PROJECT REPORT
ON
“Virtual Mouse”

Submitted in partial fulfilment of the requirements

For the award of

Diploma

in

Computer Engineering

Submitted by:
Gaurav
10621192
Section - B
Semester - VI
Year-3rd

Submitted To:
Sh. Sudipta Acharjee Sir
Dept. of Computer Engineering

Tabel Of Content

DECLARATION

I declare that this report entitled "**VIRTUAL MOUSE**" is my own work except as cited inthe 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, Sh. Sudipta Acharjee Sir who has given me this opportunity to engage in an “virtual mouse project”, as well as provided me feedbacks and suggestion to make this project possible. Without his guidance this project would not have been possible.

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 undergo 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 consist 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.

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 theinformation 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 deviceas 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 themouse 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.

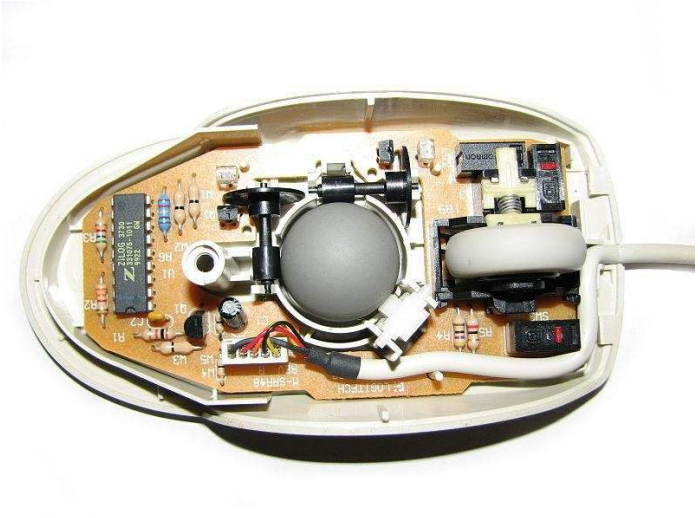


Figure: Mechanical mouse, with top cover removed

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

Advantage	Disadvantage
• Allows the users to control the computer system by moving the mouse. • Provides precise mouse tracking movements	• Prone to degradation of the mouse rollers and button switches, causing to be faulty. • Requires a flat surface to operate.

Table: 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 relyon 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 todetect 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 detectiondifficulties. 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: Optical Mouse, with top cover removed

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

Advantages	Disadvantages
• Allows better precision with lesser hand movements. • Longer life-span.	• Prone to button switches degradation. • Does not function properly while on a polished surface.

Table: 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 wherethat every technological devices can be controlled and interacted remotely withoutusing 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 externalperipheral 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 captureand 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 latestsoftware 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 convertedto 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 afew 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 thanthat, 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 awebcam, 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 positionof 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 oureveryday 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.

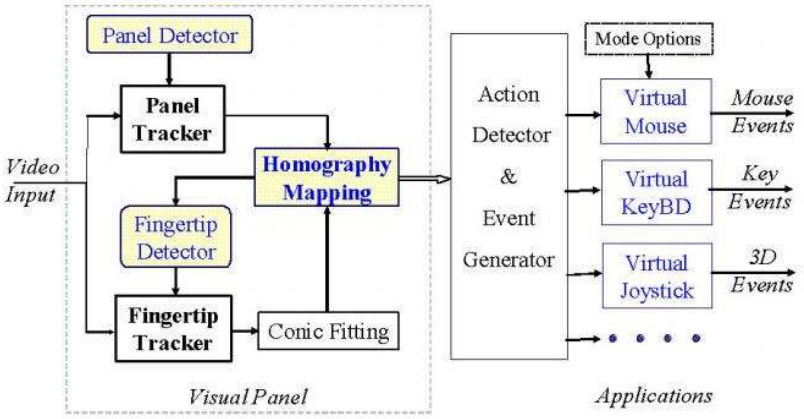


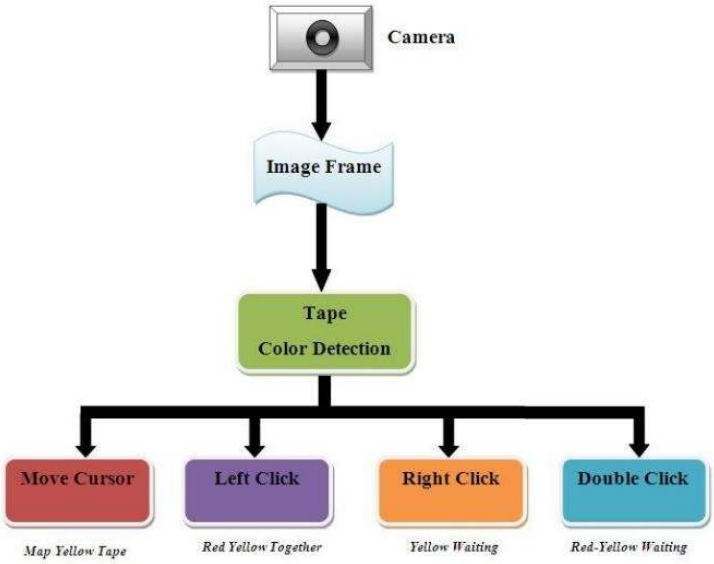
Figure: The system overview of Visual Panel (Zhengyou, Ying and Shafer, 2001)

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 accepts 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: 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 noisewithin the image.

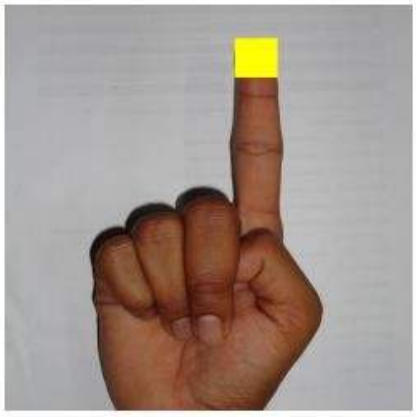


Figure: Yellow colour tape for cursor movement (Niyazi, 2012)

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 clickevents. 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.

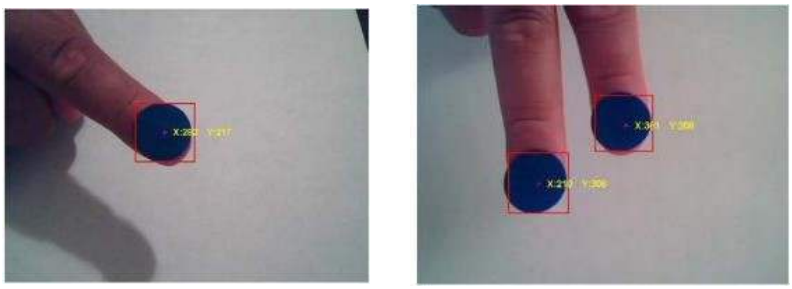


Figure: Input image using one and two pointers (Sekeroglu, 2010)

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'sbuilt 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.

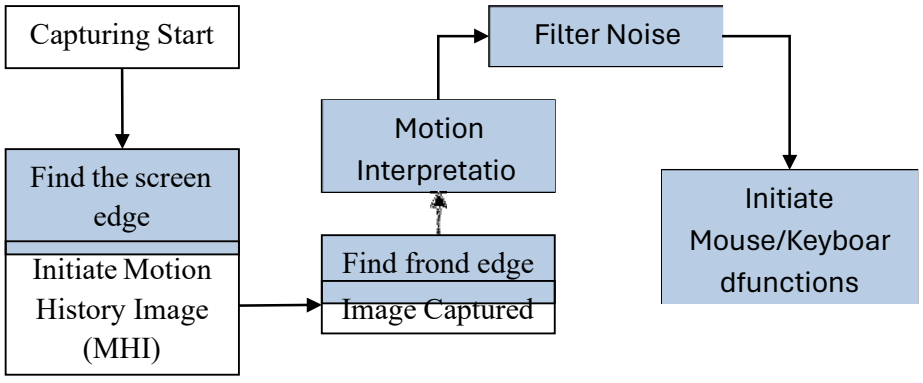


Figure: The Flow Chart of Portable Vision-Based Human Computer Interaction (Chu-Feng, 2008)

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.

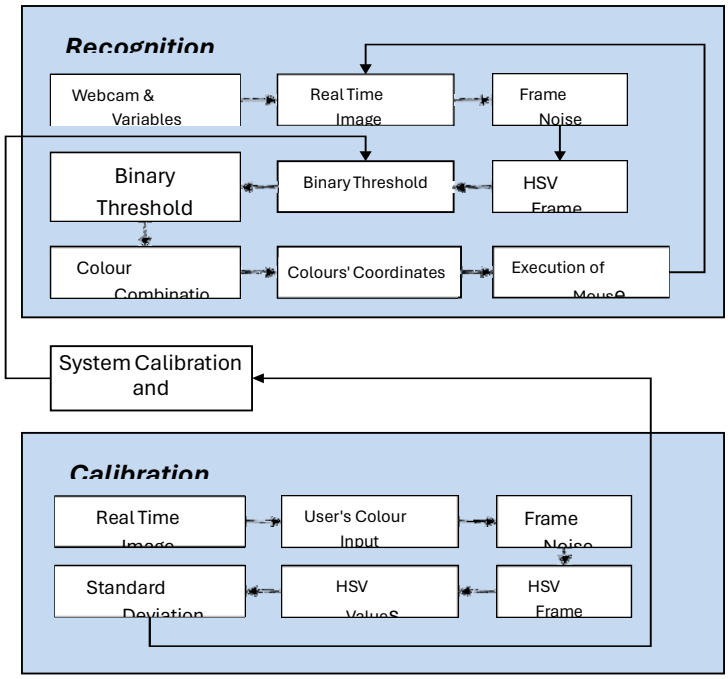


Figure: Virtual Mouse Block Diagram

3.2 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 requiredHSV values that are too small, but still underwent a series of process eventhough it's not large enough to be considered an input. To overcome thisissue, 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 thismethod, 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 beCPU intensive for most of the low-tier system. If the size of the capturedframes is too large, the time-taken for the application to process the entireframe are increase drastically. Therefore, to overcome this issue, theapplication is required to process only the essential part of the frames, andreduces the redundant filtering process that could potentially slow theapplication down.
- The difficulties of calibrating the brightness and the contrast of theframes 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 adaptiveplanning, evolutionary development, early delivery, continuous improvement, andencourages 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 workablesoftware.
- 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

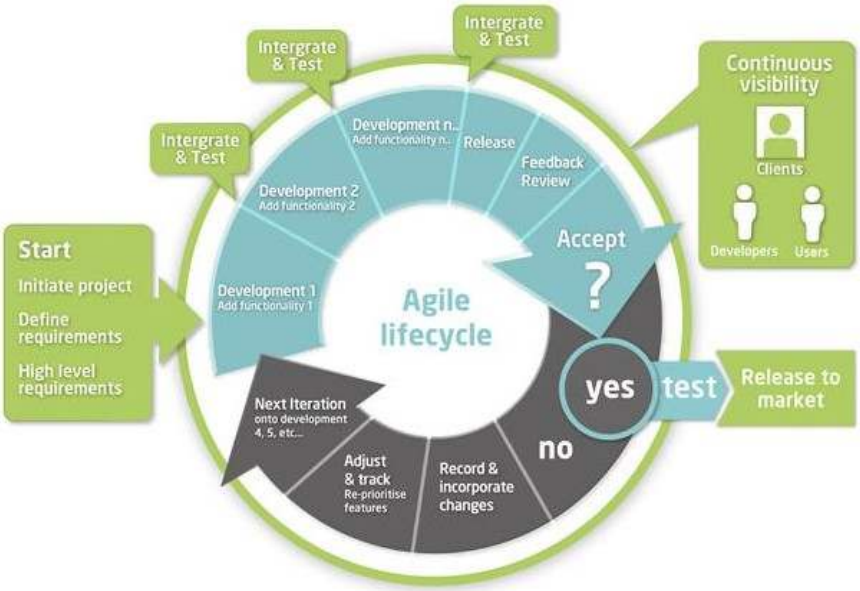


Figure: Agile method overview (Kumar N, 2011)

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, theprototype 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 whereit 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 developthe 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 VirtualMouse 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 (O**pen **S**ource **C**omputer **V**ision) 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 varies whenever the qualities of the captured frames have changed, as it may be affected by different situation in terms of environment, brightness, and the weather. The following describes the situations which may result in false detectionand/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 wholerecognition cycle, verification plan is required to be implemented in order for theprogram to perform flawlessly.

5. IMPLEMENTATION AND TESTING

5.1. Overview

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

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.

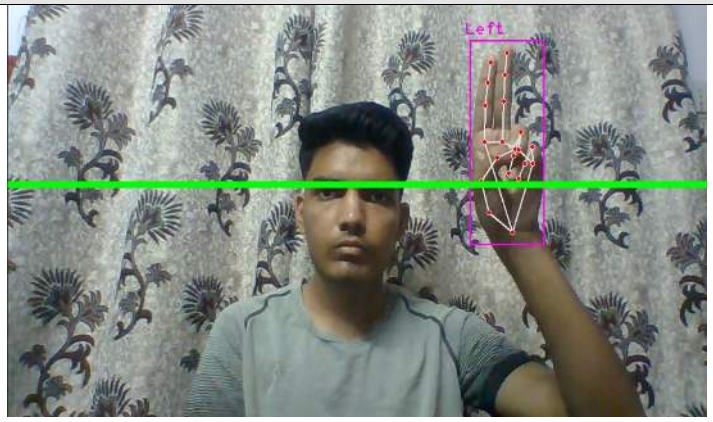
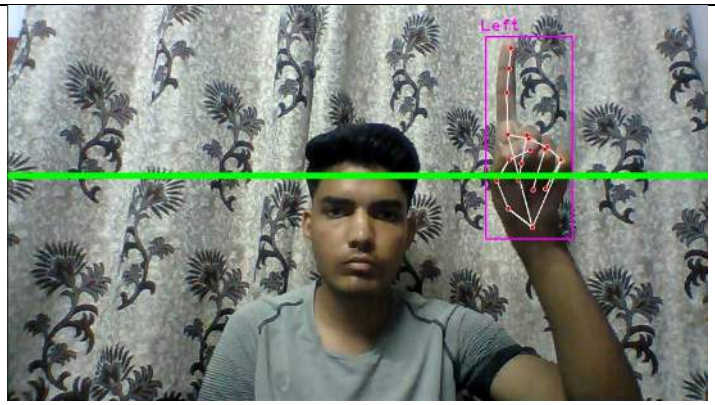
5.2. Main Code

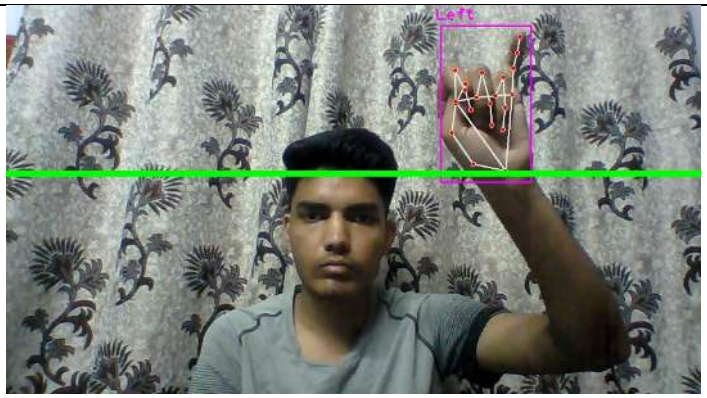
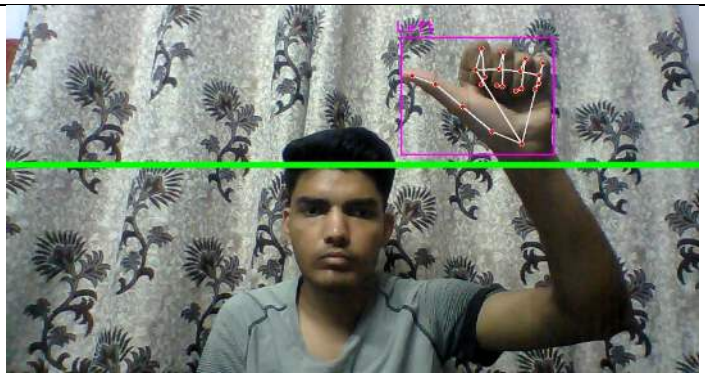
```
from cvzone.HandTrackingModule import HandDetector
import cv2
import os
import numpy as np

# Parameters
width, height = 1280, 720
gestureThreshold = 300
folderPath = "Presentation"

# Camera Setup
cap = cv2.VideoCapture(0)
cap.set(3, width)
cap.set(4, height)

# Hand Detector
detectorHand = HandDetector(detectionCon=0.8, maxHands=1)

# Variables
imgList = []
delay = 30
buttonPressed = False
counter = 0
drawMode = False
imgNumber = 0
delayCounter = 0
annotations = [[]]
annotationNumber = -1
annotationStart = False
hs, ws = int(120 * 1), int(213 * 1)  # width and height of small image

# Get list of presentation images
pathImages = sorted(os.listdir(folderPath), key=len)
print(pathImages)

while True:
    # Get image frame
    success, img = cap.read()
    img = cv2.flip(img, 1)
    pathFullImage = os.path.join(folderPath, pathImages[imgNumber])
    imgCurrent = cv2.imread(pathFullImage)
```

```
    # Find the hand and its landmarks
    hands, img = detectorHand.findHands(img)  # with draw
    # Draw Gesture Threshold line
    cv2.line(img, (0, gestureThreshold), (width, gestureThreshold), (0, 255, 0), 10)

    if hands and buttonPressed is False:  # If hand is detected

        hand = hands[0]
        cx, cy = hand["center"]
        lmList = hand["lmList"]  # List of 21 Landmark points
        fingers = detectorHand.fingersUp(hand)  # List of which fingers are up

        # Constrain values for easier drawing
        xVal = int(np.interp(lmList[8][0], [width // 2, width], [0, width]))
        yVal = int(np.interp(lmList[8][1], [150, height-150], [0, height]))
        indexFinger = xVal, yVal

        if cy <= gestureThreshold:  # If hand is at the height of the face
            if fingers == [1, 0, 0, 0, 0]:
                print("Left")
                buttonPressed = True
                if imgNumber > 0:
                    imgNumber -= 1
                    annotations = [[]]
                    annotationNumber = -1
                    annotationStart = False
            if fingers == [0, 0, 0, 0, 1]:
                print("Right")
                buttonPressed = True
                if imgNumber < len(pathImages) - 1:
                    imgNumber += 1
                    annotations = [[]]
                    annotationNumber = -1
                    annotationStart = False

        if fingers == [0, 1, 1, 0, 0]:
            cv2.circle(imgCurrent, indexFinger, 12, (0, 0, 255), cv2.FILLED)

        if fingers == [0, 1, 0, 0, 0]:
            if annotationStart is False:
                annotationStart = True
                annotationNumber += 1
                annotations.append([])
            print(annotationNumber)
            annotations[annotationNumber].append(indexFinger)
            cv2.circle(imgCurrent, indexFinger, 12, (0, 0, 255), cv2.FILLED)

        else:
            annotationStart = False

        if fingers == [0, 1, 1, 1, 0]:
            if annotations:
                annotations.pop(-1)
                annotationNumber -= 1
                buttonPressed = True

    else:
        annotationStart = False

    if buttonPressed:
        counter += 1
        if counter > delay:
            counter = 0
            buttonPressed = False
```

```
    for i, annotation in enumerate(annotations):
        for j in range(len(annotation)):
            if j != 0:
                cv2.line(imgCurrent, annotation[j - 1], annotation[j], (0, 0, 200), 12)

    imgSmall = cv2.resize(img, (ws, hs))
    h, w, _ = imgCurrent.shape
    imgCurrent[0:hs, w - ws: w] = imgSmall

    cv2.imshow("Slides", imgCurrent)
    cv2.imshow("Image", img)

    key = cv2.waitKey(1)
    if key == ord('q'):
        break
```

5.3. Outcomes

Sample Layout	Mouse Functions
	Move Cursor
	Draw

	Swipe Right
	Swipe Right