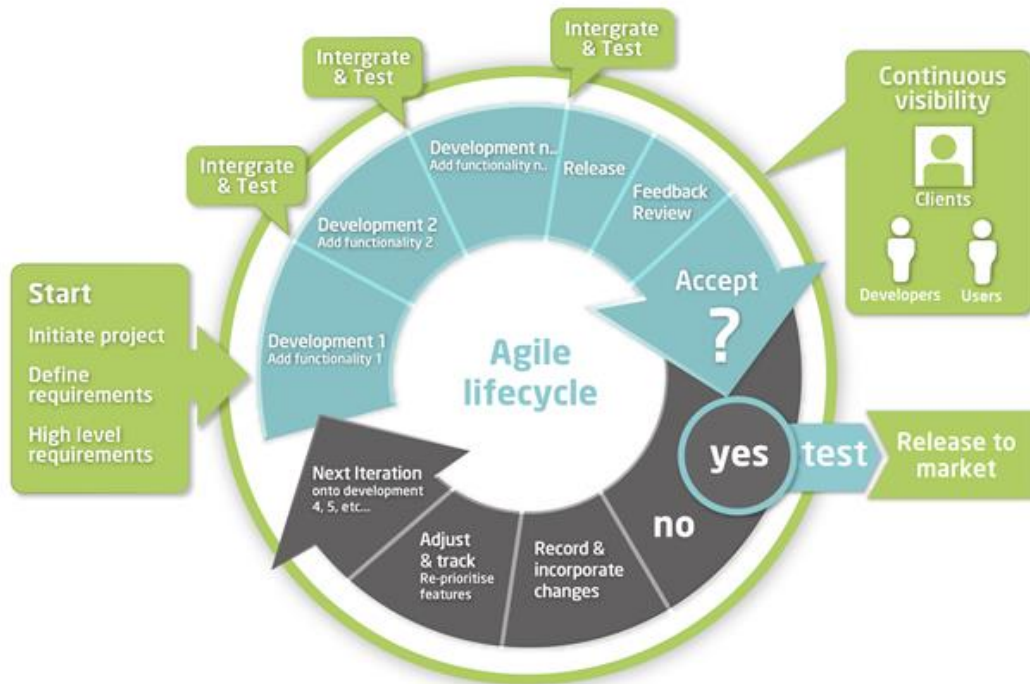
The intensity of brightness and contrast matters greatly when it comes to acquiring the required colour pixels. In order for the application to execute the entire mouse functions provided, all of the required HSV values to execute the specific mouse functions must be satisfied, meaning that the overall HSV values must be satisfied with the brightness and contrast as well. However, the calibration can be somewhat tedious as certain intensity could only satisfy part of the required HSV values, unless the original HSV values were modified to prove otherwise. To overcome this issue, the application must first start up with a calibration phase, which allows the users to choose their desired colour pixels before directing them to the main phase.

## **4. METHODS AND TECHNOLOGIES INVOLVED**

### **4.1. Methodology**

For this project we'll be using the Agile Software Development methodology approach in developing the application. The stated approach is an alternative to the traditional waterfall model that helps the project team respond to unpredictability through incremental and iterative work. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible respond to change. The following describes the principles of the Agile Software Development methodology:

- Satisfy the customer by early and continuous delivery of workable software.
- Encourage changes of requirement.
- Workable software is delivered frequently.
- Continuous collaboration between the stakeholders and the developers.
- Project are developed around motivated individuals.
- Encourage informal meetings.
- Operational software is the principle measure of progress.
- Sustainable development, able to maintain a constant pace.
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular adaption to changing circumstances



**Figure 4.1:** Agile method overview (Kumar N, 2011)

The reason for choosing this methodology is due to the fact that the Virtual Mouse are still considered to be at the introduction stage, which means it still requires a great deal of extensive research and development before it could actually make it into the market. Therefore, this project requires a thorough yet iterative planning and requirements gathering where the lifecycle will be continually revisited to re-evaluates the direction of the project and to eliminate the ambiguities in the process of the development, and at the same time welcome changes of requirements, which promotes adaptability and flexibility. Furthermore, due to the Virtual Mouse application are more towards serving the users, this project requires continuous customer collaboration, as they're essential for gathering the proper requirements in all aspects. This is why that the agile methodology is the ideal approach for developing the project.

The following describes the phases within the agile methodology approach:

- Planning

A thorough planning will be conducted in this phase where the existing systems/product, for this case, physical computer mouse will be reviewed and studied to identify the problems existed, a comparison of problems will be made to compare which problems are more crucial and requires improvement. An outline objective and the scope will be identified in order to provide an alternative solution to the problem.

- Requirement Analysis

The phase that gathers and interpreting the facts, diagnosing problems and recommending improvements to the system. In this phase, the collected problem statements will be extensively studied in order to find a proper solution or at least an improvements to the proposed system. All proposed solutions will be converted into requirements where it will be documented in a requirement specification.

- Designing

The requirement specification from the previous phase will be studied and prioritize to determine which requirement are more important where the requirement with the highest priority will be delivered first. After the study, the system design will be prepared as it helps in defining the overall system architecture and specifying the hardware and the software requirements.

- Building

The phase where the actual coding implementation takes place. By referring to the inputs from the system design, the system will be developed based on the prioritize requirements. However, due to we're using the agile methodology approach, the developed system will be considered as a prototype system where it will be integrated and tested by the users.

- Testing

The phase where the prototype system going through a series of test. The prototype system will first undergo integration where the features from the previous iteration cycle are added to the latest cycle. After the integration, the prototype system will be thoroughly tested by the users to determine whether they are satisfied with the latest deliverables, the completion of the project depends on whether they've accepted it or otherwise. If the users requires additional features or modification, feedback gathering will be conducted, which resulted in further modification of the requirements and features where it will recorded and documented for the requirement analysis phase on the next iteration.

## 4.2. Hardware and Software Requirement

### 4.2.1. Hardware Requirement

The following describes the hardware needed in order to execute and develop the Virtual Mouse application:

- Computer Desktop or Laptop

The computer desktop or a laptop will be utilized to run the visual software in order to display what webcam had captured. A notebook which is a small, lightweight and inexpensive laptop computer is proposed to increase mobility.

System will be using

Processor	:	Core2Duo
Main Memory	:	4GB RAM
Hard Disk	:	320GB
Display	:	14" Monitor

- Webcam

Webcam is utilized for image processing, the webcam will continuously taking image in order for the program to process the image and find pixel position.

#### 4.2.2. Software Requirement

The following describes the software needed in-order to develop the Virtual Mouse application:

- C++ Language

The coding technique on developing the Virtual Mouse application will be the C++ with the aid of the integrated development environment (IDE) that are used for developing computer programs, known as the Microsoft Visual Studio. A C++ library provides more than 35 operators, covering basic arithmetic, bit manipulation, indirection, comparisons, logical operations and others.

- Open CV Library

OpenCV are also included in the making of this program. **OpenCV (Open Source Computer Vision)** is a library of programming functions for real time computer vision. OpenCV have the utility that can read image pixels value, it also have the ability to create real time eye tracking and blink detection.

Software will be using:

OS	:	Window 7 Ultimate 64-bit
Language	:	C++
Tool Used	:	Open CV and CMake

### 4.3. Verification Plan

The Virtual Mouse Colour Recognition requires being able to recognize most of the colours provided by the users with high accuracy, consistency, and minimal performance impact on other processes. However, the recognition results may vary whenever the qualities of the captured frames have changed, as it may be affected by different situations in terms of environment, brightness, and the weather. The following describes the situations which may result in false detection and/or any other problem that may occur during recognition phase:

- a) The real-time images are taken under dark or bright environment condition.
- b) The real-time images are taken in a colour conflicts background.
- c) The users interact with the program in near or far distance.
- d) The real-time images are rotated in a clockwise or anti-clockwise rotation.

In order to achieve greater accuracy and consistency throughout the whole recognition cycle, verification plan is required to be implemented in order for the program to perform flawlessly. The verification plans is as follows:

#### a) Dark or bright environment condition

Procedure Number	P1
Method	Testing
Applicable Requirements	Recognize colours input in a dark environment
Purpose / Scope	To recognize colours input in a dark environment
Item Under Test	Real-time image captured from webcam
Precautions	The targeted colours must be calibrated first
Special Conditions/Limitations	If the colours are not calibrated, the program will not be able to detect the targeted colours accurately.
Equipment/Facilities	Laptop
Data Recording	None
Acceptance Criteria	The colours are detected successfully
Procedure	<ul style="list-style-type: none"> <li>• Run the program in the dark room.</li> <li>• Calibrate the desired colours.</li> <li>• Performs colour recognition by attempting to execute mouse functions by interacting with the</li> </ul>



	program.
Troubleshooting	Modify the HSV track-bars
Post-Test Activities	None

**Table 4.1:** Verification Plan P1

b) Colour conflicts condition

Procedure Number	P2
Method	Testing
Applicable Requirements	Recognize colours input in a colour conflicts background.
Purpose / Scope	To recognize colours input in a colour conflicts background.
Item Under Test	Real-time image captured from webcam
Precautions	The targeted colours must be calibrated first
Special Conditions/Limitations	If the colours are not calibrated, the program will not be able to detect the targeted colours accurately.
Equipment/Facilities	Laptop
Data Recording	None
Acceptance Criteria	Program ignores inputs to avoid unnecessary result.
Procedure	<ul style="list-style-type: none"> <li>• Run the program with more than one colour that has similar RGB values (e.g. blue and sky-blue).</li> <li>• Calibrate the desired colours.</li> <li>• Performs colour recognition by attempting to execute mouse functions by interacting with the program.</li> </ul>
Troubleshooting	Modify the HSV track-bars
Post-Test Activities	None

**Table 4.2:** Verification Plan P2

c) Various distance between webcam

Procedure Number	P3
Method	Testing
Applicable Requirements	Recognize colours input in near or far distance.
Purpose / Scope	To recognize colours input in near or far distance.
Item Under Test	Real-time image captured from webcam
Precautions	The targeted colours must be calibrated first
Special Conditions/Limitations	If the colours are not calibrated, the program will not be able to detect the targeted colours accurately.
Equipment/Facilities	Laptop
Data Recording	None
Acceptance Criteria	The colours are detected successfully regardless of distance.
Procedure	<ul style="list-style-type: none"> <li>• Run the program.</li> <li>• Calibrate the desired colours.</li> <li>• Performs colour recognition by attempting to execute mouse functions by interacting with the program in near or far distance.</li> </ul>
Troubleshooting	Modify the HSV track-bars
Post-Test Activities	None

**Table 4.3:** Verification Plan P3

d) Rotation

Procedure Number	P4
Method	Testing
Applicable Requirements	Recognize colours input in tilted or rotated angle.
Purpose / Scope	To recognize colours input in tilted or rotated angle.
Item Under Test	Real-time image captured from webcam
Precautions	The targeted colours must be calibrated first
Special Conditions/Limitations	If the colours are not calibrated, the program will not be able to detect the targeted colours accurately.
Equipment/Facilities	Laptop
Data Recording	None
Acceptance Criteria	The colours are detected successfully
Procedure	<ul style="list-style-type: none"> <li>• Flip or tilt the webcam</li> <li>• Run the program.</li> <li>• Calibrate the desired colours.</li> <li>• Performs colour recognition by attempting to execute mouse functions by interacting with the.</li> </ul>
Troubleshooting	Modify the HSV track-bars
Post-Test Activities	None

**Table 4.4:** Verification Plan P4

## 5. IMPLEMENTATION AND TESTING

### 5.1. Overview


In order to achieve accuracy, and consistency of the Virtual Mouse colour recognition, testing phase have been conducted on various scenarios.

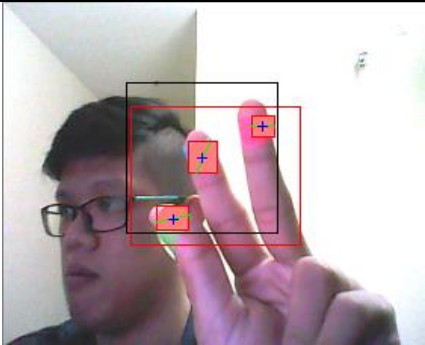

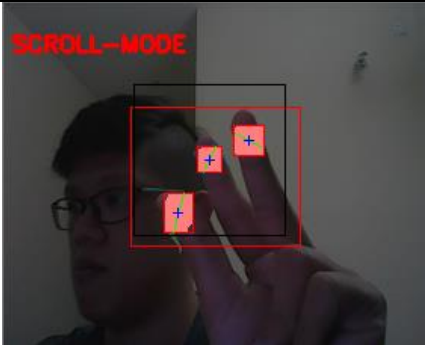
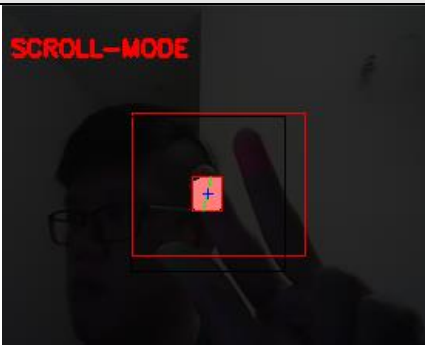
- Performance in various environments
  - Brightness
  - Colour Conflicts
  - Tilt and Rotation
  - Distance

The purpose of testing phase is to ensure that the final deliverable is able to perform flawlessly in terms of accuracy, consistency, and performance. To achieve that, the program has to able to recognize the colours input provided by the users with minimal adjustment, provide that the colours are thoroughly calibrated at first hand. Furthermore, the program is require to be able to execute the mouse functions efficiently and accurately as well.

### 5.2. Performance in Various Environments

The following describes the outcome of the program testing in various environments:

<b>Brightness</b>	
Normal Environment	
	<p>Results:</p> <p>All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly.</p>

Bright Environment	
	<p>Results:</p> <p>All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly. However, minimal adjustment of HSV track-bars were required.</p>
Brighter Environment	
	<p>Results:</p> <p>All colours cannot be recognized.</p> <p>Reason:</p> <p>The intensity of brightness were too high that greatly alters the original RGB values of the targeted colours, causing it to be unrecognizable.</p>
Dark Environment	
	<p>Results:</p> <p>All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly. However, minimal adjustment of HSV track-bars were required.</p>
Darker Environment	
	<p>Results:</p> <p>Some colours cannot be recognized.</p> <p>Reason:</p> <p>The intensity of brightness were too low that greatly alters the original RGB values of the targeted colours, causing it to be unrecognizable.</p>

**Table 5.1:** Different brightness testing results

### Colour Conflicts

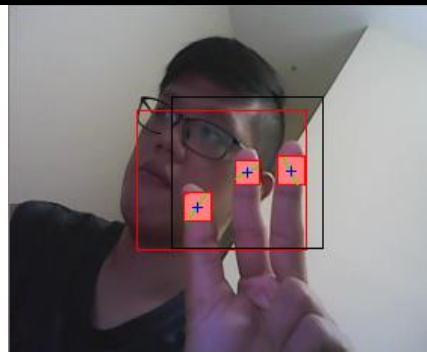


#### Results:

Colours input are successfully ignored, the conflicts were detected, causing it to stop executing unwanted mouse functions until the colour conflicts are no longer within the frame.

Table 5.2: Colour conflicts testing results

### Tilt and Rotation



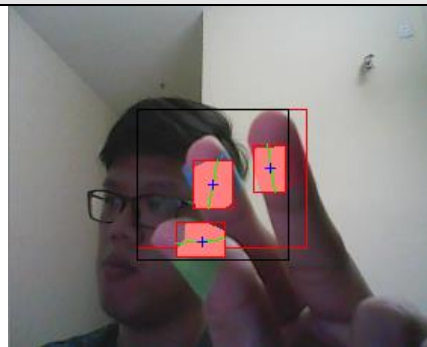
#### Results:

All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly.

Table 5.3: Tilt and Rotation testing results



### Distance

Near (15cm away from webcam)



#### Results:

All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly.

Far (25cm away from webcam)	
	<p>Results:</p> <p>All colours are successfully recognized. The three highlighted squares indicate that the targeted colours are identified, compared, and execute accordingly.</p>
Farther (35cm away from webcam)	
	<p>Results:</p> <p>All colours cannot be recognized.</p> <p>Reason:</p> <p>The area of targeted colours were too small and was filtered off.</p>

**Table 5.4:** Distance testing results

## 6. CONCLUSION

### 6.1. Overview

In conclusion, it's no surprised that the physical mouse will be replaced by a virtual non-physical mouse in the Human-Computer Interactions (HCI), where every mouse movements can be executed with a swift of your fingers everywhere and anytime without any environmental restrictions. This project had develop a colour recognition program with the purpose of replacing the generic physical mouse without sacrificing the accuracy and efficiency, it is able to recognize colour movements, combinations, and translate them into actual mouse functions. Due to accuracy and efficiency plays an important role in making the program as useful as an actual physical mouse, a few techniques had to be implemented.

First and foremost, the coordinates of the colours that are in charge of handling the cursor movements are averaged based on a collections of coordinates, the purpose of this technique is to reduce and stabilize the sensitivity of cursor movements, as slight movement might lead to unwanted cursor movements. Other than that, several colour combinations were implemented with the addition of distance calculations between two colours within the combination, as different distance triggers different mouse functions. The purpose of this implementation is to promote convenience in controlling the program without much of a hassle. Therefore, actual mouse functions can be triggered accurately with minimum trial and errors.

Furthermore, to promote efficient and flexible tracking of colours, calibrations phase was implemented, this allows the users to choose their choices of colours on different mouse functions, as long the selected colours doesn't fall within the same/similar RGB values (e.g. blue and sky-blue). Other than that, adaptive calibrations were also implemented as well, it is basically allows the program to save different set of HSV values from different angles where it will be used during the recognition phase.

In Overall, the modern technologies have come a long way in making the society life better in terms of productivity and lifestyle, not the other way around. Therefore, societies must not mingle on the past technologies while reluctant on



accepting changes of the newer one. Instead, it's advisable that they should embrace changes to have a more efficient, and productive lifestyle.

## **6.2. Limitation**

In this project, there are several existing problems that may hinder the results of colour recognitions. One of the problems is the environmental factor during the recognition phase takes place. The recognition process are highly sensitive on the intensity of brightness, as immense brightness or darkness may cause the targeted colours to be undetected within the captured frames. Besides that, distance is also the one of the problem that may affect the colour recognition results, as the current detection region can support up to 25cm radius, any display of colours exceed the mentioned distance will be considered as a noise and be filtered off.

Furthermore, the performance of the program are highly dependent on the users' hardware, as processor speed and/or resolutions taken from the webcam could have an effect on performance load. Therefore, the slower the processing speed and/or the higher the resolutions, the longer time are required to process a single frame.

## **6.3. Future Works**

There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments. The following describes the improvements and the features required:

### a) Smart Recognition Algorithm

Due to the current recognition process are limited within 25cm radius, an adaptive zoom-in/out functions are required to improve the covered distance, where it can automatically adjust the focus rate based on the distance between the users and the webcam.

### b) Better Performance

The response time are heavily rely on the hardware of the machine, this includes the processing speed of the processor, the size of the available RAM, and the available features of webcam. Therefore, the program may have better performance when it's running on a decent machines with a webcam that performs better in different types of lightings.

## BIBLIOGRAPHY

Banerjee, A., Ghosh, A., Bharadwaj, K., & Saikia, H. (2014). Mouse control using a web camera based on colour detection. *arXiv preprint arXiv:1403.4722*.

Chu-Feng, L. (2008). Portable Vision-Based HCI. [online] Available at: [http://www.csie.ntu.edu.tw/~p93007/projects/vision/vision\\_hci\\_p93922007.pdf](http://www.csie.ntu.edu.tw/~p93007/projects/vision/vision_hci_p93922007.pdf) [Accessed 25 Aug. 2015].

Park, H. (2008). A method for controlling mouse movement using a real-time camera. *Brown University, Providence, RI, USA, Department of computer science*.

Kumar N, M. (2011). *Manual Testing: Agile software development*. [online] Manojforqa.blogspot.com. Available at: <http://manojforqa.blogspot.com/2011/09/agile-software-development.html> [Accessed 27 Aug. 2015].

Niyazi, K. (2012). Mouse Simulation Using Two Coloured Tapes. *IJIST*, 2(2), pp.57-63.

Sekeroglu, K. (2010). Virtual Mouse Using a Webcam. [online] Available at: [http://www.ece.lsu.edu/ipl/SampleStudentProjects/ProjectKazim/Virtual%20Mouse%20Using%20a%20Webcam\\_Kazim\\_Sekeroglu.pdf](http://www.ece.lsu.edu/ipl/SampleStudentProjects/ProjectKazim/Virtual%20Mouse%20Using%20a%20Webcam_Kazim_Sekeroglu.pdf) [Accessed 29 Aug. 2015].

Tutorialspoint.com, (n.d.). *SDLC - Agile Model*. [online] Available at: [http://www.tutorialspoint.com/sdlc/sdlc\\_agile\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm) [Accessed 27 Aug. 2015].

Tabernae.com, (n.d.). *Software Life Cycle | Web Development Outsourcing| IT Offshore Outsourcing*. [online] Available at: <http://www.tabernae.com/process.aspx> [Accessed 28 Aug. 2015].

Zhengyou, Z., Ying, W. and Shafer, S. (2001). Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. [online] Available at:

<http://research.microsoft.com/en-us/um/people/zhang/Papers/PUI2001-VisualPanel.pdf> [Accessed 25 Aug. 2015].

# SIMILARITY REPORT

FYP Check Report Check - DUE 23-Sep-2016

Originality GradeMark PeerMark

Report Check  
BY JOHN WAN

turnitin 2% SIMILAR -- OUT OF 0

### 1. PROJECT BACKGROUND

#### 1.1. Introduction

A mouse, in computing terms is a pointing device that detects two-dimensional movements relative to a surface. This movement is converted into the movement of a pointer on a display that allows to control the Graphical User Interface (GUI) on a computer platform. There are a lot of different types of mouse that have already existed in the modern days technology, there's the mechanical mouse that determines the movements by a hard rubber ball that rolls around as the mouse is moved. Years later, the optical mouse was introduced that replace the hard rubber ball to a LED sensor to detects tabletop movement and then sends off the information to the computer for processing. On the year 2004, the laser mouse was then introduced to improve the accuracy movement with the slightest hand movement, it overcome the limitations of the optical mouse which is the difficulties to track high-gloss surfaces. However, no matter how accurate can it be, there are still limitations exist within the mouse itself in both physical and technical terms. For example, a computer mouse is a consumable hardware device as it requires

#### Match Overview

1	Steven Shafer. "Visual ... Publication	1%
2	fossies.org Internet source	<1%
3	www.allegient.com Internet source	<1%
4	blog.aguskurniawan.net Internet source	<1%
5	www.appliedlabsindia.... Internet source	<1%
6	www.coursehero.com Internet source	<1%
7	Bhattacharjee, Abhina... Publication	<1%
8	Suresh, Prashanth, Nir... Publication	<1%
9	"Solution to Exercises"... Publication	<1%
10	biblioteca.utcluj.ro Internet source	<1%

PAGE: 1 OF 42

Text-Only Report