

Virtual Mouse Using Colour Detection

Suraj Shetty, Shubham Yadav, Rohan Upadhyay, Prof. Vaihali Bodade

Computer Engineering, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India

ABSTRACT

In this paper we represent an approach for Human-computer interaction. In this interface we have tried to control mouse cursor movements and click events using hand gestures based on color detection techniques acquired using web-camera. This method mainly focuses on increasing human-computer interaction and makes it easy in use in a cost effective manner.

Keywords: Hand gesture, Human Computer Interaction, Color Detection, Web camera, Background Subtraction, frame rate, Flipping of images.

I. INTRODUCTION

Since the computer technology continues to grow up, the importance of human computer interaction is continuously increasing? In current days most of the mobile devices are using touch screen technology. However, touch screen 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 computer vision techniques and sensor technologies can be an alternative way for replacing the touch screen technology.

In our work, we have tried to control mouse cursor movement and click events of mouse using a web camera based on colour detection technique. In this technique real time video has been captured using a Webcam. The user wears colored tapes or capes to provide information and commands to the system. Individual frames of the real time video captured by the camera are separately processed for execution. The processing technique involves an image subtraction algorithm for the detection of colours. Once the colours are detected the system performs many different operations to track the cursor and performs the cursor control actions, the details are provided below. No extra hardware is required for developing this system other than the

standard webcam which is already provided in every laptop computer.

II. METHODS AND MATERIAL

A. System Description

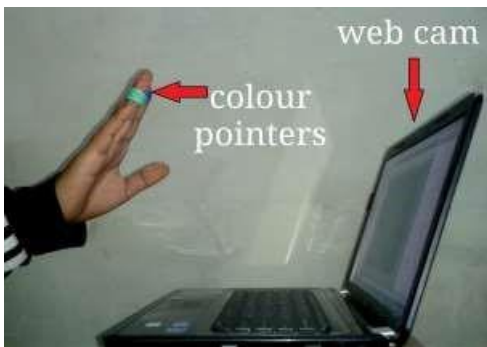
Following are the steps involved in developing our system:

1. Capturing real time video of the actions using web camera.
2. Processing every individual image frame.
3. Flipping of each image frame.
4. Converting each image frame to a grey scale image.
5. Detecting colors and extracting colors (RGB) from flipped grey scale images.
6. Converting the detected image into a binary image.
7. Finding the region of detected image and calculating the centroid of that image.
8. Tracking the pointer of mouse using the coordinates obtained from the calculated centroid of that image.
9. Simulating the left and right click events of the mouse by allocating different color pointers.

B. Capturing Real Time Video using Webcam

For the system to work we need a sensor for the detection of hand movements of the user. Here the webcam is used as sensor. The webcam captures the real time video of the hand movements of user at a fixed frame ratio and resolution which is obtained by the hardware of the camera. The frame ratio and resolution can be changed in the system by changing the HSV value if required. All these operations on captured images are done with the help of a tool mainly used in computer vision technologies called Open CV.

- Computer Webcam is used to capture the Real Time Video of hand movements.
- Video is divided into individual Image frames based on the FPS i.e. Frames per second of the camera.
- Processing the individual image frames.



C. Flipping of Images

When the webcam captures an image, it is inverted. It means if we move our hand with color pointer towards left, the image of the pointer moves towards right and vice-versa. It is similar to an image obtained when we are standing in front of a mirror (Left side is detected as right side and right side is detected as left side). To avoid this type of problem we need to vertically flip the images. The image captured is an RGB (Red, Green, and Blue) image and flipping actions cannot be directly performed on that image. So the individual color channels of the captured image are separated and then flipped actions are performed on that images individually. After performing the flipping of red, green and blue colored channels individually, the results are concatenated and a flipped RGB image is obtained.

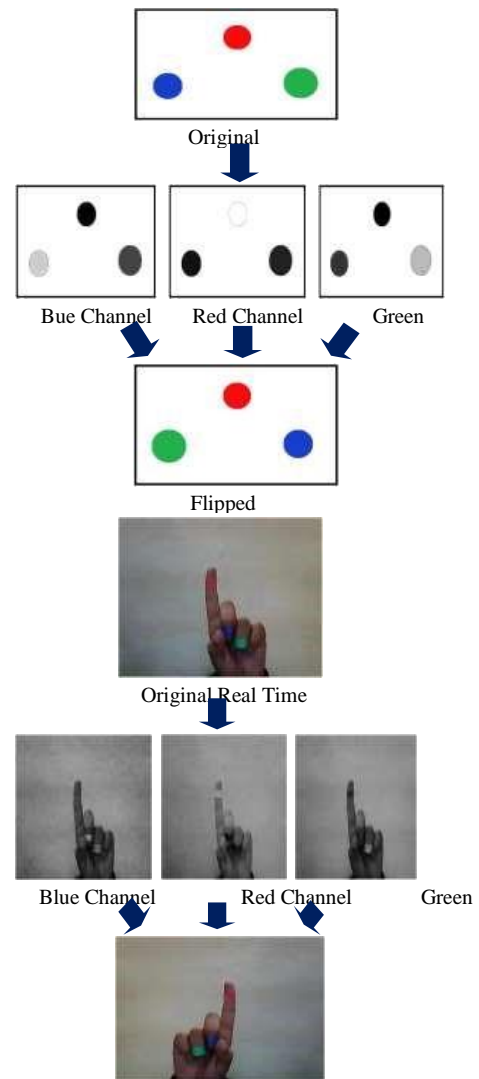


Figure 3 Flipping of a Real time Image

The following images show the entire flipping process in real time. Flipped Real Time Image

D. Converting Flipped Images into Grey-scale Images

Computational complexity of a grey-scale image is less than a coloured image. Thus to reduce the complexity, the flipped image is converted into a grey-scale image. All the necessary operations are performed after converting the flipped image into grey-scale image.

E. Color Detection

This is the most important step in the whole process. The red, green and blue coloured object is detected by subtracting the flipped colour suppressed channel from the flipped Grey-Scale Image which is done using Open

CV. This process produces an image which contains the detected object as a patch of grey surrounded by black space.

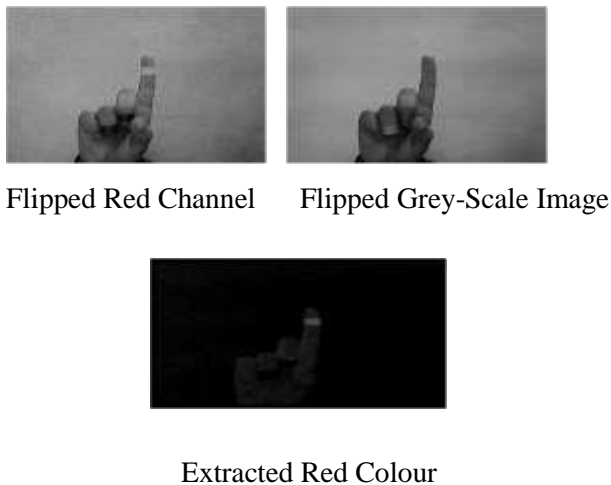


Figure 4. Detection of Red Colour

F. Converting grey-scale Image into Binary scale Image:

The grey region of the captured image is obtained after subtraction which needs to be converted into a binary image for finding the region of the detected object. A grey-scale image is consisting of a matrix containing the values of each pixel. The pixel values lay between the ranges of 0 to 255 where 0 represents pure black color and 255 represents pure white color. We use a threshold value of 20% to convert the grey-scale image to a binary image. It means that all the pixel values lying below 20% of the maximum pixel value are converted into pure black that is 0 and the rest is converted into white that is 1.

Thus the resultant image obtained is a black and white colored image. The conversion to binary is required because Open CV can only find the properties of a black and white (monochromatic) image.

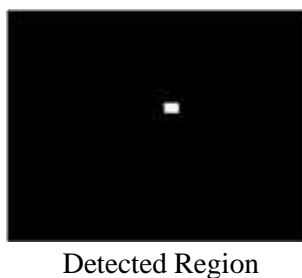


Figure 5. The obtained monochromatic image

G. Finding Centroid of an object and plotting the Centroid:

For the user to control the mouse pointer it is very necessary to define a point whose coordinates can be sent to the cursor. With these obtained coordinates, the cursor movement can be controlled by the system.

An inbuilt function in Open CV is used to find the centroid of the detected region. The output of function is a matrix consisting of the X coordinate (horizontal) and Y coordinate (vertical) of the centroid. These coordinates change with time as the pointer moves across the screen.

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

H. Tracking the Mouse pointer

Once the coordinates has been obtained, the mouse driver is accessed and the coordinates of the centroid are sent to the cursor. With the help of these coordinates, the cursor places itself in the required position. It is assumed that the pointer moves continuously, each time a new centroid is obtained and for each frame the mouse cursor obtains a new position of the pointer, thus creating an effect of tracking. So as the user moves his hands or fingers wearing colour bands across the field of view of the webcam, the mouse moves proportionally across the screen.

Since there is no inbuilt function in Open CV with the help of which it can directly access the mouse drivers of the computer. But Open CV code supports integration with other languages like JAVA. Since java is a platform independent language, it is preferred over the other languages. So a java object is created to access the mouse drivers of the computer.

Based on the detection of colours the system performs the clicking and moving events of the mouse. These colour codes can be modified based on the requirements of the system.



Figure 6. Tracked Red Colour

I. Performing Clicking Actions

The control actions of the mouse are performed by controlling the flags related with the mouse buttons. JAVA language is used to access these flags. The user

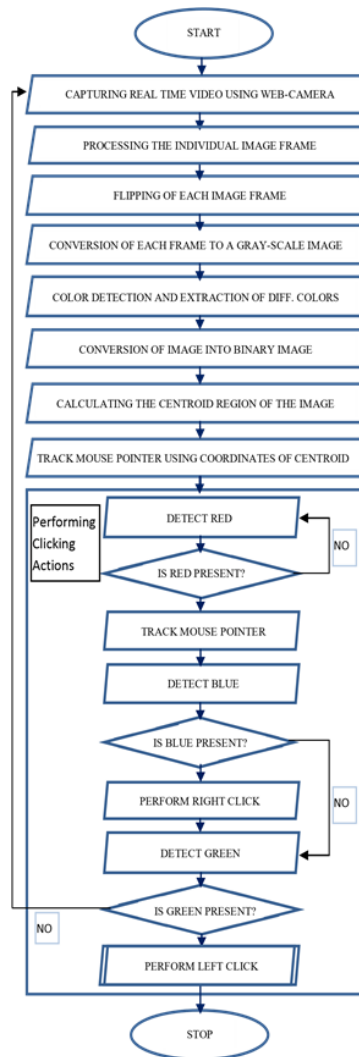
has to perform hand gestures in order to perform the control actions. Due to the use of colour pointers, the computation time required is reduced. The system becomes resistant to background noise and low illumination conditions.

Clicking action is based on simultaneous detection of two colours.

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

III. RESULTS AND DISCUSSION

A. Mouse Control Flowchart:



B. Problems and Drawbacks

Since the system is based on the capturing of image by the webcam, it is dependent on the range of webcam to a certain extent. The presence of other colours in the background may produce an error, which can be reduced by configuring threshold values and other parameters of the system. The system might run slow on certain computers with low computational abilities, because it involves lots of complex calculations in a very small amount of time.

IV. CONCLUSION

A vision based virtual mouse interface to control the mouse cursor using a real-time camera was developed that could perform all mouse tasks such as left and right clicking, double clicking and scrolling. Most vision algorithms have illumination issues and the proposed method is no exemption to it. 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. Thus, we have concluded that by using Virtual Mouse, we can ease the use of a computer system and reduce the cost for a mouse.

V. REFERENCES

- [1] Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E. Computer vision based mouse, IEEE International Conference Acoustics, Speech, and Signal Processing, 2002. Proceedings.
- [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] "Mouse Simulation Using Two Coloured Tapes by Kamran Niyazi, Vikram Kumar, implemented using colour detection technique.
- [4] Hojoon Park, A Method for Controlling the Mouse Movement using a Real Time Camera, 2008, Brown University, Providence, RI, USA Department of computer science.
- [5] "Vision-Based Hand Gesture Recognition by Asanterabi Malima developed a finger counting system to control behaviour of a robot.

- [6] Dix, A. Finlay, J. Abowd, G. D. and Beale, R. Human-Computer Interaction, 3rd ed. Pearson Education Limited, 2004.
- [7] Colombo, C. Bimbo, A. D. and Valli, A. "Visual Capture and Understanding of Hand Pointing Actions in a 3-D Environment," IEEE Transactions on Systems, Man, and Cybernetics, 33(4), University of Florence, Italy, pp.677-686, August2003.

Reference Books

- Mastering opencv with practical computer vision projects
- Learning opencv by Gary Bradski and Adrian Kaehler.

Reference Websites

- <http://www.opencv.org>
- <http://stackoverflow.com>
- <http://www.dreamincode.net>
- <http://msdn.microsoft.com/>
- <http://www.technical-recipes.com>
- <http://www.intorobotics.com>
- <http://www.progcode.in>