Virtual Mouse Based on Colour Pointer Tracking

Bhavana I V

PG Scholar, Dept. of ECE, Channabasaveshwara Institute of Technology, Gubbi, Tumkur, India

Abstract: ***There are so many devices developed to help the physical world interact with the digital world. We present a novel approach for Human Computer Interaction (HCI) where, we control cursor movement using a real-time camera. Our method is to use a camera and computer vision technology, such as image segmentation and colour tracking, to control mouse tasks (left and right clicking, double-clicking, and scrolling) and we show how it can perform everything as current mouse devices can. A color pointer has been used for the object recognition and tracking. Left and the right click events of the mouse have been achieved by detecting the number of pointers on the images.***

Keywords*:* Hand gesture, Human Computer Interaction, Color Detection, Web camera, Background Subtraction.

1. Introduction

As the present computer vision technology is growing up, the scope for human computer interaction is increasing enormously .Nowadays most of the mobile devices are using a touch screen technology. However, this technology is still not cheap enough to be used in desktop systems. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques can be an alternative way for the touch screen. Finger tracking based on virtual mouse application has been designed and implemented using a regular webcam. The motivation is to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device. The goal of this paper is to provide an easier human-machine interaction routine.

2. Related Work

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement using video devices. However, all of them used different methods to make a clicking event. One approach, by Erdem et al, used 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 passed over the region.

Another approach was developed by Chu-Feng Lien. He used only the finger-tips to control the mouse cursor and click. His 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. Paul et al, used still another method to click. They used 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."

3. Objective

A virtual mouse is software that allows users to give mouse inputs to a system without using an actual mouse. To the extreme it can also be called as hardware because it uses an ordinary web camera. A virtual mouse can usually be operated with multiple input devices, which may include an actual mouse or a computer keyboard. Virtual mouse which uses web camera works with the help of different image processing techniques. In this the hand movements of a user is mapped into mouse inputs. A web camera is set to take images continuously. The user must have a particular color in his hand so that when the web camera takes image it must be visible in the image. This color is detected from the image pixel and the pixel position is mapped into mouse input. Depending upon the size of the image taken by camera various scaling techniques are used because the pixel position in the image will not have a correspondence with screen resolution.

4. Introduction to the System

In our work, we have tried to control mouse cursor movement and click events using a camera based on colour detection technique. Here real time video has been captured using a Web- Camera. The user wears coloured tapes to provide information to the system. Individual frames of the video are separately processed. The processing techniques involve an image subtraction algorithm to detect colours. Once the colours are detected the system performs various operations to track the cursor and performs control actions. No additional hardware is required by the system other than the standard webcam which is provided in every laptop computer.

4.1 Image Processing

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible. Here image processing is in the sense that we are splitting each pixel of the image into RGB components.

4.2 Image Analysis

Image analysis is the extraction of meaningful information from images; mainly from digital images by means of digital image processing techniques. Image analysis tasks can be as simple as reading bar coded tags or as sophisticated as identifying a person from their face. In this project we do various image analysing techniques. The main thing done is the color detection.

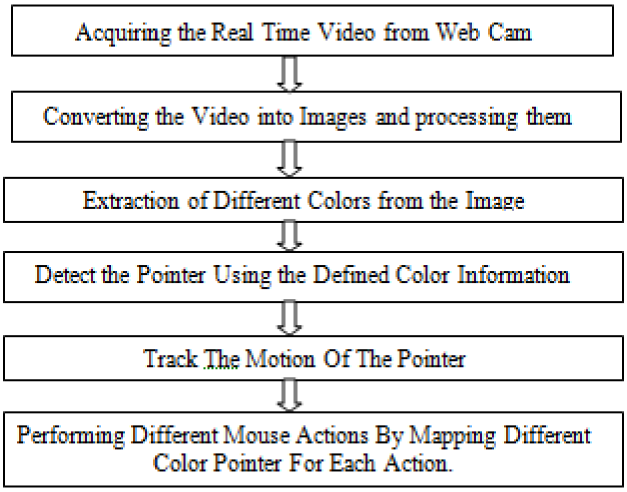
At first we receive an image from the web cam. Then each pixel is retrieved from the image and extracts the red, green and blue values (RGB) from each pixel. Now we can easily detect a particular color since all the colors are combinations of RGB values. Here we just to try to detect only Red Green and Blue colors. This is done by traversing through the image, retrieving each pixel, extracting RGB values and then comparing the color values to the RGB values.

5. System Description

In the object tracking application one of the main problems is object detection. Instead of finger tips, a color pointer has been used to make the object detection easy and fast. A circle blue sticker is used as a color pointer in this study. To simulate the click events of the mouse three fingers with three color pointers has been used.
The basic algorithm is as follows;

- Set a pointer in the image
- Detect the pointer using the defined color information
- Define the region and the center of the pointer and draw a bounding box around it
- Track the motion of the pointer
- Move the cursor according to the position of the center of the pointer
- Simulate the left click and the right click of the mouse

5.1 Basic Block Diagram


Acquiring the Real Time Video from Web Cam
Converting the Video into Images and processing them
Extraction of Different Colors from the Image
Detect the Pointer Using the Defined Color Information
Track The Motion Of The Pointer
Performing Different Mouse Actions By Mapping Different Color Pointer For Each Action.

Flow Chart 1: Basic Flow Chart

5.2 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
- Processing of individual Framesfor region-of-interest detection, retargeting, video and image classification, etc.

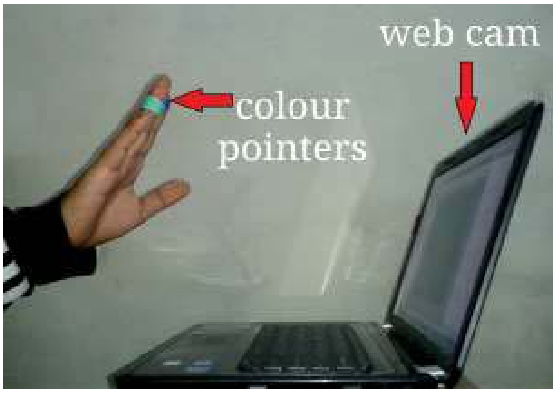


Figure 1: Capturing the Real Time Video

5.3 Flipping of Images

When the camera captures an image, it is inverted. This means that if we move the color pointer towards the left, the image of the pointer moves towards the right and vice-versa. It’s similar to an image obtained when we stand in front of a mirror (Left is detected as right and right is detected as left). To avoid this problem we need to vertically flip the image. The image captured is an RGB image and flipping actions cannot be directly performed on it. So the individual color channels of the image are separated and then they are flipped individually. After flipping the red, blue and green colored channels individually, they are concatenated and a flipped RGB image is obtained.

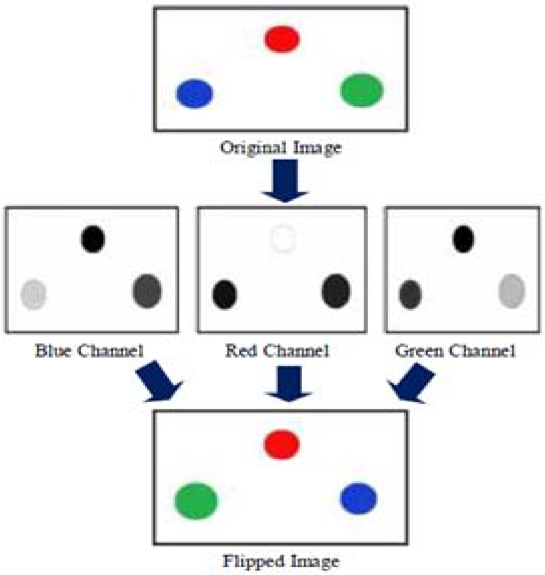


Figure 2: Flipped Image

5.4 Conversion of Flipped Image into Gray scale Image

As compared to a colored image, computational complexity is reduced in a gray scale image. Thus the flipped image is converted into a gray scale image. All the necessary operations were performed after converting the image into gray scale.

5.5 Color Detection

This is the most important step in the whole process. The red, green and blue color object is detected by subtracting the flipped color suppressed channel from the flipped Gray-Scale Image. This creates an image which contains the detected object as a patch of grey surrounded by black space Conversion of gray scale Image into

5.6 Binary scale Image

The grey region of the image obtained after subtraction needs to be converted 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 pixel values lay between the ranges 0 to 255 where 0 represents pure black and 255 represents pure white color. We use a aiming features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region or else non face region.

5.7 Finding Centroid of an object and plotting

For the user to control the mouse pointer it is necessary to determine a point 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 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.

- Centroid of the image is detected
- Its co-ordinates are located and stored in a variable

5.8 Tracking the Mouse pointer

Once the coordinates has been determined, the mouse driver is accessed and the coordinates are sent to the cursor. With these coordinates, the cursor places itself in the required position. It is assumed that the object moves continuously, each time a new centroid is determined and for each frame the cursor obtains a new position, thus creating an effect of tracking. So as the user moves his hands across the field of view of the camera, the mouse moves proportionally across the screen. There is no inbuilt function in MATLAB which can directly access the mouse drivers of the computer. But MATLAB code supports integration with other languages like C, C++, and JAVA. Since java is a machine independent language so it is preferred over the others. A java object is created and it is linked with the mouse drivers. Based on the detection of other colors along with red the system performs the clicking events of the mouse. These color codes can be customized based on the requirements.

5.9 Performing Clicking Actions

The control actions of the mouse are performed by controlling the flags associated 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. Furthermore the system becomes resistant to background noise and low illumination conditions. The detection of green and blue colors follows the same procedure discussed above. Clicking action is based on simultaneous detection of two colors.

- If Red along with Green color is detected, Left clicking action is performed.
- If Red along with Blue color is detected, Right clicking action is performed.

6. Problems and Drawbacks

Since the system is based on image capture through a webcam, it is dependent on illumination to a certain extent. Furthermore the presence of other colored objects in the background might cause the system to give an erroneous response. Although by configuring the threshold values and other parameters of the system this problem can be reduced but still it is advised that the operating background be light and no bright colored objects be present.

The system might run slower on certain computers with low computational capabilities because it involves a lot of complex calculations in a very small amount of time. However a standard pc or laptop has the required computational power for optimum performance of the system. Another fact is that if the resolution of the camera is too high then the system might run slow. However this problem can be solved by reducing the resolution of the image by making changes in the system.

7. Results

The necessary steps to be followed are as follows:

- Firstly, we need to interface the camera using MATLAB.
- Start the camera to capture the images continuously.
- Capturing the real time video using the started up camera.
- Processing the individual image frame obtained from the video.
- Flipping of each image frame.
- Conversion of each frame to a GRAY-SCALE image.
- Color detection and extraction of different colors.
- Conversion of each frame into BINARY-SCALE image.
- Find the center and draw the bounding box.
- Track mouse pointer using coordinates of centroid.
- Different mouse actions are performed based on the steps below.

a. First, we need to detect red color in the image.
b. If red color pointer is detected, we need to track the mouse pointer.
c. If blue color pointer is detected, we need to perform right click.
d. If double blue color pointer is detected, we need to perform double clicking action.

e. If none of the color has been detected again we need to try tracking different colors from the image.

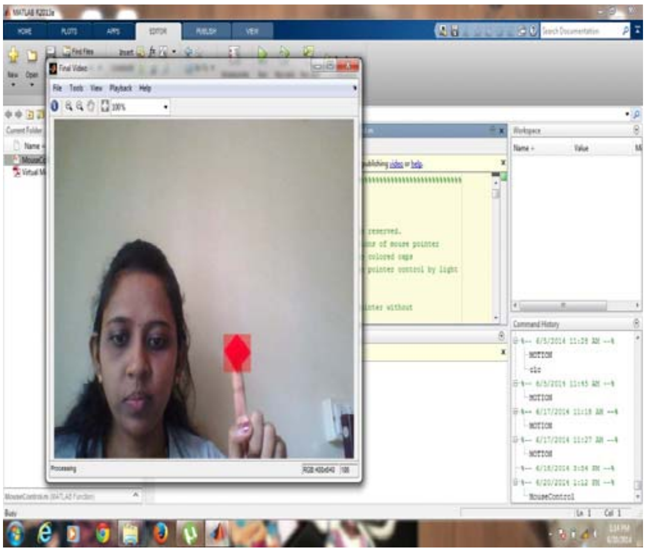


Figure 3: red color is detected and a bounding box in drawn around it. Red pointer movement is mapped with mouse movement

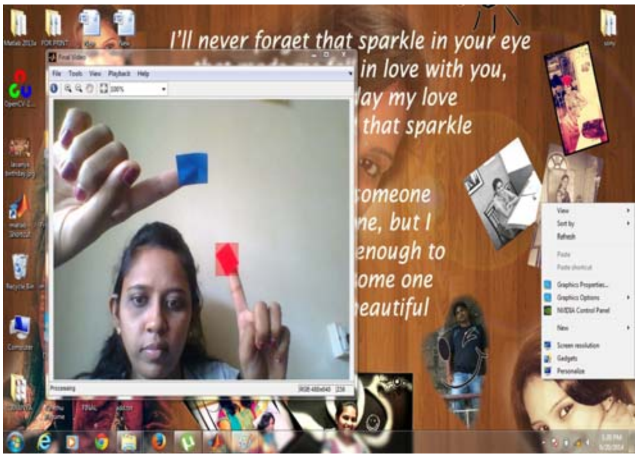


Figure 4: A red along with a blue pointer is detected. Bounding box will be drawn for both the detected color pointers. Red pointer indicates right click which pop up the window as shown in the fig above

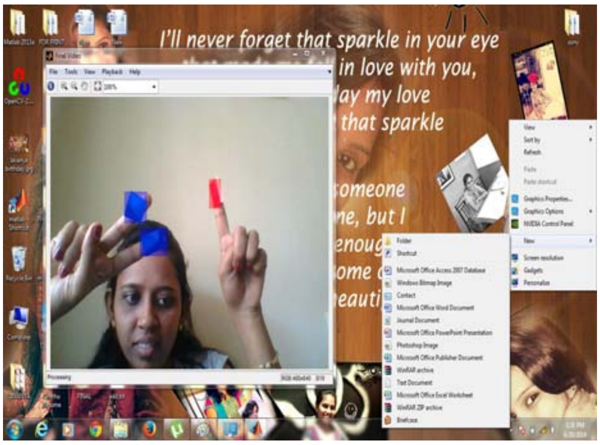


Figure 5: Two blue pointers are detected. Indicating double click which will open perform the double clicking action creating a new folder on the desktop as shown above.

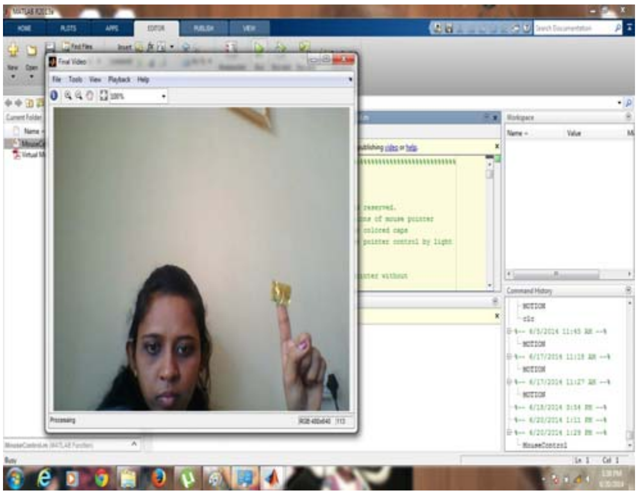


Figure 6: Since we have not defined yellow color, bounding box will be not created. No actions will be performed.

8. Conclusion

In this paper, an object tracking based virtual mouse application has been developed and implemented using a webcam. The system has been implemented in MATLAB environment using MATLAB Image Processing Toolbox. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation.

This technology can be used to help patients who don't have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person's motions are tracked and interpreted as commands. Most of the applications require additional hardware which is often very costly. Our motive was to create this technology in the cheapest possible way.

References

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

[2] 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 .

[3] Hojoon Park, "A Method For Controlling The MouseMovement using a Real Time Camera", 2008, Brown University,Providence ,RI ,USA, Department of computer science http://www.mathworks.com/matlabcentral/fileexchange/28757-tracking-red-color-objects-using-matlab.

[4] http://www.mathworks.com/help/techdoc

[5] http://www.mathworks.com/support/solutions/en/data/1-2X10AT/index.html?solution=1-2X10AT

[6] Chris Solomon and Toby Breckon, "Fundamentals Of Digital Image Processing", Third edition, John Wiley & Sons, Ltd,2011

[7] Kermit Sigmon, "MATLAB Primer", Third Edition, 1990.