

VIRTUAL MOUSE

A MTE Project report
Submitted in the partial fulfillment of the requirements
for the Award of Degree
OF

Master of Technology
In
INFORMATION TECHNOLOGY

Submitted by:
RAUNAK PODDAR
2K21/ISY/20
KARAN YADAV
2K21/ISY/11

Under the supervision of
Dr. Dinesh K. Vishwakarma



DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi – 110042

NOVEMBER 2021

INTRODUCTION

One of the important challenges in Human Computer Interactions is to develop more intuitive and more natural interfaces. A lot of research is being done in the fields of Human Computer Interaction (HCI) and Robotics. Researchers have tried to control mouse movement using video devices for HCI. However, all of them used different methods to make mouse cursor movement and clicking events. One approach, used index finger for cursor movement and angle between index finger and thumb for clicking, used finger tip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region. Another one is only the fingertips to control the mouse cursor and click. Clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Another method to click is the motion of the thumb (from a „thumbs-up“ position to a fist) to mark a clicking event thumb.

Movement of the hand while making a special hand sign moved the mouse pointer. One more method is Real-time system that can track the 3D position and 2D orientation of the thumb and index finger of each hand without the use of special markers or gloves. System could be used for single pointing and pinching gestures. Another is a finger counting system to control behaviour of a robot. A study of the existing systems for on-screen choice selection reveals that people are still limited to the use of devices such as mouse, touchpad, joystick, trackball and touch screen. All these devices need contact of hand with them.

In our project we tried to implement mouse movement using index finger and middle finger. The webcam will first detect the user's hand and try to find out the positions of the fingers. If only our index finger tip is up then our virtual mouse is in movement mode.

We can able to move our mouse cursor using our index finger tip. If both index finger and middle finger is up then our virtual mouse is in clicking mode. For clicking we need to move our index finger and middle finger close to each other. When the distance between the index finger and middle finger is become shorter the click operation will be performed. The movement and clicking is done with the help of autopsy library in python. Our project is basically based on the concept that how we can control the mouse without having physical contact.

IMPLEMENTATION OF PROJECT

LIBRARIES USED

- ▶ OpenCV: - OpenCV is used for image processing and drawing.
- ▶ Media pipe:- We are using Media pipe for hand detection and tracking.
- ▶ Autopy: - For controlling the mouse movements and for performing clicks. Autopy have several functions which will control the mouse movements.
- ▶ NumPy: - NumPy is used to perform certain mathematical calculations and for working with arrays.
- ▶ Time: - Used to calculate FPS (Frames per Second).

Implementation

For implementing our virtual mouse, we have followed the following points

Point 1: We will first detect our hand (If there are multiple hands then system will detect only one hand).

```
success, img = cap.read()
img = detector.findHands(img)
lmList, bbox = detector.findPosition(img)
```

Point 2: We will find out the co-ordinates of the index finger and the middle finger.

```
x1, y1 = lmList[8][1:] #Getting the co-ordinates of index finger
x2, y2 = lmList[12][1:] #Getting the co-ordinates of middle finger
```

Point 3: Now we will check which fingers are up.

```
fingers = detector.fingersUp()
#Point 3: Here we are checking which finger is up
```

Point 4: We will convert the co-ordinates of the index finger and middle finger.

```
# Point 4: Converting the coordinates
x3 = np.interp(x1, (frameR, wCam-frameR), (0, wScr))
y3 = np.interp(y1, (frameR, hCam-frameR), (0, hScr))
```

Point 5: We will also apply some smoothening techniques such that our cursor don't fluctuates too much. Taking smoothening value as 7 (Random value).

```
clocX = plocX + (x3 - plocX) / smoothening  
clocY = plocY + (y3 - plocY) / smoothening
```

Point 6: If our index finger is up and middle finger is down then we are in moving mode and we can now move the mouse cursor.

```
#Point 4: To enable moving of mouse cursor we are checking for the index finger tip  
up  
if fingers[1] == 1 and fingers[2] == 0:  
    autopy.mouse.move(wScr - clocX, clocY)
```

Point 7: If both index finger and middle fingers are up then we are in clicking mode.

```
if fingers[1] == 1 and fingers[2] == 1:
```

Point 8: We will find out the distance between the index finger and middle finger.

```
length, img, lineInfo = detector.findDistance(8, 12, img)
```

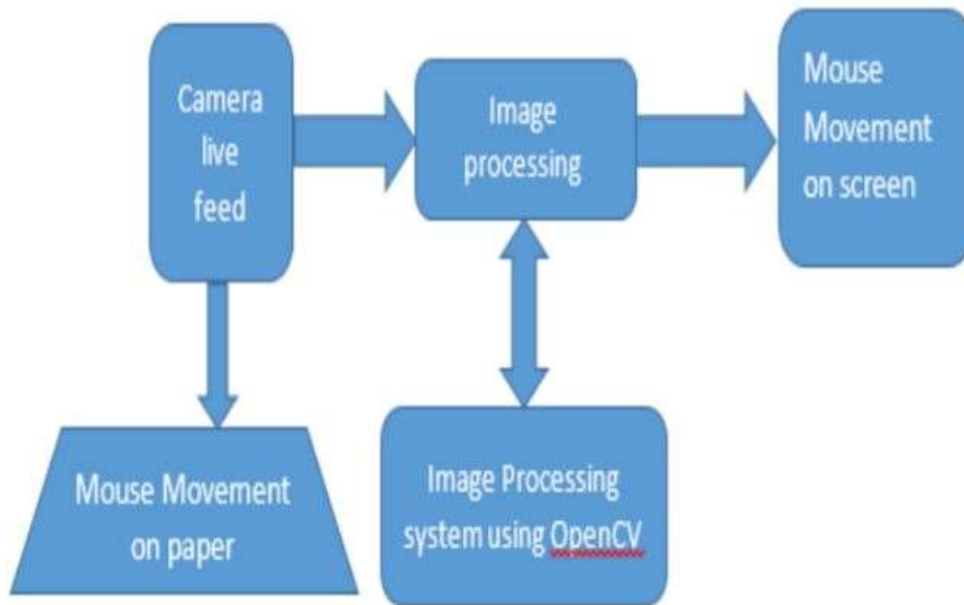
Point 9: To perform clicking operation we need move index finger and middle finger close to each other such that the distance between them decreases. If the distance between them decreases then we will perform clicking operation.

```
if length < 40:  
    cv2.circle(img, (lineInfo[4], lineInfo[5]), 15, (0, 255, 0), cv2.FILLED)  
    autopy.mouse.click()
```

Point 10: We are also calculating fps (Frames per second) and displaying it in the left top of the screen.

```
cTime = time.time()  
fps = 1/(cTime-pTime)  
pTime = cTime  
cv2.putText(img, str(int(fps)), (28, 58), cv2.FONT_HERSHEY_PLAIN, 3, (255, 8, 8), 3)
```

SYSTEM ARCHITECTURE



The camera captures the live feed of the user. It sends the image for the processing. The processing of image is done by OpenCV and other libraries. After the processing we can able to perform movement of mouse and clicking.

In the object tracking application one of the main problems is object detection. Finger tips has been used to make the object detection easy and fast. To simulate the click events of the mouse the fingertips of index finger and middle finger is required.

Major applications:

- Robot controlling is one of the major applications of this system. Controlling robots without machines or extra devices can be an excellent addition to this technology.
- . • Critical events like a battleground, operation theater, mining fields can be controlled by gesture mouse.
- Virtual Reality or Augmented Reality based games can be played without any extended or wireless devices with bare hands more comfortably.
- For patients with no limb control, this system can be proved very useful and effective.
- This mouse system can be applicable for sign language for the dumb & deaf people. It can help them to interact with computing machines.

OUTPUT & RESULTS:

When we keep our hand in front of camera it is detecting the hand and showing landmarks. With the help of autopsy we can able to detect the fingers and fingertips.

We will check for the index number 8 (fingertip of index finger) and index number 12 (fingertip of middle finger) as shown in Fig 1.

We will perform operation by checking which finger is up.

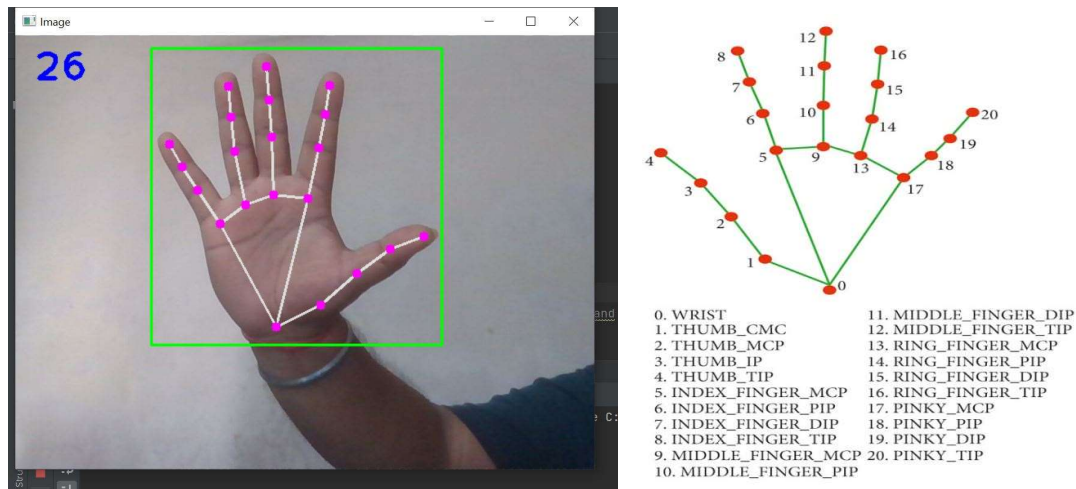


Fig 1: a) Hand Detection and landmarks b) Landmarks or co-ordinates of Hand

If the index finger is up then our mouse is in movement mode and we can move our mouse with our fingertip as shown in Fig 2.

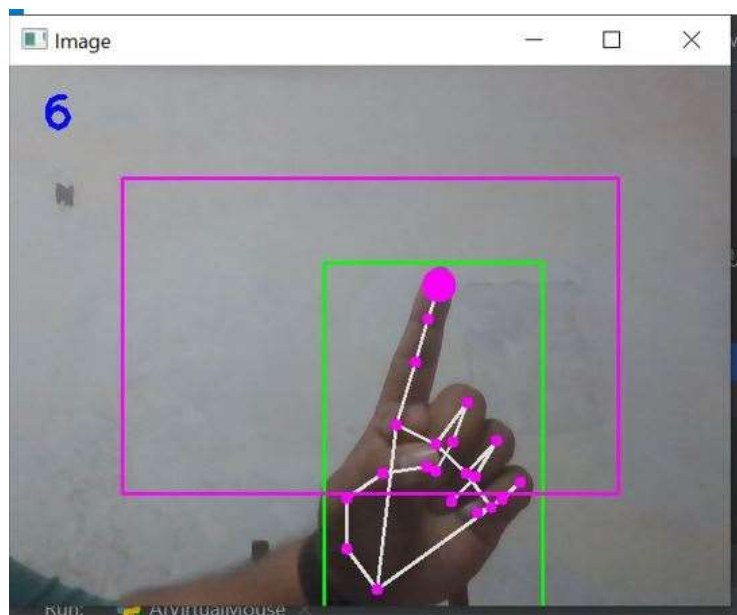


Fig 2: Performing movement operation

If both index finger and middle finger is up then we have clicking mode. By moving both the finger closer i.e., to the red circle as shown in Fig 3 we can able to perform click operation.

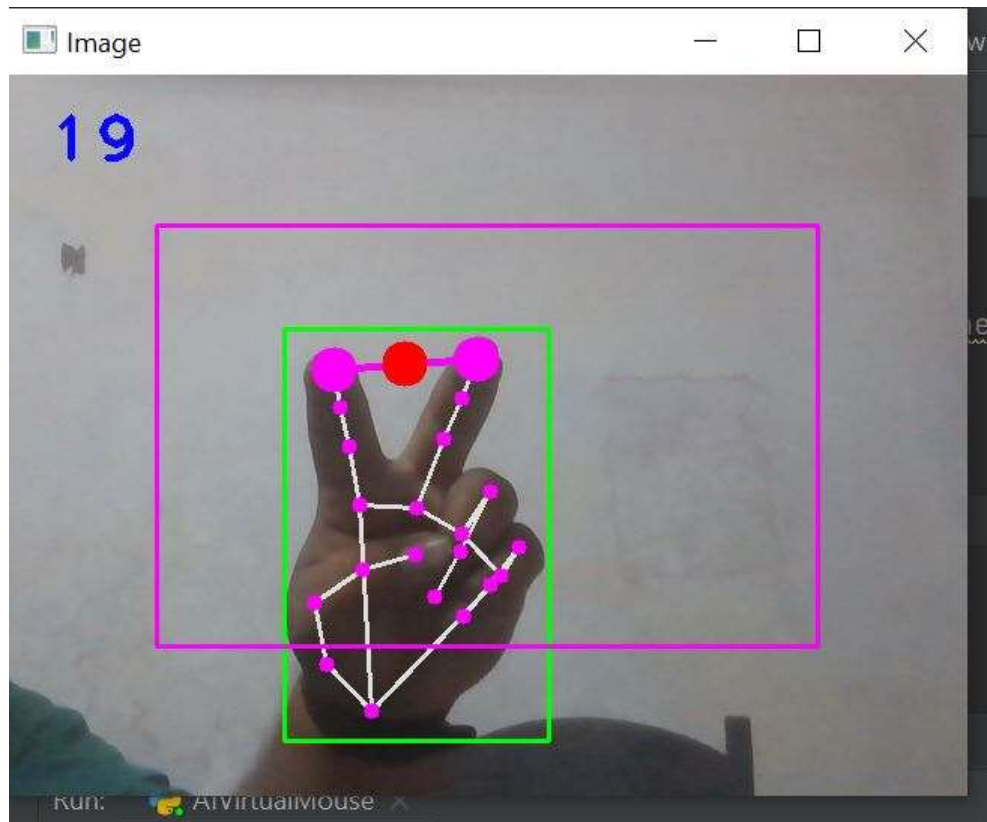


Fig 3: Performing click operation

CONCLUSION

This project represents concept of image processing which is having wider scope in recent years. We are presenting the concept of virtual mouse . In this study, an object tracking based virtual mouse application has been developed and implemented using a webcam. Virtual mouse is capable of performing accurate control of remote display and simulating mouse.

REFERENCES

- [1] Report, December 18, 2003 [1] Hojoon Park, A Method for Controlling Mouse Movement using a Real-Time Camera, 2008
- [2] Chu-Feng Lien, Portable Vision-Based HCI - A Real-time Hand Mouse System on Handheld Devices.
- [3] Robertson P., Laddaga R., Van Kleek M., Virtual mouse vision based interface, January 2004
- [4] Shahzad Malik, Real-time Hand Tracking and Finger Tracking for Interaction CSC2503F Project
- [5] Asanterabi Malima, Erol Ozgur, and Mujdat Cetin, A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control.