Virtual Mouse Using Hand Gesture

Mr. Gajendra Moroliya[1], Mr. Sahil Patwekar[2], Prof. S.P. Gopnarayan[3]
[1,2,3]Electronics and telecommunication, AISSMS IOIT, Pune, India

Abstract — *This project presents a new approach for controlling mouse movement using a real-time camera. Major approaches consist of adding more buttons or changing the position of the tracking ball of mouse. Instead, we suggest to change the design of hardware. Our concept is to use a camera and computer vision technology, as image segmentation and gesture recognition, to control mouse tasks (clicking and scrolling) and we show how it can perform everything current mouse devices can. This project shows how to build this mouse control system.*

Keywords- HCI, Background Subtraction, Colour Detection, Gesture controlled mouse, Image Processing.

1. INTRODUCTION

Since the computer technology continues to grow up, the importance of human computer interaction is enormously increasing. Touch screen technology in mobile devices are in trend at present. However, this technology is expensive to be used in desktop systems. Computer vision techniques can be an alternative way for the touch screen and creating a virtual human computer interaction device such as mouse or keyboard using a webcam. In this study, finger tracking based a virtual mouse application has been designed and implemented using a regular webcam. The motivation was to create an object tracking application to interact with the computer and develop a virtual human computer interaction device.

2. COMPONENTS

The components used in this project can't be specific, since this project is a prototype for all computers. As such, certain prerequisites are as follows:

2.1 Hardware

- Webcam

  Webcam is a necessary component for detecting the image. Sensitivity of mouse is directly proportional to resolution of camera. If the resolution of camera is good enough, an enhanced user experience is guaranteed. The webcam serves the purpose of taking real time images whenever the computer starts. On the basis of gestures and motion of fingers, system will decide the respective action.

- Colors Strips

2.2 Software

- MATLAB

  MATLAB version greater than 2013a is used for making this project. Image Acquisition toolbox and Image processing toolbox is necessary for developing the firmware. Along with this, an updated version of Java is required.

- Windows OS

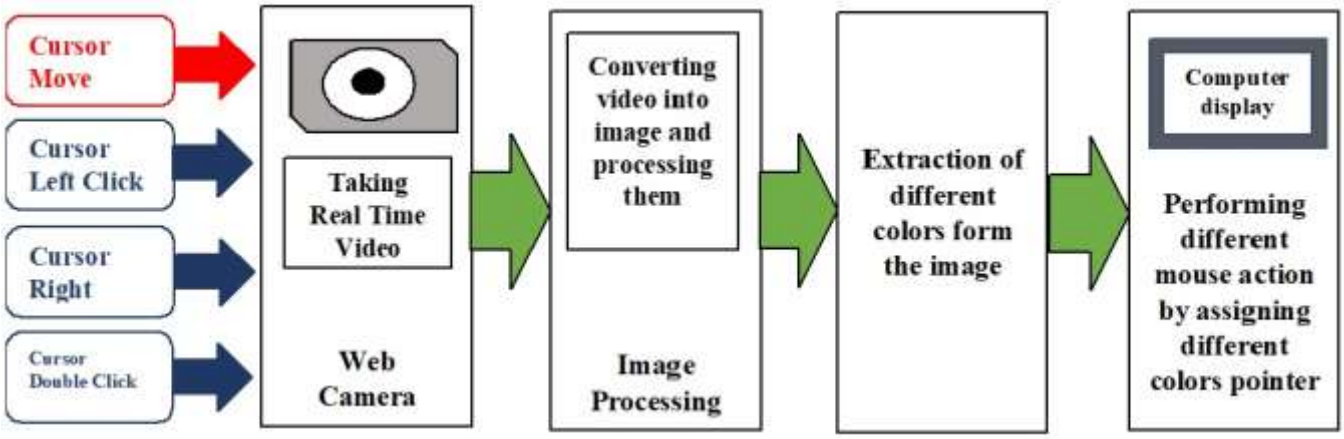

Fig 2. Block Diagram

3. SYSTEM DEVELOPMENT

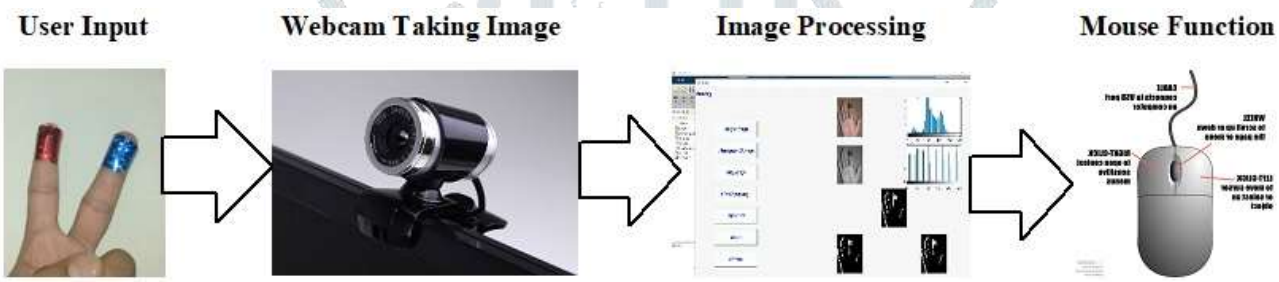

Fig 3. Overview of system

3.1 Capturing the Real Time Video

For the system to work we need a sensor to detect the hand movements of the user. The webcam of the computer is used as a sensor. The webcam captures the real time video at a fixed frame rate and resolution which is determined by the hardware of the camera. The frame rate and resolution can be changed in the system if required. Computer Webcam is used to capture the Real Time Video. Video is divided into Image frames based on the FPS (Frames per second) of the camera

3.2 Flipping of Images

When the camera captures an image, it is inverted. This means the movement of the color pointer is and image is inverse in action. For example, if we move pointer towards the left, the image move towards the right and vice-versa. It is similar to an image obtained when we stand in front of a mirror (right is detected as left and Left is detected as right). To evade these complications, we need to vertically flip the images. The image captured is an RGB image and flipping can't be performed on it directly. So, the distinct color channels of the image are parted, then they are flipped independently. After flipping the red, blue colored channels individually, they are concatenated and a flipped RGB image is obtained.

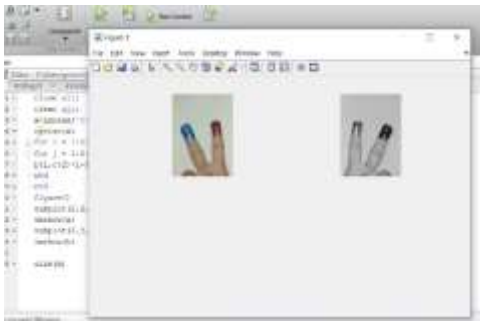

Fig3.2. Flipping of Images

3.3 Conversion of Flipped Image into Grayscale Image

As compared to a colored image, computational difficulty is reduced in a grayscale image. Consequently, the flipped image is transformed into a grayscale image. Rest necessary processes were performed post converting the image into grayscale.

The function can be used as,

y = rgb2gray(x);

Histogram table,

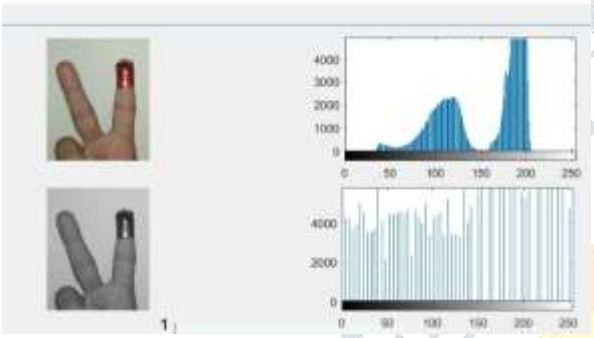

Fig 3.3.1 Histogram of red image

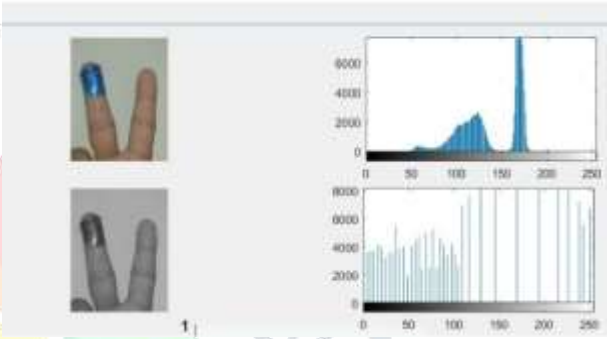

Fig 3.3.2 Histogram of blue image

Threshold value of Red object = 0.27

Threshold value of Blue object = 0.18

3.4 Color Detection

Color Detection is the vital step in the entire process. The red and blue color object is detected by deducting the flipped color suppressed channel from the flipped Grayscale image. Results an image which encompasses the detected object as a patch of grey surrounded by black space.

3.5 Conversion of Gray Scale Image into Binary Scale Image

The grey region of the image obtained after deduction needs to be transformed to a binary image for finding the region of the detected object. A grayscale image consists of a matrix containing the values of each pixel. The gray scale image contains pixel values lay between the ranges 0 to 255, where 1 represents pure white and 0 represents pure black and 255 represents pure white color. We use a threshold value to convert the image to a binary image. This means that all the pixel values lying below threshold value is converted to pure white that is 1 and the rest is converted to black that is Thus the resultant image obtained is a monochromatic image consisting of only black and white colors. Gary image is important conversion to binary is required because MATLAB can only find the properties of a monochromatic image. whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. An inbuilt function in MATLAB is used to find the centroid of the detected region. The output of given function is a matrix consisting of the X (horizontal) and Y (vertical) coordinates of the centroid. These coordinates change with time as the object moves across the screen.

The function can be used as

BW = im2bw(I, level);

Fig 3.5.1 Gray image into Binary image

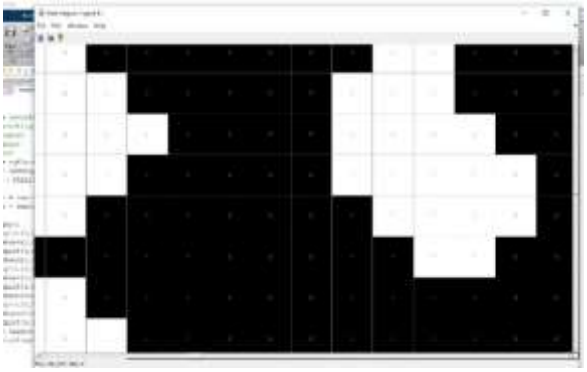

Fig 3.5.1 Binary image (0 and 1)

3.6 Erosion and dilation

Morphological closing operation is then performed on the obtained binary image. The morphological close operation is a dilation followed by erosion. It groups together pixels in close proximity to form a single object. The result is a binary image showing only moved blue objects. The factors of the neighborhood such as shape and size can be decided by the programmer, thereby constructing programmer-defined morphological operations for the input image. The most basic morphological operations are dilation and erosion (see fig 3.6.1 & fig 3.6.2). Dilation added pixels to the objects in an image, while erosion removes pixels on object boundaries. According to the structuring element used, the number of pixels added or removed will differ.

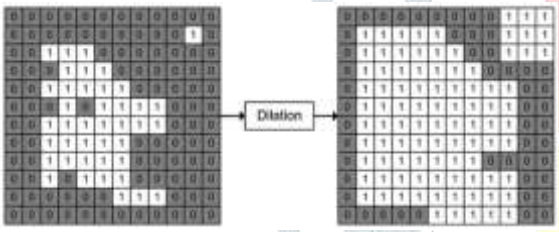

Fig 3.6.1 Dilation

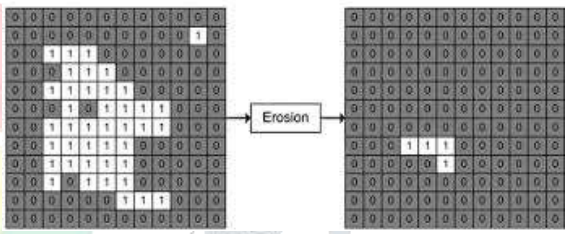

Fig 3.6.2 Erosion

Fig 6.3.3 Perform Erosion and dilation

3.7 Finding Centroid of an object and plotting Bounding Box

To control the mouse pointer it is essential to determine a point whose coordinates can be sent to the cursor. Considering these coordinates, an inbuilt function in MATLAB is used to find the centroid of the detected region then the system can control the cursor movement. The output of function is a matrix consisting of the X (horizontal) and Y (vertical) coordinates of the centroid. suppose objects cross the screen then these coordinates changes with time

3.8 Performing Clicking Actions

The control actions of the mouse are completed by controlling the flags related with the mouse buttons. JAVA is used to access these flags. The user has to perform hand gestures in order to create the control actions. Due to the use of color pointers, the computation time required is reduced. Further more the system becomes resistant to background noise and low illumination conditions. The recognition of blue color trails the same procedure mentioned above. Base of clicking action is real-time detection of two colors.

If Red along with single Blue color is detected, Left clicking action is performed. If Red along with three Blue color is detected, double clicking action is performed.

4. CONCLUSIONS

In this system red object detection and some matlab code to do mouse functionality such as right click, left click double click and right and left move also object detection using a real-time camera. This system is based on computer vision algorithms and can do all mouse tasks. However, it is difficult to get stable results because of the variety of different color and skin colors of human races. Most vision algorithms have illumination issues. From the results, we can expect that if the vision algorithms can work in all environments then our system will work more efficiently. This system could be useful in presentations and to reduce work space.

5. REFERENCES

1. A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E."Computer vision based mouse", Acoustics, Speech, andSignal Processing, Proceedings. (ICASS). *IEEE International Conference*.
2. Hojoon Park, "A Method for Controlling the Mouse Movement using a Real Time Camera", *Brown University*, Providence, RI, USA, Department of computer science.
3. Chang S. L., Chen L. S., Chung Y. C. and Chen S.W, 2004, "Image Recognition", IEEE Transactions on Intelligent Transaction Systems, Vol. 5, No.1, pp. 42-57.
4. The MATLAB website. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/2 8757-tracking-red-color-objects-using-matlab
5. Shashank Prasad, Shubhra Sinha " Real-time Object Detection and Tracking in an Unknown Environment" IEEE
6. Dipali Shahare and Ranjana Shende, "Moving Object Detection with Fixed Camera and Moving Camera for Automated Video Analysis," International Journal of Computer Applications Technology and Research, vol. 3, issue 5, pp. 277-283, 2014