

Cursor Movement Control Using Colour Detection

Ashvini Shastrakar, Jitesh Raman, Mamun Paul, Neha Ramteke, Prof. Vishal Sathwane

Computer Science and Engineering, JD College of Engineering, Nagpur, Maharashtra, India

ABSTRACT

This paper mainly focuses on increase of human-computer interaction. Through this project we have tried to control cursor movement with all the click events using colour detection through web-cam. This method provides an easy way to handle the human-computer interaction in a cost effective manner.

Keywords : Human-Computer Intercommunication, Cursor Control, Motion Detection, Gestures Recognition, HSV (Hue Saturation Value), OpenCV.

I. INTRODUCTION

Since the technology is growing up tremendously and in the same manner human-computer interaction is also growing. Today touch screen technology has evolved so much that every devices has adopted this technique. Voice command is also one of the great technologies that we are using in day to day life. Since these technologies are not cheap that it could be used in desktop systems. Creating a virtual mouse or keyboard using computer vision techniques and sensor technologies can be a way for replacing the touch screen and voice command technology. Gestures are powerful means of communication among humans. In fact, gesturing is also in our communication with other people by often continuing gesture while speaking on the cell phone. To provide ideas it requires a separate complementary modality speech. In conversation, information associated with hand gestures which include is degree, discourse structure, spatial and temporal structure. So, a natural correlation between humans and computing devices can be achieved by using hand gestures for communication.

In our work, we have tried using a web camera based on colour detection technique to control cursor movement with all click events. The user must wears colored tapes or capes to give commands to the system. The real time video captured by the camera must process individual frames are separately for the execution. The processing methods requires an image background subtraction algorithm for the detection of colour. Through detected colours the system performs many different operations to track the cursor and perform all the cursor actions. No extra hardware is required other than the standard webcam.

II. METHODS AND MATERIAL

A. System Description

Steps involved in developing our system:

1. Capturing real time video through web camera.
2. Every individual image frame is then being processed.
3. Flipping of each image.
4. Conversion of each image frame to a grey scale image.

5. Detection and extraction of colors (RGB) will be taking place by the flipped grey scale images.
6. Conversion of the detected image into a binary image.
7. Finding the region where the image is being detected and calculating the centroid of that image.
8. From the calculated centroid of that image the pointers of mouse will be tracked using the coordinates that are obtained.
9. Allocating different color pointers left and right click events of the mouse are simulated.

B. Capturing Real Time Video using Webcam

For the detection of hand movements of the user we are using the webcam that will work as a sensor. The hand movements of user are captured at a fixed frame ratio and resolution which is obtained by the hardware of the camera. By changing the HSV value the frame ratio and resolution can be changed if required. These operations on captured images are done with the help of a tool mainly called Open CV.

- The captured video by the webcam is divided into individual Image frames based on the Frames per second of the camera.
- Process the individual image frames.

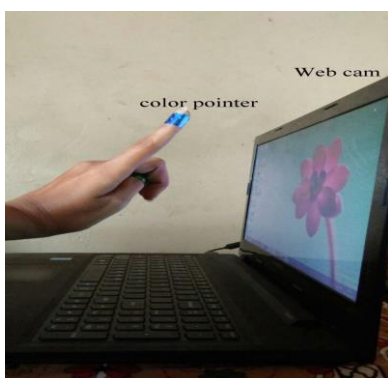


Fig 1. Tracking Color Pointer

C. Flipping of Images

Image captured by the webcam is inverted. In other words with color pointer if we move our hand towards left, the image of the pointer moves towards right and vice-versa. It is similar to the standing in

front of a mirror. So we need to flip the images vertically. Flipping can't be performed, as the image captured is an RGB (Red, Green, and Blue). So we need to separate the individual color channels of the captured image and then perform the flipping actions of those images individually. After flipping colored channels individually, the results are concatenated and a flipped RGB image is obtained. The following images show the entire real time flipping process.

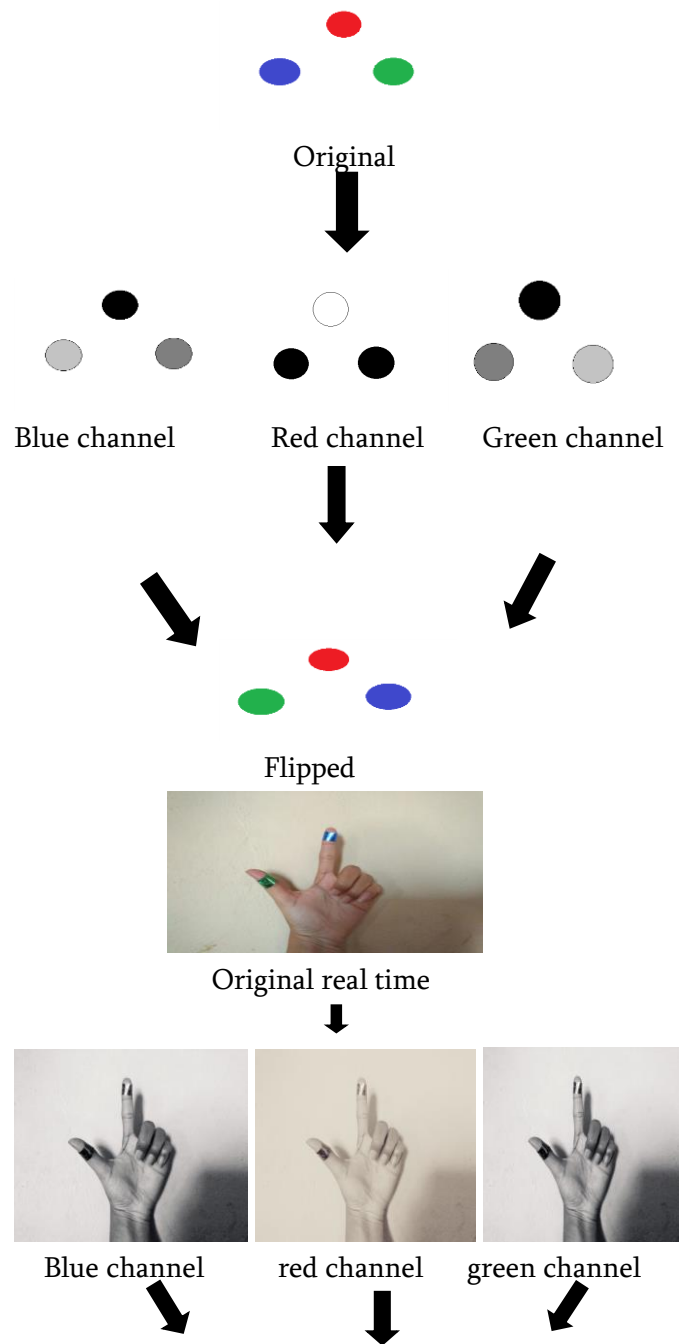




Fig 2. Flipping of real time image

D. Converting flipped image into grey-scale images

Computational complexity is decreases in a gray scale image, as compared to a color image. In this flipped image is converted into gray-scale. All operations were performed after conversion of image into gray-scale.

E. Color Detection:

It is the most important process in this project. Here we used blue and green color tape to detect the image. Object is detected by subtracting the flipped color channel from the flipped grey-scale image.

A color detection method identifies the pixels in an image that match a specified color or color range. Object detection is the most important task of computer vision.

We used blue color for controlling the cursor and green color for open and close the application.



Flipped red channel



Flipped gray-scale image



Extracted red color

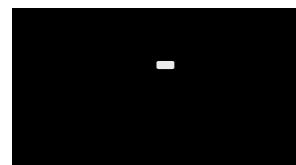
Fig 3. Detection of Blue and Green Color

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

For finding the region of the detected image we need to convert the gray scale image into binary image

after the subtraction. A grey-scale image consists of matrix which contains the values of each pixel. The values of the pixel ranges from 0 to 255 where 0 represents pure black color and 255 represents pure white color. We need a threshold value of 20% that will convert the grey-scale to a binary image and all the pixel values that are below 20% of the maximum pixel value will be converted into pure black i.e. 0 and the rest will be converted to white that is 1.

Thus the resultant image will be in b/w colored image. Open CV contain properties of a black and white (monochromatic) image.



Detected region

Fig 4. The obtained monochromatic image

G. Finding Centroid of an Object:

First define a point whose coordinates are stored in a variable can be sent to the cursor with these coordinates the movement can be controlled by the system.

Functions of OpenCV are used to find the centroid of the detected region. It contains matrix of X and Y coordinates.

Once coordinates are obtained the mouse driver is accessed and coordinates of the centroid are sent to cursor. Coordinates make the cursor places itself in the required position. It is assumes that pointer moves continuously. Each time a new position of pointer is obtained which affect the tracking. So, the user moves across the screen.

OpenCV doesn't contain any inbuilt function to access directly drivers often complete that's why we are using JAVA. Since JAVA is platform dependent language. So JAVA object is created to access the drivers. Colour code can be modified based on the system. The control action of mouse are performed by

controlling the flag related with mouse button we use JAVA.

- Use of colour pointers the computational time is reduced.
- System becomes resistant to background noise.

The presence of other colour in the background may produce an error which can be reduced to configuration threshold values and other parameters.

H. Trace the Mouse Pointer

When the coordinates has been obtained, the mouse pointer is accessed and obtained coordinates of the centroid are sent to the cursor and according to coordinates, the cursor places itself in the required position and pointer moves continuously. Whenever a new centroid is obtained for each frame, the mouse cursor obtains a new position of that pointer, thus creating an effect tracing. As the user moves his or her hands or fingers wearing colour tape across the view of the webcam, the mouse moves accordingly across the screen.

In OpenCV, its code supports integration with languages like JAVA. Java is a platform independent language, where a java object is created to access the mouse pointer of the computer.

Depending on the detection of colours the system performs the clicking action and all moving events of the mouse on the computer. These colour codes can be improved depend on the requirements of the system.

I. Performing Clicking Actions:

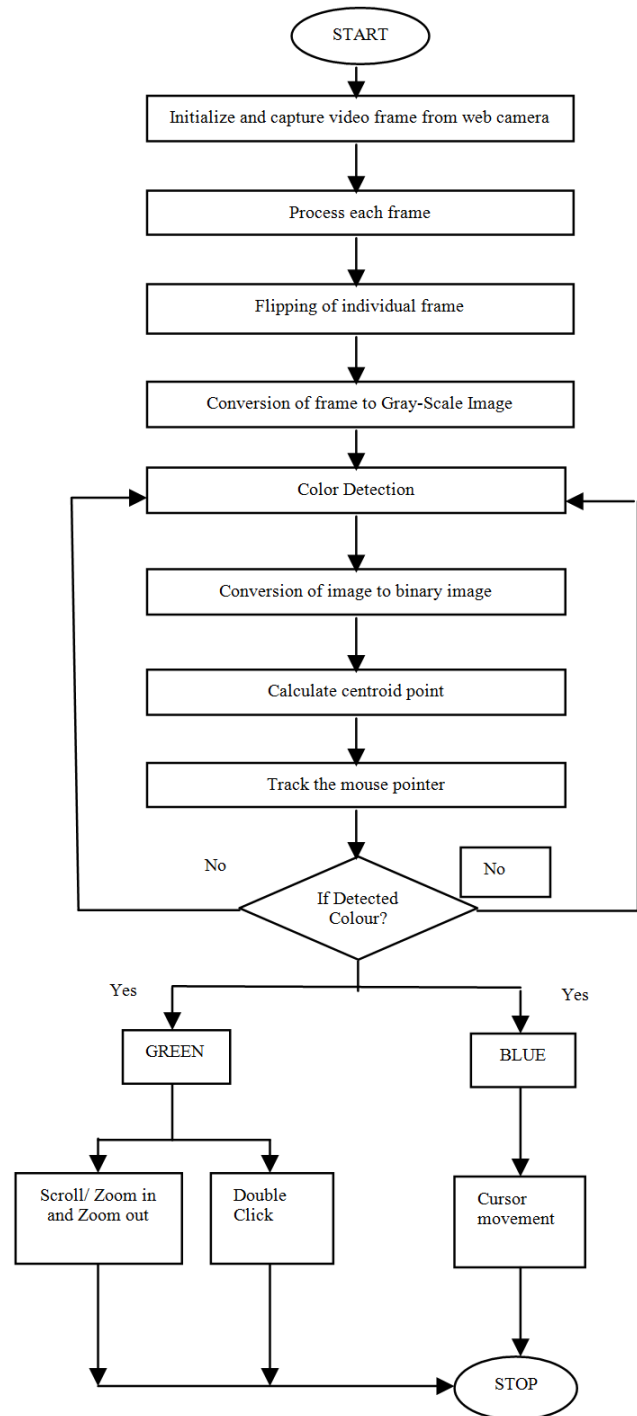
Clicking action is performed by simultaneous detection of two colours:

If Blue colour is detected, then clicking action is performed.

If Blue along with Green colour is detected, opening and closing of the application is performed.

III. RESULTS AND DISCUSSION

A. Cursor Control Flowchart



B. Problem and Drawback:

Since this system is capturing the image through a webcam. In this system if in certain cases web camera cannot capture the image properly then operation cannot perform.

In some cases due to presence of other colors 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 due to the high resolution of the camera.

User should always sit nearer to the desktop, because external hardware mouse and resolution of the camera. If mouse is damage in any situation then cost will increase.

IV. CONCLUSION

We have design a system which control mouse cursor and implement mouse cursor using a real time web-camera. Implementation of all mouse function such as open the application using hand gestures like notepad, paint, command prompt etc. A new technique has been proposed to increase the adaptability and response time of system. We have developed a system to control and implementation its function using a real time camera. The key of this project is to make human and machine interface, easy and simple interactive. This technology can be used to help the patient. Who do not have control of their hand? Our aim is to create this technology in a cheapest way.

We used open CV library in this project, it is the strength of the proposed method in this paper.

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

- [8]. Mastering OpenCV with practical computer vision projects
- [9]. Learning OpenCV by Gary Bradski and Adrian Kaehler.

Reference Websites

- [10]. <https://www.opencv.org>
- [11]. <https://stackoverflow.com>
- [12]. <https://www.dreamincode.net>
- [13]. <https://msdn.microsoft.com/>
- [14]. <https://www.technical-recipes.com>
- [15]. <https://www.intorobotics.com>
- [16]. <https://www.progcode.in>
- [17]. <https://www.mathworks.in>