International Journal on Applications in Information and Communication Engineering
Volume 1: Issue 5: May 2015, pp 5-7. www.aetsjournal.com ISSN (Online) : 2394-6237

Virtual Mouse based on Color Tracking of Face Makeup

Abishek Easter Raj.A, Anto Daniel.S

Abstract— In this project we present an approach for Human computer Interaction (HCI). where we have tried to control the mouse cursor movement and click events of the mouse using face makeup. Face makeup were acquired using a camera based on color detection technique. This method mainly focuses on the use of a Web Camera to develop a virtual human computer interaction device in a cost effective manner. 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 must do makeup their face to provide information to the system. Lipstick used to control mouse cursor. Eye shadow is used for the click function

Keywords--Machine vision; Color tracking; Virtual Mouse ; Human-Computer Interaction.

I. INTRODUCTION

Human Computer Interaction today greatly emphasizes on developing more spontaneous and natural interfaces. The Graphical User Interface (GUI) on Personal Computers (PCs) is quiet developed, well defined and provides an efficient interface for a user to interact with the computer and access the various applications effortlessly with the help of mice, track pad, etc. In the present day scenario most of the mobile phones are using touch screen technology to interact with the user. But this technology is still not cheap to be used in desktops and laptops. Our objective was to create a virtual mouse system using Web camera to interact with the computer in a more user friendly manner.

II. RELATED WORK

Our project was inspired by a paper of LI WENSHENG where he used Computer vision technology and Web camera to control mouse movements. He used finger-tips to control the mouse cursor and the angle between the thumb and index finger was used to perform clicking actions. He used the fingertip color to track the mouse pointer.

III. INTRODUCTION TO THE SYSTEM

In our work, we have tried to control mouse cursor movement and click events using a camera based on color detection technique. Here real time video has been captured using a Web-Camera. The user must do makeup their face to provide information to the system. Lipstick used to control mouse cursor. Eye shadow is used for the click function.

IV. SYSTEM DESCRIPTION

Following are the steps in our approach

(i) Capturing real time video using Web-Camera. (ii) Processing the individual image frame. (iii) Flipping of each image frame. (iv) Conversion of each frame to a grey scale image. (v) Color detection and extraction of the different colors (RGB) from flipped gray scale image (vi) Conversion of the detected image into a binary image. (vii) Finding the region of the image and calculating its centroid. (viii) Tracking the mouse pointer using the coordinates obtained from the centroid. (ix) Simulating the left click and the right click events of the mouse by assigning different color pointers.

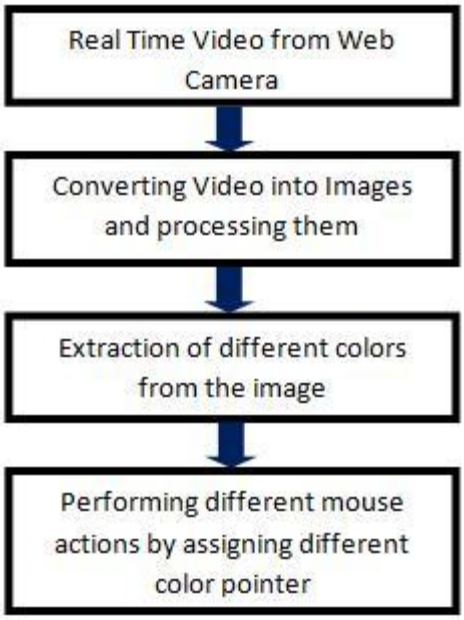


Fig 1: Basic Block Diagram of the System

A. Capturing the real time video:

1.Computer Webcam is used to capture the Real TimeVideo, 2.Video is divided into Image frames based on the FPS (Frames per second) of the camera, 3. Processing of individual Frames.

B. 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.

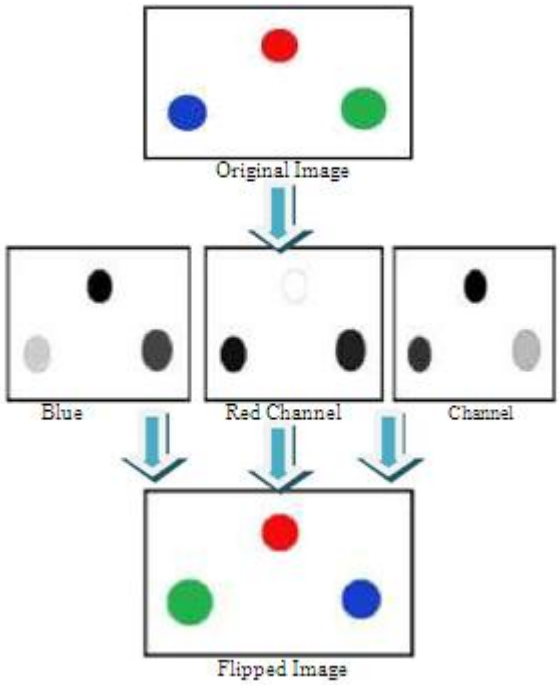


Fig 2: Flipped image

Fig 1: Flipped red and Gray-scale image

C. 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.

D. 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.

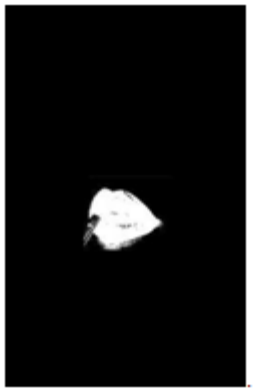


Fig 4: Detection Region

E. Conversion of gray scale Image into 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 gray scale 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 threshold value of 20% to convert the image to a binary image. This means that all the pixel values lying below 20% of the maximum pixel value is converted to pure black that is 0 and the rest is converted to white that is 1. Thus the resultant image obtained is a monochromatic image consisting of only black and white colors. The conversion to binary is required because MATLAB can only find the properties of a monochromatic image.

F. Finding Centroid of an object and plotting:

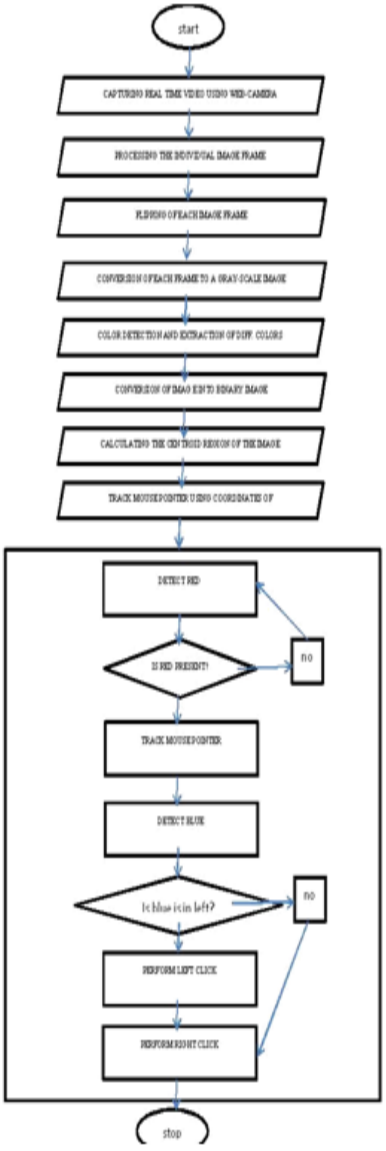
For the user to control the mouse pointer it is necessary to determine a point whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. An inbuilt function in MATLAB is used to find the centroid of the detected region. The output of function is 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

G. 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



V. 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

VI. 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

REFERENCE

[1] A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E. "Computer vision based mouse", Acoustics, Speech, and Signal Processing, Proceedings. (ICASS). *IEEE International Conference*, 2002
[2] Hojoon Park, "A Method for Controlling the Mouse Movement using a Real Time Camera", *Brown University*, Providence, RI, USA, Department of computer science, 2008
[3] Chu-Feng Lien, "Portable Vision-Based HCI – A Realtime HandMouse System on Handheld Devices", *National Taiwan University*, Computer Science and Information Engineering Department.
[4] Kamran Niyazi, Vikram Kumar, Swapnil Mahe, Swapnil Vyawahare,"Mouse Simulation Using Two Coloured Tapes", Department of Computer Science, University of Pune, India, *International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2*, March 2012
[5] K N. Shah, K R. Rathod and S. J. Agravat, "A survey on Human Computer Interaction Mechanism Using Finger Tracking" *International Journal of Computer Trends and Technology, 7(3)*, 2014, 174-177
[6] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, 07458
[7] Shahzad Malik, "Real-time Hand Tracking and FingerTracking for Interaction", CSC2503F Project Report, December 18, 2003
[8] The MATLAB website.