International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Human Computer Interaction Using Hand Gestures To Control Mouse Events

C. Jahnavi 11189A045, K. Nirisha 11189A110

ABSTRACT

A new approach for controlling mouse movement using a real-time camera. Most existing approaches have been involved to change mouse parts such as adding more buttons or changed the position of the tracking ball. Instead, we propose to change the hardware design. The proposed method overcomes the current shortcomings such as usage of hardware mouse, various movements lead to pains in the hands,is based on a visualization effect of movement followed by computer vision technology, such as image segmentation and gesture recognition, to control mouse tasks (left and right clicking, double-clicking, and scrolling) and we can observe how it will perform current mouse devices.Hand gestures are acquired using a camera based on the color detection technique. It is basically focusing on the use of a Web Camera to develop a virtual human-computer interaction device in a cost-effective manner. Keeping all the essential aspects in mind, we have created a framework which recognizes the movement of fingers and various patterns formed by them. In addition, color caps have been used on fingers to distinguish it from the background colors. Thus, we have been performed the gestures recognition for various mouse events. In the experimental work, we have used the OpenCV library in python.

Keywords:Human-Computer Interaction, Colour Detection, Web camera, Gesture Recognition, Image Processing

1. Introduction

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 a Web camera to interact with the computer in a more user-friendly manner that can be an alternative approach for the touch screen. This activity on the gesture-based user interface (UI) has been proliferating in the last decade. The main reason for these technologies becomes more popular is because it can be applied to many different fields easily and efficiently. Human-Computer Interaction today greatly emphasizes on developing more spontaneous and natural interfaces. The Graphical User Interface (GUI) on Personal Computers (PCs) is quite 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, trackpad, etc.

.

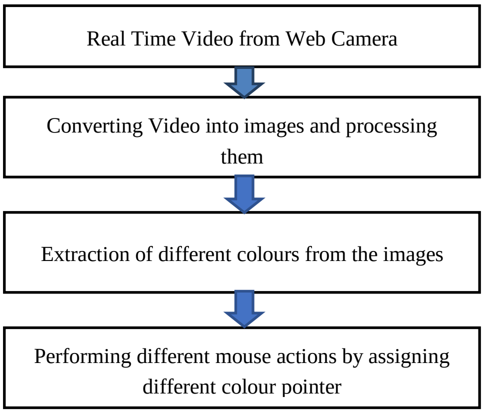
Real Time Video from Web Camera
Converting Video into images and processing them
Extraction of different colours from the images
Performing different mouse actions by assigning different colour pointer

Figure 1. Block diagram of proposed method

2. Literature Survey

Computer vision based mouse is to control the mouse tasks and recent challenges, erden et al. [1] have been investigated a camera and computer vision based technologies, such as image segmentation and gesture recognition. To overcome the limitations of erden et al.[3]this project, we have taken inspirational from the Hojoon Park which is inspired by computer vision-based technology that has the capability to control mouse movements in web camera. However, he used finger-tips to control the mouse cursor and the angle between the thumb and index finger was used to perform clicking actions. Chu-Feng Lien had used an intuitive method to detect hand motion by its Motion History Images (MHI). In this approach, the only fingertip was used to control both the cursor and mouse click. In his approach, the user needs to hold the mouse cursor on the desired spot for a specific period of time for clicking operation [2]. Kamran Niyazi et al used Web camera to detect color tapes for cursor movement. The clicking actions were performed by calculating the distance between two colored tapes in the fingers. In K N Shah et al. [9] have represented some of the innovative methods of the finger tracking used to interact with a computer system using computer vision. They have divided the approaches used in Human-Computer Interaction (HCI) into two categories.

- HCI without using interface.
- Moreover, they have mentioned some useful applications using finger tracking through computer vision.

In this study described the motivation and the design considerations of an economical head-operated computer mouse. In addition, it focuses on the invention of a head-operated computer mouse that employs two tilt sensors placed in the headset to determine the head position and to function as a simple head-operated computer mouse. One tilt sensor detects the lateral head-motion to drive the left/right displacement of the mouse. The other one detects the head’s vertical motion to move up and down with respect to the displacement of the mouse. A touch switch device was designed to contact gently with operator’s cheek. The operator may puff his cheek to trigger the device to perform a single click, double clicks, and drag commands. This system was invented to assist people with disabilities to live an independent professional life.

3. Problem Statement

Computer vision-based mouse can easily be applied to the web services, smart home systems, robot manipulation, and games. That is why tracking non-rigid motions from sequential videos have been a great interest to the computer vision community. We grew up interacting with the physical objects around us. How we manipulate these objects in our lives every day, we use gestures not only to interact with objects but to interact with each other and this brings us a step closer to Human-object relationship by using gesture recognition technique. In this research still webcam has been used to recognize the gestures. There is no need for 3D or stereo cameras and above research has also been tested on low-cost 1.3-megapixel laptop webcam.

The work is dedicated to a computer vision-based mouse that acts as an interface between the user and various computing devices in the dynamic environment. This paper presents the technique to perform numerous mouse operations thus obviating the need for hardware used for interaction between the user and the computing device. The same approach can be applied to endless tasks such as browsing images, playing games, changing T.V channels, etc. There is a threshold value for distance (in meters) between the user and camera which can further be varied according to the camera's resolution. It means if the subject who wants to be recognized with his hand gestures in some environment, the subject has to come close to certain fixed distance to the camera. This research was done on 1.3-megapixel webcam with a threshold value of 2m. The main elements of this technology are as follows:

- Object tracking using Canny edge Detection Algorithm

4. Architecture

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

Processing of individual Frames

2 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.

3.Conversion of Flipped Image into Grayscale Image

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

4Color 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.

5 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 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 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 openCV can only find the properties of a monochromatic image.

6 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 OPENCV is used to find the centroid of the detected region. The output of the 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 coordinates are located and stored in a variable

7 Tracking the Mouse Pointer

Once the coordinates have 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 OPENCV which can directly access the mouse drivers of the computer. But OPENCV 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.

8 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 the simultaneous detection of two colors.

- If green color is detected, Left clicking action is performed,If yellow color is detected, Moves the cursor along the color detected.

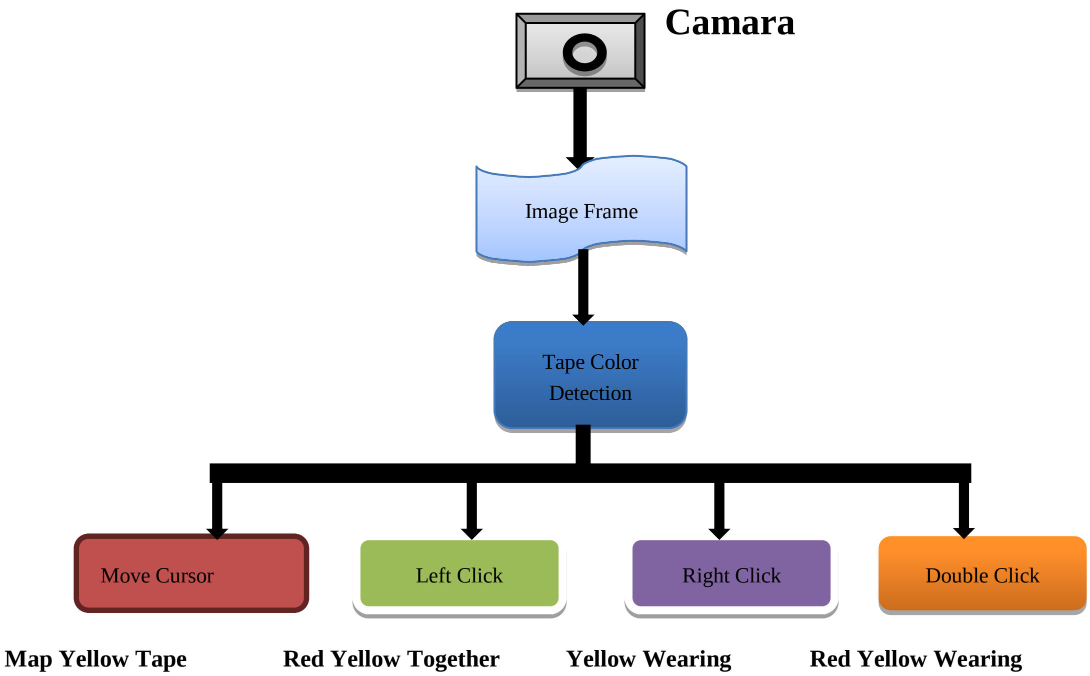

Figure 2. Architecture

6. Implementation

The objective of this system is to 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 hand gestures. Hand gestures were acquired using a camera based on the 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 the color detection technique. Here real time video has been captured using a Web Camera. The user wears colored 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 colors. Once the colors are detected the system performs various operations to track the cursor and performs control actions, the details of which are provided below. No additional hardware is required by the system other than the standard webcam which is provided in every laptop computer.

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. Colour detection and extraction of the different colors (RGB) from flipped grayscale 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.

Result:

We tested all mouse tasks such that left click, right click, double-click, dragging, and scrolling on windows. The tested system is that Core2-Duo T8300, 2GB memory, GeForce 8300, Microsoft Windows XP, Microsoft LifeCam VX-1000 (640x480 resolution, 15fps). We could not compare with the mouse device because this hand gesture system always shows lower performance than a real mouse device. Instead of comparing with a real mouse, weallowed using this system to four testers to know how it can be adapted easily.

We designed four experiments to get performance. In the first experiment, we placed an icon on the center of the desktop window and put the cursor in the top-left corner. We then measured the time in how long it takes to select the icon. In the second experiment, the position of the icons is the same and we only measured the time to show the drop-down menu on the icon. In the third experiment, the position of the icons is the same as the second experiment and we measured the time to open the icon. In the final experiment, we opened a web site (http://news.yahoo.com) on the center of the desktop window and measured the time until the scroll bar moves from top to bottom. The results are shown below: (time = sec). The results obtained by the experiments for various mouse events such as cursor movement, left click, right click, double click, drag, etc has been shown in the below graph. The distance between the user and camera was kept fixed as 1m for the below experiment

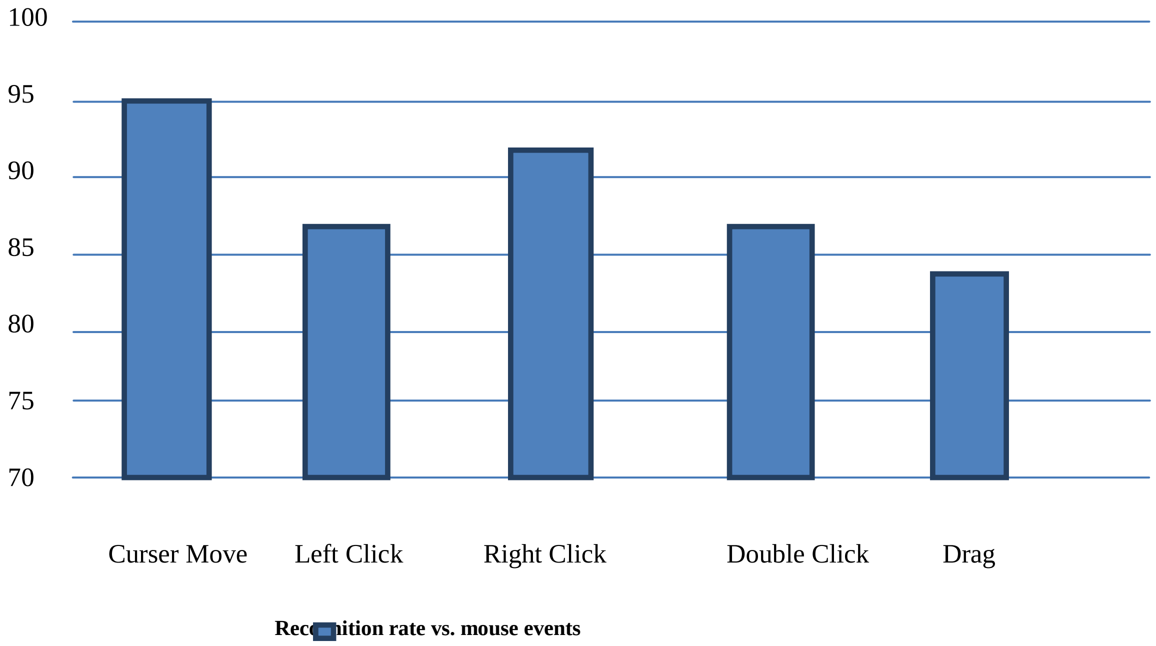

Fig.7. Recognition rate vs. mouse events

The higher the percentage of recognition rate the more likely the gesture is going to be recognized e.g. cursor move may fail 5% of time i.e. there may be a delay in moving the cursor to its new position.

Below graph shows the response rate of cursor movement when the distance between user and camera is varied from 0.3m till 2.4m. The smoothness in cursor movement decreases when the distance from the webcam is increased. For both the experiments 1.3-megapixel laptop webcam was used.

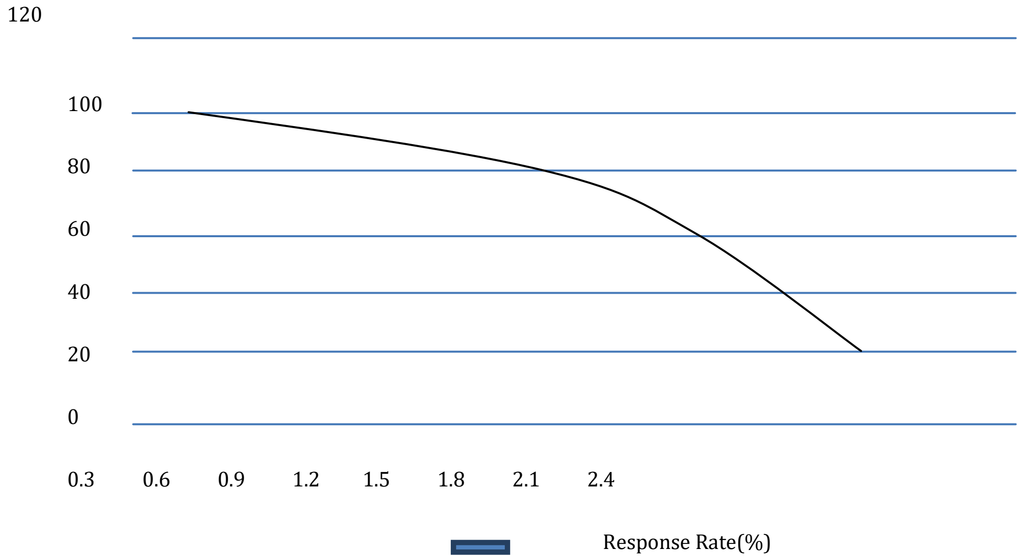

Fig .8. Response Rate.

Four users tested clicking, double-clicking, and scrolling. Measuring the time until finishing a given task, we estimated the system performance. Every tester was new in this system. The clicking task showed similar times with all testers. However, the scrolling result showed the time as unstable. The reason is that the focus of the camera lens works automatically so when the index finger went to the border on the screen then part of the hand colors becomes dim. Eventually, hand segmentation failed because colors could not be on the color range which is in the segmentation part. Thus, we could not get a clear shape of the hand from the video streaming.

7. 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 python environment using OpenCV Libraries. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Furthermore a similar technology can be applied to create applications like a digital canvas which is gaining popularity among artists. 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.

Motion detection and Skin detection are extensively used to enable cursor movement using color caps.

The movement of mouse cursor is achieved with a good precision. However, it is very difficult to get stable results because of the variety of lighting in the background. The performance of the system can be improved by further by using a powerful camera. Most of the applications require additional hardware which is often very costly. Our motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system. Various application programs can be written exclusively for this technology to create a wide range of applications with the minimum requirement of resources.

REFERENCES

A. Banerjee, A. Ghosh, K. Bharadwaj, H. Saikia, Mouse Control using a Web Camera based on Colour Detection, Int. J. Comput. Trends Technol. 9 (2014) 15–20. doi:10.14445/22312803/ijctt-v9p104.

Y. Chen, E.R. Hoffmann, R.S. Goonetilleke, Structure of Hand/Mouse Movements, IEEE Trans. Human-Machine Syst. 45 (2015) 790–798. doi:10.1109/THMS.2015.2430872.

K. Arai, R. Mardiyanto, Camera as Mouse and Keyboard for Handicap Person with Troubleshooting Ability, Recovery, and Complete Mouse Events, Int. J. Hum. Comput. Interact. 1 (2010) 46–56.

Y. Sato, Y. Kobayashi, H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In Proceedings of IEEE, International Conference on Automatic Face and Gesture Recognition (FG), 2000. pp. 462-467.

Y. L. Chen, T. S. Kuo, W. H. Chang, and J. S. Lai, “A novel position sensors-controlled computer mouse interface for the disabled,” in Proc. IEEE Engineering Medicine Biology, Chicago, IL, July 2000, pp. 3945–3948.