iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XII **Month of publication:** December 2021

DOI: https://doi.org/10.22214/ijraset.2021.38160

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

Virtual Mouse using OpenCV

Mr. Devanshu Singh[1], Mr. Ayush Singh[2], Mr. Aniket Kumar Singh[3], Mr. Shailesh Chaudhary[4], Asst. Prof. JayvirSinh Kher[5]
[1, 2, 3, 4, 5]*Computer Science and Engineering, Parul University, Vadodara, Gujarat, India*

Abstract: This research introduces a novel method for controlling mouse movement with a real-time camera. Adding more buttons or repositioning the mouse's tracking ball are two common ways. Instead, we recommend that the hardware be redesigned. Our idea is to employ a camera and computer vision technologies to manage mouse tasks (clicking and scrolling), and we demonstrate how it can do all that existing mouse devices can. This project demonstrates how to construct a mouse control system.

I. INTRODUCTION

As computer technology advances, the importance of human-computer interaction grows exponentially. Touch screen technology in mobile devices is currently popular. However, this technology is prohibitively expensive for use in desktop systems. Using a webcam, computer vision techniques can be used as an alternative to a touch screen and to create a virtual human-computer interaction device such as a mouse or keyboard. In this study, a virtual mouse application based on finger tracking was designed and implemented using a standard webcam. The goal was to develop a virtual human-computer interaction device and an object tracking application to interact with the computer.

II. 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:

A. Hardware

1) PC/Laptop
2) Webcam

For the image to be detected, a webcam is required. The sensitivity of the mouse is proportional to the camera's resolution. A better user experience is ensured if the camera's resolution is high enough. The camera is used to capture real-time photos whenever the computer is turned on. The system will choose the appropriate action based on gestures and finger motions.

B. Software

1) Windows OS
2) OpenCV

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

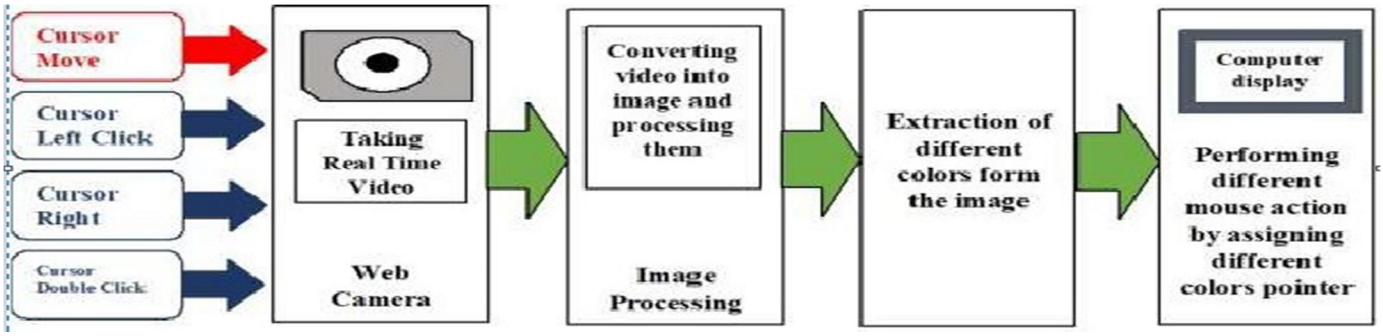


Fig: Block Diagram

©IJRASET: All Rights are Reserved

III. SYSTEM DEVELOPMENT

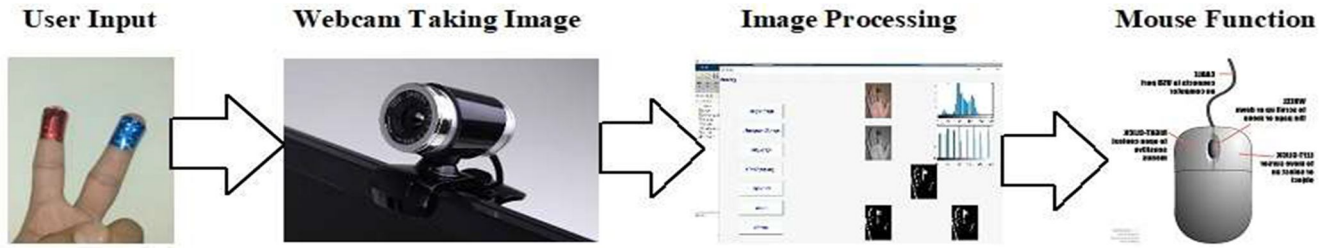


Fig: Overview of system

A. Capturing the Real-Time Video

We'll need a sensor to detect the user's hand movements for the system to work. As a sensor, the computer's webcam is employed. The webcam records real-time video at a defined frame rate and resolution determined by the camera's hardware. If necessary, the system's frame rate and resolution can be modified. The Real-Time Video is captured using a computer webcam. Based on the camera's FPS (frames per second), video is divided into image frames.

B. Flipping of Images

The image is reversed when it is captured by the camera. This means that the color pointer moves in the opposite direction of the image. If we move the pointer to the left, the image will go to the right, and vice versa. It looks like the image we get when we stand in front of a mirror (right is detected as left and Left is detected as right). To avoid these issues, we must vertically flip the photographs. Because the image acquired is an RGB image, it cannot be directly flipped. As a result, the image's distinct color channels are separated and then flipped independently. After independently flipping the red and blue colored channels, they are concatenated and flipped together.

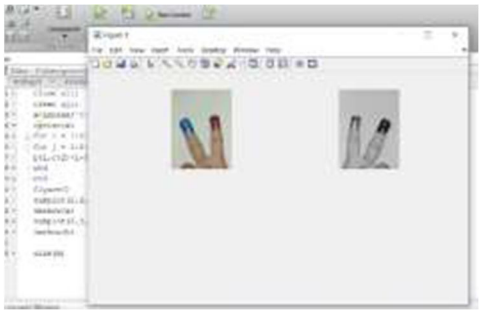


Fig: Flipping of images

C. Conversion of Flipped Image into Grayscale Image

The computational difficulty of a grayscale image is lower than that of a colored image. As a result, the flipped image is converted to a grayscale version. After converting the image to grayscale, the remaining steps were completed.

y = rgb2gray(x); is an example of how to use the function.

Histogram Table,

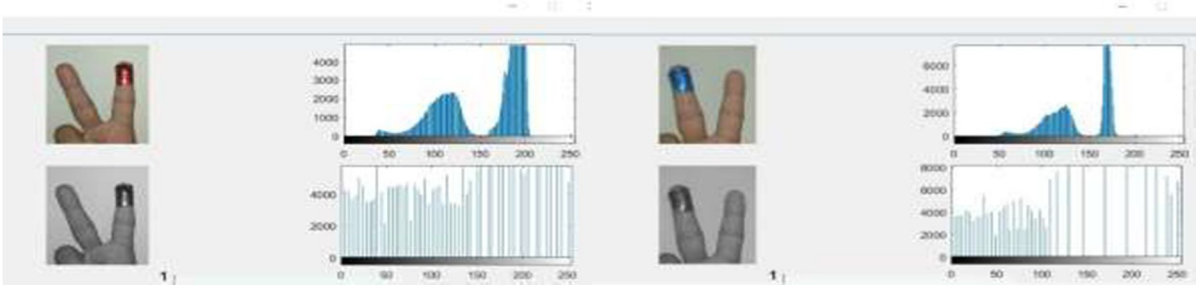


Fig 3.3.1 Histogram of red image
Threshold value of Red object = 0.27

Fig 3.3.2 Histogram of blue image
Threshold value of Blue object = 0.18

©IJRASET: All Rights are Reserved

D. Color Detection

The crucial phase in the entire procedure is color detection. By subtracting the flipped color suppressed channel from the flipped Grayscale image, the red and blue color object is discovered. The detected object appears as a patch of grey surrounded by black space in the photograph.

E. Conversion of Gray Scale Image into Binary Scale Image

To determine the region of the identified object, the grey region of the image acquired after deduction must be converted to a binary image. Each pixel's value is represented by a matrix in a grayscale image. Pixel values in the greyscale image range from 0 to 255, with 1 representing pure white, 0 representing pure black, and 255 representing pure white color. To transform the image into a binary image, we use a threshold value. This indicates that all pixel values below the threshold value are transformed to pure white (number 1), while the remainder is turned to black (number 2). As a result, the final image is monochromatic, consisting solely of black and white colors. The grey image is crucial. Because MATLAB can only identify the attributes of a monochrome image, conversion to binary is required. whose coordinates can be supplied to the cursor The system can control cursor movement using these coordinates. The centroid of the observed region is determined using a built-in MATLAB function. The result of the given function is a matrix containing the centroid's X (horizontal) and Y (vertical) coordinates. As the object moves across the screen, these coordinates vary over time.



Fig 3.5.1 Gray image into Binary image

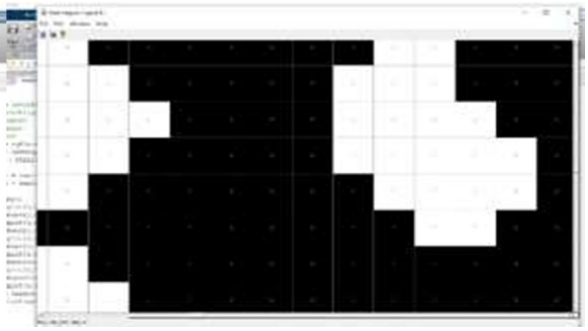


Fig 3.5.1 Binary image (0 and 1)

F. Erosion and Dilation

The resultant binary image is then subjected to a morphological closure process. A dilatation followed by erosion is used to close the morphological gap. It combines pixels that are near together to form a single entity. The output is a binary image with only blue objects that have migrated. The shape and size of the neighborhood can be determined by the programmer, resulting in programmer-defined morphological operations for the input image. Dilation and erosion are the most basic morphological activities (see fig 3.6.1 & fig 3.6.2). Dilation increases the number of pixels in an image's objects, whereas erosion reduces the number of pixels on object boundaries. The numeric value of the number of pixels added or removed will differ.

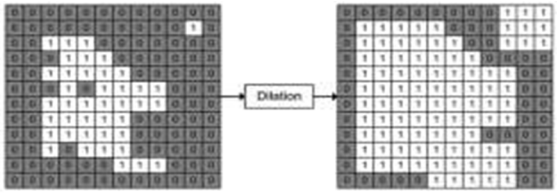


Fig 3.6.1 Dilation

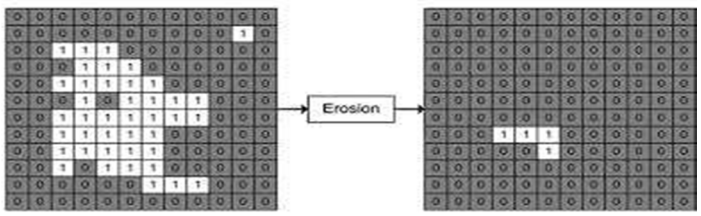


Fig 3.6.2 Erosion

©IJRASET: All Rights are Reserved



Fig 6.3.3 Perform Erosion and dilation

G. Finding Centroid of an object and plotting Bounding Box

It is necessary to determine a point whose coordinates can be communicated to the cursor to control the mouse pointer.

Using these coordinates, a built-in MATLAB function is utilized to compute the centroid of the detected region, after which the system can control cursor movement. The function returns a matrix containing the centroid's X (horizontal) and Y (vertical) coordinates. If objects cross the screen, the coordinates will vary over time.

Controlling the flags associated with the mouse buttons completes the mouse control activities.

These flags are accessed via JAVA. To create control actions, the user must use hand gestures. The computation time is lowered due to the use of color pointers. The device gets more resistant to background noise and low lighting situations as time goes on. The process for recognizing blue colors is the same as the one described previously. The real-time detection of two colors is the foundation of the clicking action.

Left-clicking is performed if Red and a single Blue color are identified. If the colors Red and Blue are identified, a double-click action is executed.

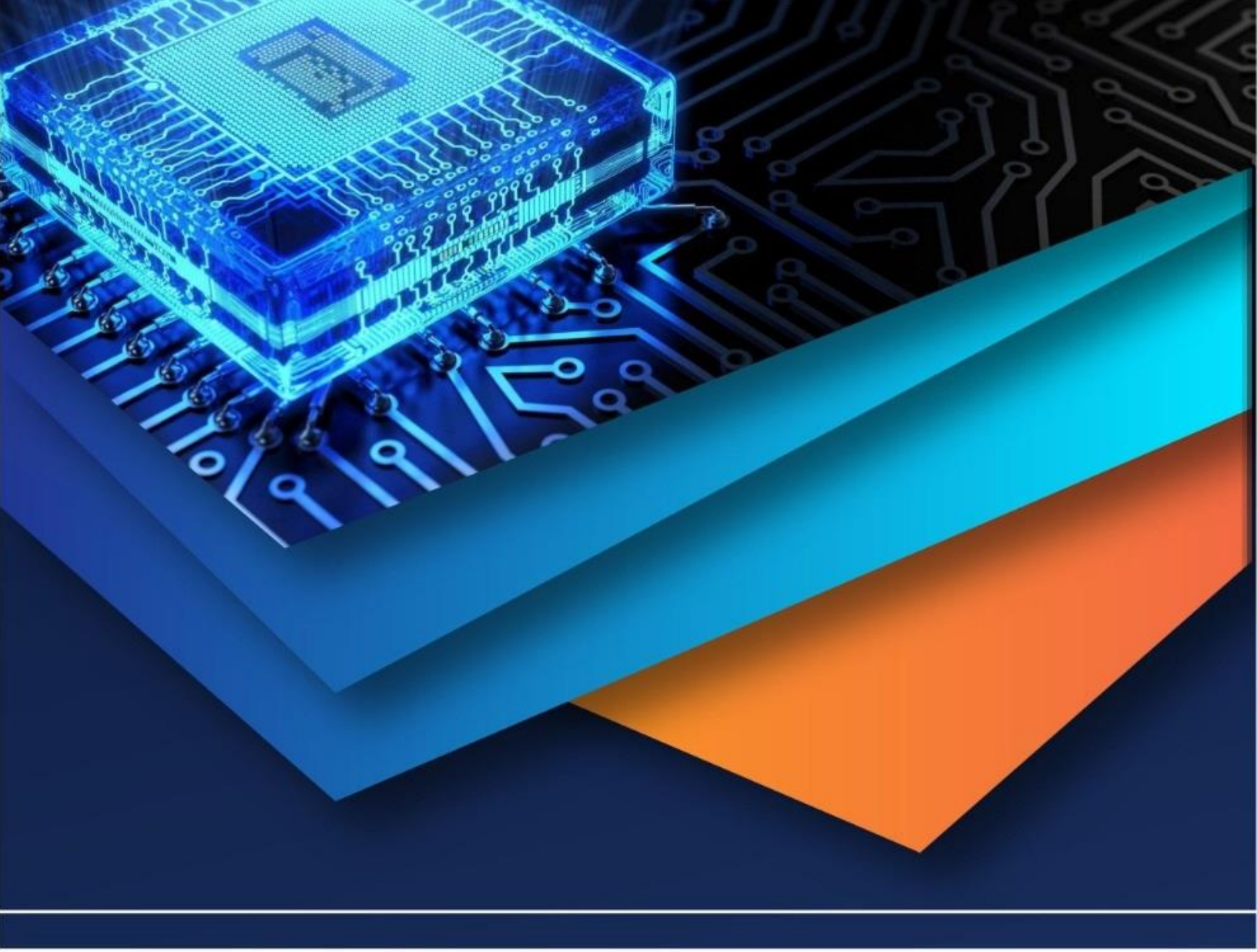
IV. CONCLUSIONS

This system uses a real-time camera to identify red objects and some Matlab code to perform mouse functions such as right-click, left-click, double-click, and right and left movement. This system works with all mouse tasks and is based on computer vision techniques. However, because of the diversity of human skin tones and colours, it is difficult to obtain consistent results. The majority of vision algorithms have difficulty with illumination. According to the findings, if the vision algorithms can work in all conditions, our system will be more efficient. This system could be handy in presentations as well as for saving space.

REFERENCES

[1] Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E.“Computer vision-based mouse”, Acoustics, Speech, And Signal 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 Share 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

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)