International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

CURSOR MOVEMENT USING HAND GESTURE

Shweta Sanjay Chavan, Rutuja Ramchandra Devkate, Shivani Kumari Chaudhary, Vishakha Pratap Mane,
Guide -Miss Sonali sathish malame
Department of Computer science and engineering
SITCOE Ichalkaranji

ABSTRACT

The goal of the paper is to improve the recognition of the human hand postures in a Human Computer Interaction application, the reducing of the time computing and to improve the user comfort regarding the used human hand postures. The authors developed an application for computer mouse control. The application based on the proposed algorithm, hand pad color and on the selected hand feature presents good behavior regarding the time computing. The user has an increased comfort in use of the system due to the proposed hand postures. Also, the system works well having the same behavior under very low illuminance level and high illuminance level.

Keywords

Hand gesture, colour detection, human computer interaction (HCI), HSV (hue, saturation, value) Colour Model, Web camera, RYB (red, yellow, blue).

1. INTRODUCTION

Human Computer Interaction, mainly deals with natural interfaces. The Graphical User Interface (GUI) on Personal Computers (PCs) is developed, providing an efficient interface for a user to interact with the computer and access the various applications effortlessly.

Today, the technology used for interaction by mobile phone is mainly touch screen. But this technology is still not that cheap to be used in laptops and desktops. The aim is to create a virtual mouse system having a user-friendly interface and an alternative to a touch screen.

Human Computer Interaction is not just limited to keyboard and mouse interaction. The interaction between human beings comes from different sensory modes like gesture, speech, facial and body expressions. Being able to interact with the system naturally is becoming even more important in many fields of Human Computer Interaction.

The idea which the paper proposes uses gesture to operate the cursor. Gestures are a major form of human communication. The primary goal of gesture recognition is to create a system that can identify specific human gesture and use them to correctly identify a corresponding action for controlling the device. A user can control a computer by making a specific gesture in front of a video camera which is linked to the computer.

This paper is based on human computer interaction. To make fewer human efforts to operate their computer and to use hand gestures to communicate with the computer:

- The paper emphasizes to command your computer using natural hands gesture
- No remote controls required
- No wearable devices required
- Only one thing required: YOUR HAND

The aim is to create a virtual mouse system that is a user-friendly device and an alternative to a touch screen. By applying computer vision technology and controlling the mouse by natural hand gestures, one can reduce the workspace required. This paper proposes a novel approach that uses real-time video capturing to control the mouse movement.

2. LITERATURE SURVEY

Many researchers in the robotics and human computer interaction fields have tried to control mouse movement using various devices. However, different technologies have different methods to simulate the mouse movement.

In [1], the wearable unit will consist of buttons on the fingertip for a click, an accelerometer inside the glove for sensing the motion and getting the direction of motion. This would be connected to a microcontroller and a transmitter to communicate wirelessly to the base unit. The accelerometer reads the tilt of each axis and outputs each as an analog voltage. Push buttons are used for enabling and disabling the mouse and for making a left click and right click. RF module is used to provide wireless communication.

In [2], Zhi-hua Chen uses real-time hand gesture recognition using the finger segmentation method. The hand region is extracted from the background with the background subtraction method Then, the palm and fingers are segmented to detect and recognize the fingers. Finally, a rule classifier is applied to predict the labels of hand gestures. Basically, what the position of the fingers is based on that it will only show the pattern of fingers on the screen.

In [3], Chu-Feng Lien used only the fingertips to control the mouse cursor and click. His clicking method was based on image density. It required the user to hold the mouse cursor on the desired spot for a short period of time. It starts with retrieving images from camera drivers. Upon a valid image, it first finds the location of the possible projected screen by Canny Edge detection. After finding the target screen, it calculates the position mapping between device resolution and camera resolution. The projected screen detection is executed after every 10 seconds in case the target screen or the camera is moved accidentally. After the basic environment setup process, the system reads images in a ring buffer and shows the differences between frames based on MHI.

In [4], Kamran Niyazi used a web camera to detect colour tapes for cursor movement. The clicking actions were performed by calculating the distance between two coloured tapes in the fingers. For left click on the very first step, the system records the distance (say D) between the yellow and red tapes in the index finger and the thumb respectively. Here, the index and thumb must be apart as much as possible to get maximum distance. This distance is regarded as the threshold distance for the event. Now, as the thumb moves towards the index finger, the distance between the fingertips or in other words, the distance between yellow and red tapes is decreased. In the second step, when the thumb is closer to the index finger the system records the reduced distance. When the distance between the tapes is reduced to D' or less then consider the event as the left click event of the mouse cursor. The concept of waiting time is used to simulate the right click event of the cursor. If the yellow tape on the index finger is waiting for 7 seconds(say) in front of the camera pointing at the same location, then the event is recognized as the right click event of the mouse. Here, the distance between the red and yellow tapes should be between D and D' respectively. In the same way, the double click event of the cursor can also be made, considering the waiting time.

In [5], A. Erdem uses a fingertip 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 (fingertip) passed over the region.

3. PROBLEM STATEMENT

The hand gesture mouse is based on Human Computer Interaction (HCI) and Image Processing which performs the mouse functions such as cursor movement, left click, right click, scrolling using hand gestures. The hand gesture is captured using a web camera and image processing is done using OpenCV. The system identifies the colored region on the fingertips.

4. PROPOSED SYSTEM OVERVIEW

4.1 SYSTEM OVERVIEW

The aim of human computer interaction is to develop a system that is easy to interact through natural interfaces. Presently, computing environments include wired or wireless devices connected to your computer. Though many users may find this interaction limited or inconvenient to always use a device.

There are different methods to apply other techniques and sensors to enrich user experience. For example, a video camera and computer vision techniques may be used to capture the shape and movements of the hand. The movements of the hand may be analyzed to effortlessly communicate with the system. This approach may lead to a faster, more natural style of interaction to perform certain tasks.

The paper, Mouse Simulation Using Three Colored Tapes is in ubiquitous computing. Here, three colored tapes will be wrapped around the fingers. One of the tapes will be used for controlling cursor movement while the relative distance between the center of specific two-colored tapes will be used for the click events of the mouse. Thus, the system will provide a new experience for users in interacting with the computer.

4.2 Technical Overview:

The platform used for implementation is Anaconda3. Anaconda3 is an open source distribution of Python language. Anaconda Navigator (virtual environment manager) helps in eliminating the need to learn to install each library independently also it gives high-performance computing.

The program uses Image Processing, it is a method to perform some operations on an image, in order to get an enhanced image of the palm or to extract some useful data. It is a type of processing in which image or characteristics/features are extracted associated with that image to perform mouse tasks. The program is implemented in Python. It uses image processing module OpenCV and implements the mouse actions using python specific library PyAutoGUI.

Packages installed for successful compilation of the program are Python interpreter to read and execute the code, OpenCV an open source computer vision is used for capturing real-time video and for the analysis of features of the hand. NumPy for numerical operations and PyAutoGUI is a library function that provides a method for controlling the mouse. Video captured by the webcam is processed and only the three colored tips are extracted. The distance between the centers of a particular color is calculated using the method of moments and depending upon their relative positions mouse actions are performed.

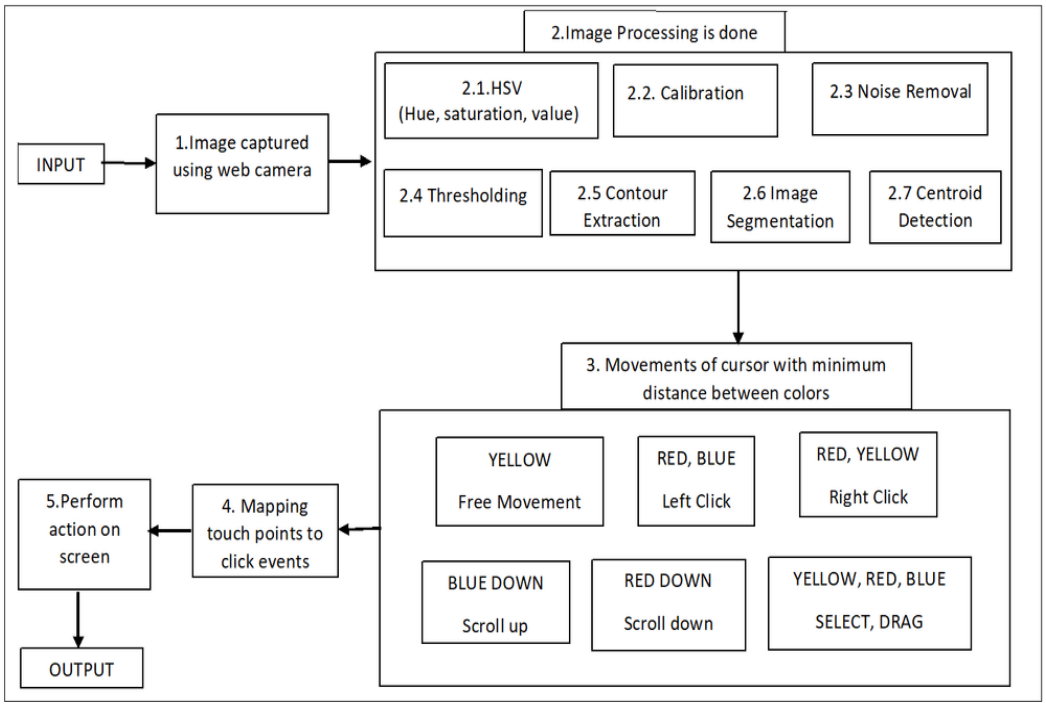

Figure 1: Proposed system architecture

```
# finding the centroid using the method
of 'moments'
M=cv2.moments(contour[0])
if M['m00'] != 0:
    cx=int(M['m10']/M['m00'])
    cy=int(M['m01']/M['m00'])
center = (cx,cy)
```

4.2.1 Capturing the real-time video

The web camera will be used as a sensor. It will be used to capture the real-time video and to detect the hand movements. The web camera captures the movements through a fixed frame size. Users can use an inbuilt camera or can externally connect a camera.

4.2.2 4Hand Recognition and Colour Tape Detection

The first step is to separate the essential hand pixels from the non-hand pixels. This can be done by a background subtraction scheme which segments any potential foreground hand information from the non-changing background. The background image is captured and segmented, only the appropriate background is used for further processing. After background subtraction, the process of skin segmentation is done. Here, a histogram-based skin classifier assigns each of the RYB pixels in the training set to either a 3D skin histogram or a non-skin histogram. The skin segmentation process outputs an image that is ready for the detection of color tapes on the fingers. For this, an algorithm based on HSV color space is used which is very effective to select a certain color out of an image. The idea is to convert the RYB pixels into the HSV color plane so that it is less affected by variations in shades of similar color. Then, a tolerance mask is used over the converted image in the saturation and hue plane. The resulting binary image is then run through a convolution phase to reduce the noise introduced.

4.2.3 Cursor Position

Defining cursor position with yellow. The user might be left or right handed, so a video frame of rectangular sub-portion of smaller size is considered. Due to the noise captured and vibrations of the hand, the centers keep dislocating the mean position. It uses setcursorposition() so that the new center position is set close to the old center. Now the three centers are sent for deciding what action to be performed depending upon their relative position. This is done by the chooseAction() in the code depending upon its output. The performAction() carries out either of the following pyaotogui library function: click, select, drag, scroll.

4.2.4 Controlling mouse

a) Moving mouse cursor

For finding the cursor on the screen and for the cursor moment any specific color (yellow) on a finger is used to perform the movement on the screen.

b) Clicking Action

Clicking action is based on the detection of the colors and center tip of the finger

- If Blue along with Red is detected, then Left Click action is performed.
- If Red along with Green is detected, then Right Click action is performed.
- If Red is detected and the movement of the finger is in the downward direction than Down Scroll action is performed.
- If Red is detected and the movement of the finger is in the upward direction than Up Scroll action is performed.
- If Red, Green, Blue are detected together, then Drag action is performed.

4.3 Pseudo Code

Step 1. Conversion to HSV (Hue, Saturation, Value)

The video captured is converted into HSV format. This helps to detect an object with a certain colour.

```
hsv=cv2.cvtColour(frame,
cv2.COLOUR_BGR2HSV)
```

Step 2. Colour Calibration

Colour calibration is used to measure and/or adjust the colour response of a device. Input for colour range of each colour is in inRange().

```
blue_range=np.array([[88,78,20],
[128,255,255]])
yellow_range=np.array([[21,70,80],
[61,255,255]]
red_range=np.array([[0,95,82], [180,255,255]])
```

Each colour range is calibrated by using cv2.inRange() function.

```
mask=cv2.inRange(hsv_frame, colour_Range[0],
colour_Range[1])
```

Step 3. Noise Removal

The noise removal algorithms reduce or remove the visibility of noise by smoothing the entire image, leaving areas near contrast boundaries. It is done in two step morphism i.e. erosion and dilation. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

```
eroded = cv2.erode(mask, kernel, iterations=1)
dilated = cv2.dilate(eroded, kernel, iterations=1)
```

Step 4. Centroid Detection

The centers of each colour is detected using the method of moments.

```
M = cv2.moments(contour[0])
    if M['m00'] != 0:
        cx = int(M['m10']/M['m00'])
        cy = int(M['m01']/M['m00'])
```

Step 5. Cursor Position

The yellow coloured finger is used for cursor movement. Due to the noise captured while calibrating the colour the centroid vibrates, to reduce the vibrations of the cursor it compares the new position with the previous position of the cursor using the def setCursorPos(). The new cursor position is calculated in such a way that the mean deviation from the desired steady state is reduced.

Step 6. Perform Mouse Movements

Based on the position of the cursor left click, right click, scroll up, scroll down, drag/select are performed.

```
IF distance yellow, red<50 & yellow, blue<50 &
red, blue<50
DO drag
elseif distance red, blue<40
DO left click
elseif distance yellow, red<40
DO right click
elseif distance yellow, red<40, red-blue>120
DO scroll down
elseif distance blue-red>110:
DO scroll up
else setCursorPos ()
```

5. RESULT AND DISCUSSION

5.1 Calibration

Each colour value is adjusted so that centroid of each colour is detected.

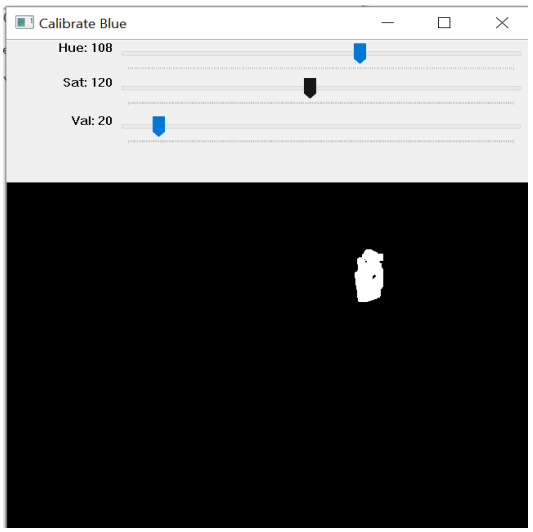

Figure 2: Calibration

5.2 Centroid Detection

Centers of yellow, red, blue are detected.

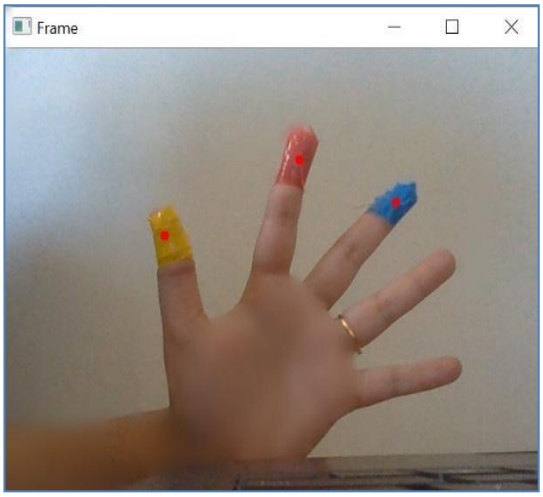

Figure 3: Centroid detection

5.3 Mouse Function

5.3.1 Left click

When the Red and Blue tapes are brought together and cross a certain threshold distance left click action isperformed.

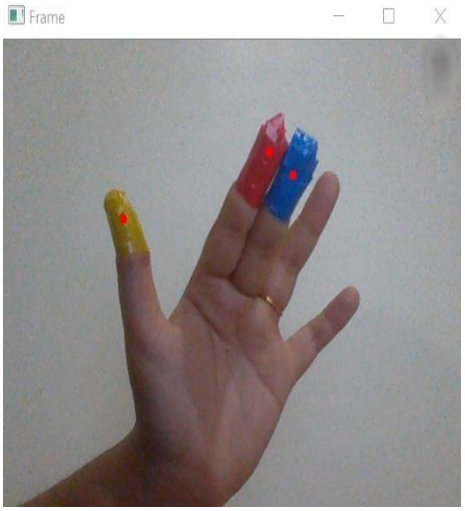

Figure 4: Hand gesture for Left click action

5.3.2 Right click

When the Red and Yellow tapes are brought together and cross a certain threshold distance, right-click action is performed.

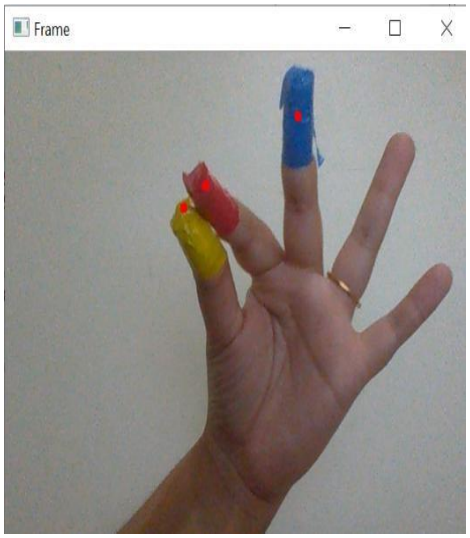

Figure 5: Hand gesture for Right click action

5.3.3 Drag and select

All three tapes should be together for selection and dragging. The distance should be minimized between the centers of the colour.

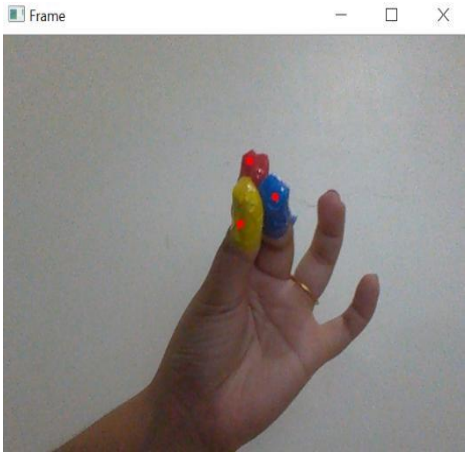

Figure 6: Hand Gesture for Drag and Select

5.3.4 Scrolling up

Move the blue coloured finger in the downward direction to scroll up.

Figure 7: Hand gesture for Scroll up operation

5.3.5 Scrolling down

Move the red coloured finger in the downward direction to scroll down.

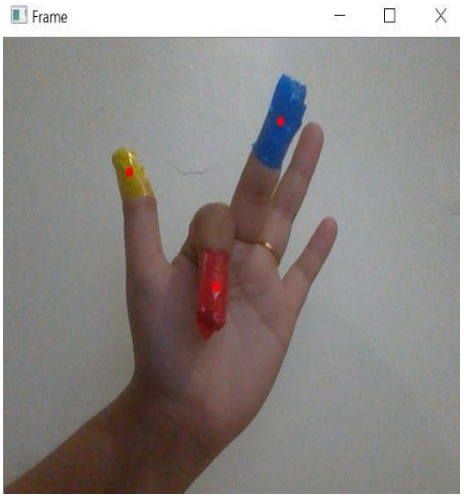

Figure 8: Hand gesture for Scroll down operation

6. CONCLUSION:

The system implemented will simulate mouse movement using hand gestures and a real-time camera. All the mouse tasks such as left clicking, right clicking, and scrolling are implemented. In the future, additional features such as enlarging and shrinking windows, closing the window, etc. will be implemented. This method is created under a standardized operating system. Various application programs can be developed exclusively from this technology to create a wide range of applications with the minimum requirements of resources.

7. FUTURE SCOPE:

Future work can involve the introduction of an infrared sensor, so it can detect hands perfectly when the lighting conditions are not good. In a progressive implementation, this method can be implemented without using coloured tapes. The application can be further extended to gaming. Gestures for page zoom in and zoom out can be implemented. This technology has wide applications in the fields of augmented reality, biomedical instrumentation, computer graphics, prosthetics, and computer gaming.

8. REFERENCES

[1] Chetana S. Ingulkar, A.N. Gaikwad; Hand Data Glove: A wearable real-time device for human computer Interaction; J. International Journal of Science and Engineering, ISSN-2347-2200, Volume1, Number 2; 2013.

[2] Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, and Yu-Bo Yuan; Research Article Real-Time Hand Gesture Recognition Using Finger Segmentation; Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China;25 June 2014.

[3] Chu-Feng Lien; "Portable Vision-Based HCI – A Real-time Hand Mouse System on Handheld Devices"; National Taiwan University, Computer Science and Information Engineering Department.

[4] Kamran Niyazi, Vikram Kumar, Swapnil Mahe, and Swapnil Vyawahare; Mouse Simulation Using Two Coloured Tapes; Department of Computer Engineering, AISSMS COE, University of Pune, India International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2; March 2012.

[5] A. Erdem, E. Yardimci, Y. Atalay, V. Cetin; Computer vision-based mouse, A. E. Acoustics, Speech, and Signal Processing; Proceedings. (ICASS). IEEE International Conference; 2002.

[6] H.S. Grif, C.C. Farcas, „Mouse Cursor Control System Based on Hand Gesture", Procedia Technology, 22 (2016), 657-661.

[7] H.S. Grif, Z. German, A.Gligor, Hand posture mouse. Procedia Technology, 19 (2015), 766–771.

[8] V. John, M. Umetsu, A. Boyali, S. Mita, M. Imanishi, N. Sanma, S. Shibata, Real-time Hand Posture and Gesture-based Touchless Automotive User Interface using Deep Learning. In 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 869-874.

[9] C. Lim, J. Choi, J.I. Park, H. Park, Interactive Augmented Reality System Using Projector-camera System and Smart Phone. In 2015 IEEE International Symposium on Consumer Electronics (ISCE), 2015, pp.1-2.

[10] O. Ozturk, A. Aksac, T. Ozyer, R. Alhajj, Boosting real-time recognition of hand posture and gesture for virtual mouse operations with segmentation. Applied Intelligence. 43(4) (2015) 786–801.

[11] G. Plouffe, A.M. Cretu, Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Warping. IEEE Transactions on Instrumentation and Measurement. 65(2) (2016) 305 – 316.