VIRTUAL MOUSE APPLICATION

Roshan Hyalij[1], Siddharth Oturkar[2], Atharv Kasodekar[3] , Varun Kothawade[4] , Kamlesh Patil[5], Prof.Aashutosh kale[6]

[1-5]Dept of Computer Engineering, MET Institute of Engineering, Nashik.
[6]Professor, Dept. of Computer Engineering, MET Institute of Engineering, Nashik

---***---

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 color recognition program will constantly be acquiring real-time images where the images will be 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 colors position from the converted frames. After that, it will proceed to compare the existing colors within the frames with a list of color combinations, where different combinations consist of different mouse functions. If the current colors 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. INTRODUCTION

In today's world we see lots of development happened in the field of Technology. Today's technology is combined with the technique called Artificial Intelligence. This project is also based on small part of AI. This project presents finger movement gesture detection on our computer's window using camera & handling the whole system by just moving your one finger.

Using finger detection methods for instant camera access and user-friendly user interface makes it more easily accessible. The system is used to implement motion tracking mouse, a signature input device and an application selector. This system reduces the use of any physical mouse which saves time and also reduces efforts.

1.1 PROBLEM STATEMENT

- To design motion tracking mouse which detect finger movements gestures instead of physical mouse.

- To design an application (.exe file) with user friendly user interface which provides feature for accessing motion tracking mouse feature.

- The camera should detect all the motions of hand and performs the operation of mouse.

- Implement such code where motion tracker mouse has drag & drop feature along with scrolling feature.

- User Interface must be Simple & easy to understand.

- 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.

- Implement such code where camera can recognize each and every finger movement & responds according to it:

1.2 MOTIVATION

- Create such application which introduce use of AI world in small manner

- Exploring vision-based interfaces is motivated by the unnaturalness of some of the conventional input devices such as mice and joysticks in many intelligent environments where intuitive interactions and teleoperations are required.

- 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 device can be controlled and interacted remotely without using any peripheral devices such as the remote, keyboards, etc. it doesn't just provide convenience, but it's cost effective as well.

- 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.

• The motivation is to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device.

1.3. OBJECTIVES

• Create such application which is part of AI.

• To design to operate with the help of a webcam.

• User should able to easily install in their computer.

• User should able to use feature of Drag & Drop.

• Also, it must have Scrolling feature.

• To design a virtual input that can operate on all surface.

• To convert hand gesture/motion into mouse input that will be set to a particular screen position.

• UI of application should be easy to use.

• Program should run as fast as possible without any lag.

• There should be no heavy task which can disturb user.

1.4. REQUIREMENT ANALYSIS

1. Problem Recognition: To implement application which can access camera of system & start detecting motion of finger to act as a cursor of a mouse to handle whole system.

2. Evaluation & Synthesis:

• The problem statement should be divided into two parts, one is implementation of user

Interface & second is implementation of code which can access camera and movements of finger.

• After working on both parts separately, at the end just Integrate it to look it more neat and good.

• At the end converting all this code in .exe file. So, it should be sharable and everyone can use it.

3. Modeling:

• Data Design: In this particular, we have to collect all the required data like how many finger inputs we required, how many fingers we required for motion tracking. Also we have to implement how many parts we want to display on UI.

• System Models: System should have Web camera support to trace motion of hand. Also, it has some python libraries preinstalled so it should be easy to access.

• UI Design: To implement user interface first of all we have to decide environment required for the development of the application. After that according to architecture we should start implementing the design

4. Specification:

• Use cases: After developing this application user should be able to access their system through Motion Tracker Application.

• User are able to use this Application to maintain eye distance between their device and himself.

Also, during streaming he/she can able to access their system without any movement of their body.

5. Review: We are developing such application which is combination of AI & Web. After completing this project user can access their system with the help of their finger by using system's camera.

2. SOFTWAREREQUIREMENT SPECIFICATIONS(SRS)

• SRS of Motion Tracker Application:

a) Purpose of Application: Motion Tracker Application is the application software where after clicking on TRACK option which is given after just opening of application, instead of using physical mouse user can use their finger as a mouse & whatever the direction we give to our finger according to that cursor on the system move & user can do their work.

b) How application works: This application is combination of UI & AI. First of all, user open application which is .exe file. After opening of file, user interface appears after that user have option of TRACK, after clicking on that button our backend program gets executed where that code contains accessing of camera to finger tracking motions.

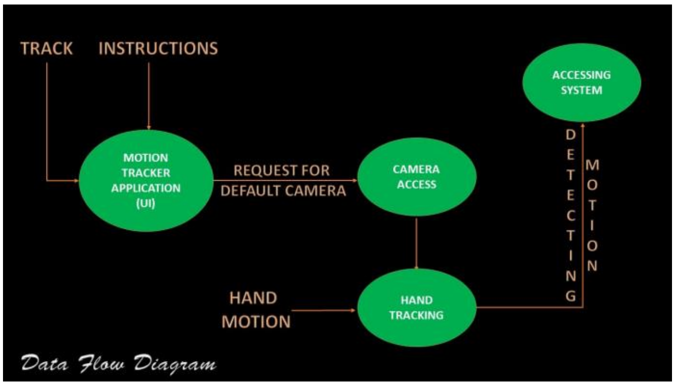
c) Functional Requirement: After developing this application user should be able to access their system through Motion Tracker Application. Calculations required for this application is all related to motion detect operations & Data processing is done by using different python libraries like NumPy & media-pipe.

d) Interface Requirements: This application communicates with UI & Python code with the help of TKINTER of python. Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.

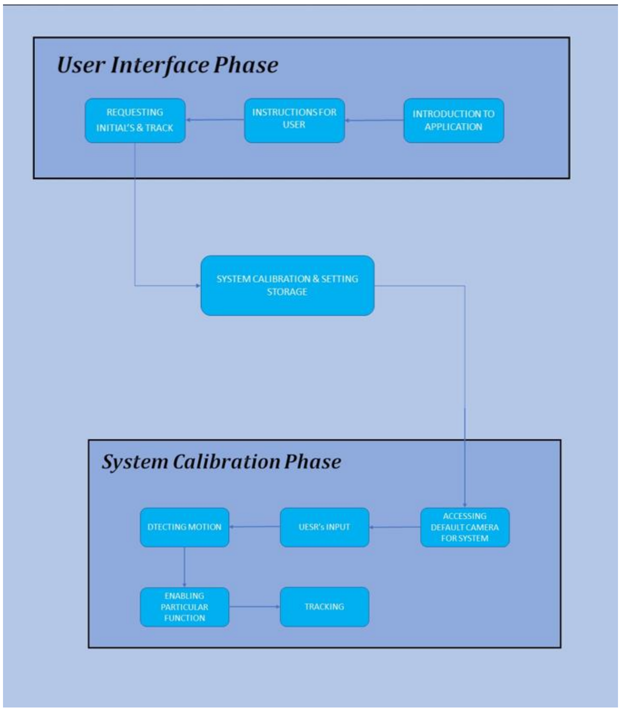
e) Overview: We are developing such application which is combination of AI & Web. Implement such code where camera can recognize each and every finger movement & responds according to it. After completing this project user can access their system with the help of their finger by using system's camera. System should have Web camera support to trace motion of hand. Also, it has some python libraries preinstalled so it should be easy to access.

3. DATA FLOW DIAGRAM (DFD)

Description of Motion Tracker Application:



4. METHODOLOGY & PROPOSED SYSTEM BLOCK DIAGRAM


User Interface Phase
REQUESTING INITIAL'S & TRACK
INSTRUCTIONS FOR USER
INTRODUCTION TO APPLICATION
SYSTEM CALIBRATION & SETTING STORAGE
System Calibration Phase
DTECTING MOTION
UESR's INPUT
ACCESSING DEFAULT CAMERA FOR SYSTEM
ENABLING PARTICULAR FUNCTION
TRACKING

Description of each block:

a) User Interface Block:

a. Introduction to Application: In this block, when user start MOTION TRACKER

APPLICATION then first window prompts as Introduction window where some basic information is displayed.15

b. Instructions For User: User should find it easy to use application, so we have added two important points related to using of application. By reading these two instructions user is able to use system easily.

c. Track (Requesting Initials): After reading of instructions, now user able to use TRACKING feature, so by just clicking on TRACK button, system automatically gets access to systems camera for further communication.

b) System Calibration & Setting Storage:

a. In this block, system check for all requirements which are need to detect hand movement.

According to program system auto detects all the hardware requirement & sends request to the system for access.

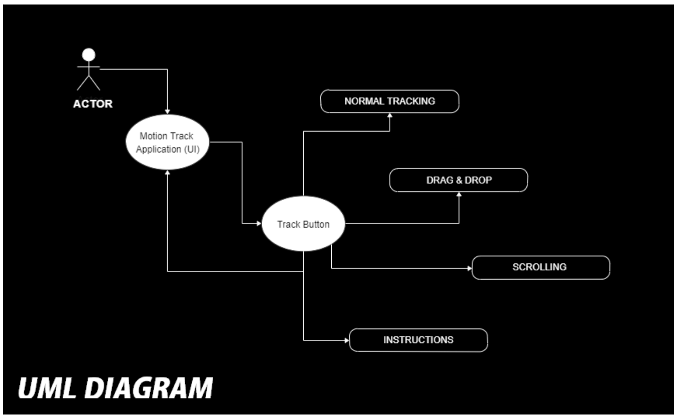
c) System Calibration Phase:

a. Accessing Default Camera: Just after request came from system, Default camera which is set automatically by system is start giving access & camera gets turned on.

b. User Input: After starting of camera, now user is able to move his/her finger according to his/her choice & this movement camera detects and take it as a Input for motion.

c. Detecting Motion, enabling particular function & Tracking: When user starts moving his/her finger, camera captures motion and python libraries such as media pipe, NumPy helps to run code & according to input given by user, system calls particular function wrote in program and start's executing particular code which results in the motion of cursor.

5. UML DIAGRAM



6. SOFTWARE AND HARDWARE REQUIREMENT

A) Hardware Requirement:

a) 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 of the application.

System will be using Processor: Core2Duo

Main Memory: 2 GB RAM (Minimum)

Hard Disk: 512 GB (Minimum**)**

Display: 14" Monitor (For more comfort)

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.

B) Software Requirement:

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

Python: To access camera & tracking all hand motion, python is very easy & accurate to use. Python comes with lots of build in libraries which makes code short and easily understandable. Python version required for building of this application is 3.7

Open CV Library: OpenCV are also included in the making of this program.

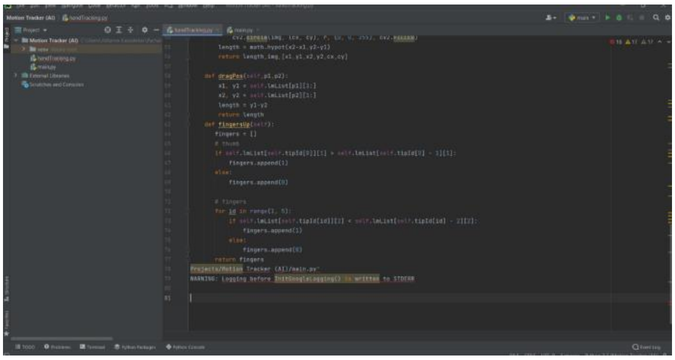
OpenCV (Open-Source Computer Vision) is a library of programming functions for real time computer vision. OpenCV have the utility that can read image pixels value, it also has the ability to create real time eye tracking and blink detection.

Tkinter: The tkinter package is the standard Python interface to the Tk GUI toolkit.

Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems. To make UI for application we used Tkinter.

7. IMPLEMENTATION

Implementing python code by using pycharm

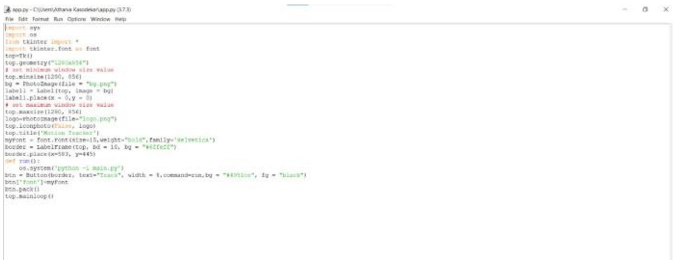


```
        cv2.circle(img, (cx, cy), r, (0, 0, 255), cv2.FILLED)
        length = math.hypot(x2-x1,y2-y1)
        return length,img,[x1,y1,x2,y2,cx,cy]

    def dragPos(self,p1,p2):
        x1, y1 = self.lmList[p1][1:]
        x2, y2 = self.lmList[p2][1:]
        length = y1-y2
        return length
    def fingersUp(self):
       fingers = []
       # thumb
       if self.lmList[self.tipId[0]][1] > self.lmList[self.tipId[0] - 1][1]:
           fingers.append(1)
       else:
           fingers.append(0)

       # fingers
       for id in range(1, 5):
           if self.lmList[self.tipId[id]][2] < self.lmList[self.tipId[id] - 2][2]:
               fingers.append(1)
           else:
               fingers.append(0)
       return fingers
Projects/Motion Tracker (AI)/main.py"
WARNING: Logging before InitGoogleLogging() is written to STDERR
```

Implementing UI code by Using Tkinter



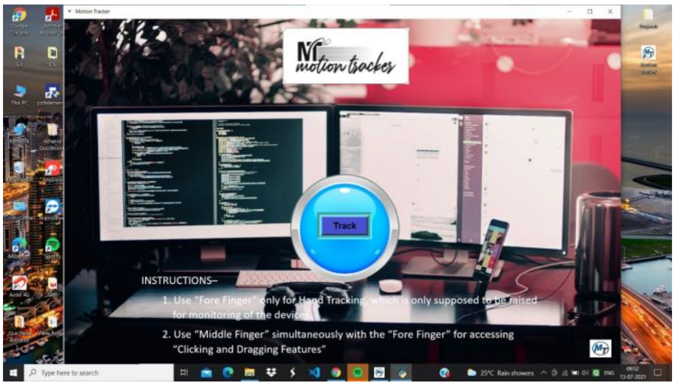
app.py - C:\Users\Atharva Kasodekar\app.py (3.7.3)

```
import sys
import os
from tkinter import *
import tkinter.font as font
top=Tk()
top.geometry("1280x856")
# set minimum window size value
top.minsize(1280, 856)
bg = PhotoImage(file = "bg.png")
label1 = Label(top, image = bg)
label1.place(x = 0,y = 0)
# set maximum window size value
top.maxsize(1280, 856)
logo=PhotoImage(file="logo.png")
top.iconphoto(False, logo)
top.title('Motion Tracker')
myFont = font.Font(size=15,weight="bold",family='Helvetica')
border = LabelFrame(top, bd = 10, bg = "#6ffeff")
border.place(x=583, y=445)
def run():
    os.system('python -i main.py')
btn = Button(border, text="Track", width = 8,command=run,bg = "#49b1ce", fg = "black")
btn['font']=myFont
btn.pack()
top.mainloop()
```

8. RESULT AND TEST CASES

Main application (executable application)

Main Window



INSTRUCTIONS–

1. Use "Fore Finger" only for Hand Tracking, which is only supposed to be raised for monitoring of the device.

2. Use "Middle Finger" simultaneously with the "Fore Finger" for accessing "Clicking and Dragging Features"

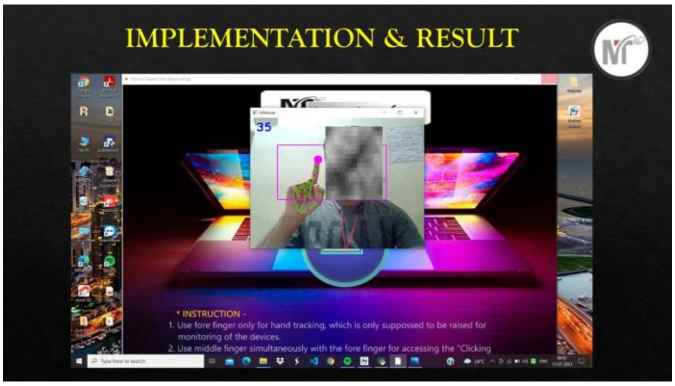
After clicking on icon, the main window of Motion Tracking Application appears. Which consist of "Track Button" and few "Important Instructions" at the bottom of the Window.

Instructions –

1. Use "Fore Finger" only for Hand Tracking, which is only supposed to be raised for monitoring of the devices.

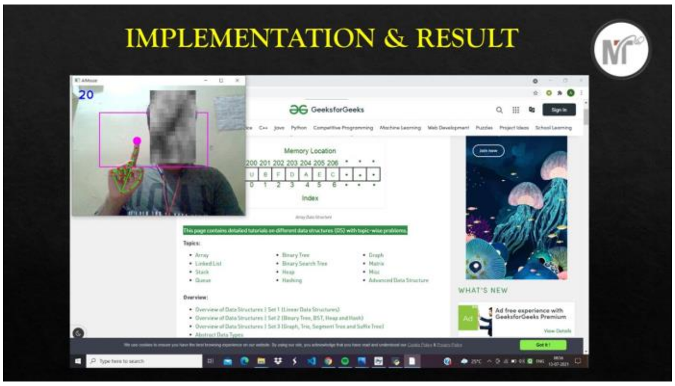
2. Use "Middle Finger" simultaneously with the "Fore Finger" for accessing "Clicking and Dragging Features.

Once after clicking on track button, the camera of your device will be automatically accessed by this application and a small window will pop up in which "Your hand will start getting monitored.



IMPLEMENTATION & RESULT

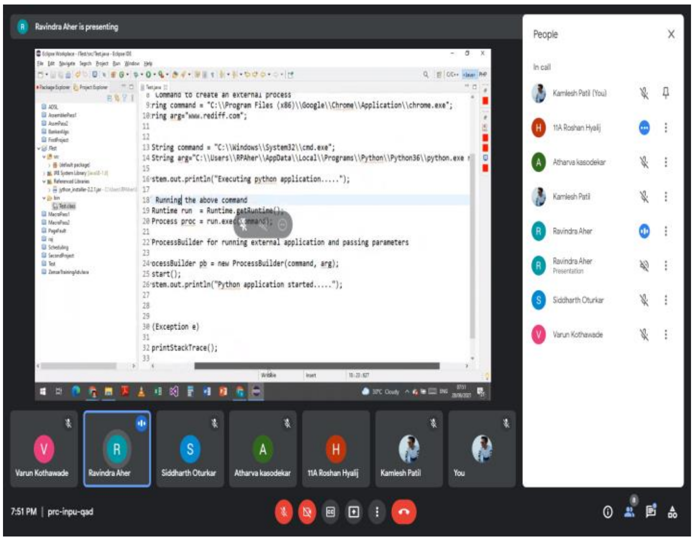
* INSTRUCTION -
1. Use fore finger only for hand tracking, which is only suppossed to be raised for monitoring of the devices.
2. Use middle finger simultaneously with the fore finger for accessing the "Clicking



IMPLEMENTATION & RESULT

Finally your Hand Tracking gets started. Now you can start moving the cursor and can use the features like "Clicking and Dragging".

DISCUSSION RELATED INTEGRATION PART WITH STAFF



Ravindra Aher is presenting

```
 Command to create an external process
ring command = "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe";
ring arg="www.rediff.com";


String command = "C:\\Windows\\System32\\cmd.exe";
String arg="C:\\Users\\RPAher\\AppData\\Local\\Programs\\Python\\Python36\\python.exe 

stem.out.println("Executing python application.....");

 Running the above command
Runtime run  = Runtime.getRuntime();
Process proc = run.exec(command);

ProcessBuilder for running external application and passing parameters

ocessBuilder pb = new ProcessBuilder(command, arg);
start();
stem.out.println("Python application started.....");



(Exception e)

printStackTrace();
```

9. CHALLENGES FACED

1) Accessing Web cam: The first challenge was to correctly detect the hand with a webcam. We needed a Computer Vision library for this purpose. Many are available but we decided to go ahead with OpenCV as it is the most popular and has been ported to many languages and is supported on many operating systems from Android to Windows.

2) Choosing Libraries: After deciding library for accessing camera, next challenge is for hand tracking & colored animation requirement. To do this particular we have decided to go with libraries of python like numpy & mediapipe. This both libraries are very popular & suitable specially for windows application. .

3) Choosing Appropriate Python Version: To access all python libraries and communicate them with other libraries is main task, for that we need to decide python particular version which can do this accurately. So, we decided to go two python version back which is python 3.7 This version accurately integrates numpy, mediapipe & opencv library.

4) Integrating Python Code & UI: Most big challenge until now is Integration part i.e. How to Integration with attractive user interface. After searching lot about this we finally decided to go with Tkinter package of python. Tkinter is very popular package for creating simple & attractive user Interfaces.

10. CONCLUSIONS

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. After implanting such type of application there is big replacement of physical mouse i.e., there is no need of any physical mouse. Each & every movement of physical mouse is done with this motion tracking mouse (virtual mouse

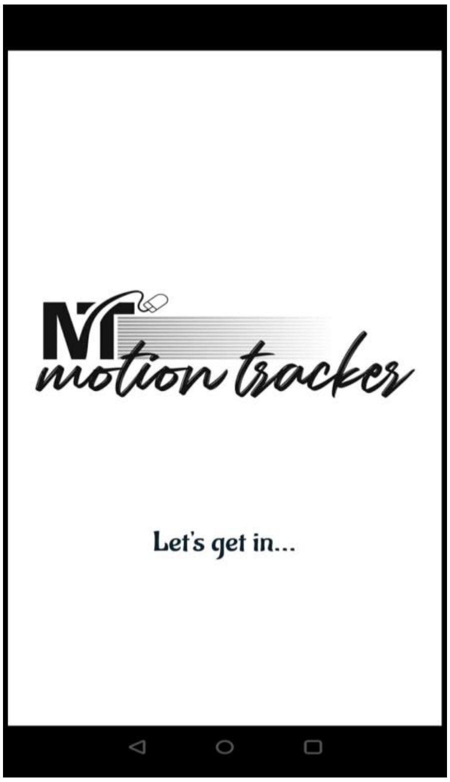
11. FUTURE SCOPE

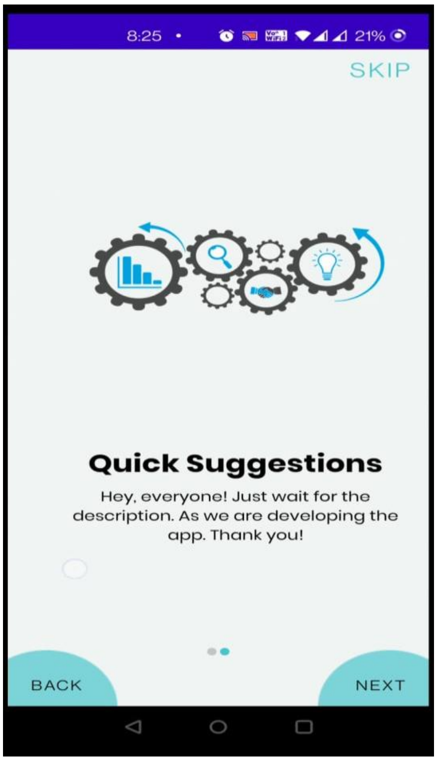
There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments. The following describes the improvements and the features required:

a) Smart Movement: 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.

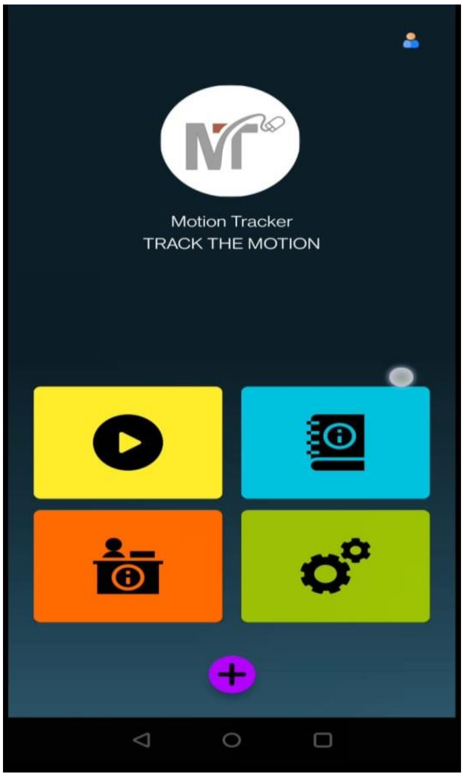
b) Better Accuracy & Performance: The response time are heavily relying on the hardware of the machine, this includes the processing speed of the processor, the size of the available RAM, and the available features of webcam. Therefore, the program may have better performance when it's running on a decent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

c) Mobile Application: In future this web application also able to use on Android devices, where touchscreen concept is replaced by hand gestures.





Hey, everyone! Just wait for the description. As we are developing the app. Thank you!



ACKNOWLEDGEMENT

It gives us great pleasure in presenting the project report on "MOTION TRACKER APPLICATION" .This Project was fully based on virtual hand monitoring system application software which can be used in Laptop, Mobiles, Tablets and also in some digital screen devices which consists of camera. We would like to take this opportunity to thank our internal guide for this project Prof. Ashutosh Kale for giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful. We are also grateful to Prof. M. U. Kharat, Head of Computer Engineering Department, College. for his indispensable support and suggestions .

REFERENCES

[1] International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 1– Mar 2014 ISSN: 2231-2803 www.internationaljournalssrg.org Page 15 Mouse Control using a Web Camera based on Colour Detection.

[2] K N. Shah, K R. Rathod and S. J. Agravat, "A survey on Human Computer Interaction Mechanism Using Finger Tracking" International Journal of Computer Trends and Technology, 7(3), 2014, 174-177

[3] Tutorialspoint.com, (n.d.). SDLC - Agile Model. [online] Available athttp://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm

[4] Python GUI Programming With Tkinterhttps://realpython.com/python-gui-tkinter/

[5] Python numpy ,https://numpy.org/

[6] The Python Standard Library https://python.readthedocs.io/en/latest/library/index.html

[7] The MATLAB website. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/2 8757-tracking-red-color-objects-using-matlab

[8] Pycharm

https://www.jetbrains.com/pycharm .