Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition

Vantukala VishnuTeja Reddy[1], Thumma Dhyanchand[2], Galla Vamsi Krishna[3], Satish Maheshwaram[4]
Department of Electronics and Communication Engineering, National Institute of Technology Warangal, Warangal, India – 506004
[1]vishnuvantukala@gmail.com, [2]dhyanchandthumma.098@gmail.com, [3]vamsigalla21@gmail.com, [4]satishm@nitw.ac.in

***Abstract*— In human-computer interaction, virtual mouse implemented with finger tip recognition and hand gesture tracking based on image in a live video is one of the studies. In this paper, virtual mouse control using finger tip identification and hand gesture recognition is proposed. This study consists of two methods for tracking the fingers, one is by using colored caps and other is by hand gesture detection. This includes three main steps that are finger detection using color identification, hand gesture tracking and implementation on on-screen cursor. In this study, hand gesture tracking is generated through the detection of the contour and formation of a convex hull around it. Features of hands are extracted with the area ratio of contour and hull formed. Detailed tests are performed to check this algorithm in real world scenarios.**

Keywords—color identification, finger tracking, hand gesture recognition, image processing, virtual mouse.

I. INTRODUCTION

Image processing, a division of signal processing, can consists of an image or a video as input and output as an image or various parameters of it. Gesture recognition and tracking is a kind of image processing process.In recent times, a number of gesture recognition techniques have been proposed. Hand tracking has several applications including motion capture, human-computer interaction and human behavior analysis [1][2]. Several types of sensors and detection gloves are used for hand motion detection and tracking [3][4]. Instead of using more expensive sensors simple web cams can identify the gesture and track the motion.

The main objective is to find the solution for the finger tracking in the real world and the control of cursor of a computer is still performed physically. There may be some difficulty in most of the applications to control the mouse physically. We can make use of web cam and with the help of some algorithms, we can control the cursor operations without touching the mouse physically [5]. This work presents the implementation and analysis of real time tracking of fingers which involves in making a gesture so that gestures can be used in various applications of mouse like movement, single click, double click, right click, scrolling [6].

Basically it can be done in two methods of identifying fingers. One is by using colored fingertips, other is by using bare fingers and hand gestures [7][8]. We can identify gestures using neural networks [9]. Color identification and tracking is simpler than using algorithms like neural networks [10]. We use some color tip on fingers to identify and then detect the motion of the color by background subtraction and improving this for bare finger gesture tracking without any use of colored tips [11]. This involves processing of a running video using image processing algorithms and then track the fingers [12].

This work was equally contributed by the first three authors and was supervised by Dr. Satish Maheshwaram.

This paper is organized as follows: Section II describes the overall system description, color identification, gesture identification in detail. Section III shows the experimental results and how the system works. Finally, this paper leads to a conclusion in Section IV.

II. METHODOLOGY

In this paper, we design an algorithm to detect the fingers, identify the gestures and to control the operations of a mouse. This paper consists of three sub-sections namely color identification, gesture identification and virtual mouse control along with the overall system description.

In this paper, we use the Python language for the implementation of algorithm. We use the OpenCV library for image processing and pyautogui library for mouse control. The algorithm use two types of methods for implementation of mouse control. One includes the usage of color caps and other uses the recognition of bare hand gesture recognition.

A. Overall System Description

The overall diagram of the system is depicted in Fig. 1. It is divided into two methods; 'fingertip detection' using colored caps and 'gesture recognition'. It involves the interfacing of the video, processing the images by background subtraction. Background subtraction helps in neglecting the steady objects considering only foreground objects. Fingertip detection involves the color identification, circle identification and guessing the fingers. Gesture recognition involves skin color identification, contour detection and convex hull formation and then guessing the gesture. Respective mouse operation can be done accordingly.

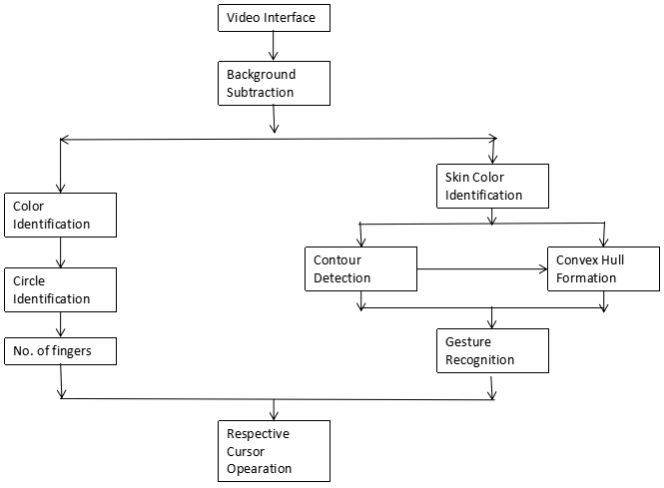

Fig 1. Overall system block diagram

B. Color Caps

This includes four stages to get the finger gesture from the real time fed live video. They are namely video

978-1-7281-4094-3/20/$31.00 ©2020 IEEE

extraction, background subtraction, color identification and circle identification.

1) *Video Extraction:* Initially, the video taken by web cam is taken using the Video Capture function in OpenCV library. The extracted video has a size of 512x512 pixel. Then, from the video, a frame is extracted which is used for further processing of the image. The optimum number of frames that are extracted per second are 12. The extracted frame image is in BGR(Blue, Green, Red) format. This image is converted to HSV(Hue, Saturation, value) format using BGR to HSV conversion function of OpenCV.

2) *Background Subtraction*: The frame image extracted contains multiple objects in it. But only the foreground that contains the hand with color caps is required for processing. Hence, the background has to be neglected. So, background subtraction algorithm from OpenCV is required for the extraction of foreground objects. In background subtraction algorithm, the present frame image is compared with a 21st prior frame image in the past. The objects that had no motion in 20 or higher frames are considered to be the background objects and hence neglected. Since, the motion of the color caps is always present while performing the mouse operations, they are considered as the foreground objects. The shadows of the hand in the frame image are neglected using the command word detect shadows in background subtraction algorithm. After background subtraction algorithm, we get a masked greyscale image containing the foreground objects. The bit-wise 'and' of this greyscale image with the original frame image gives the background subtracted image. This phenomenon can be seen in Fig. 2 where there is no motion of hand and result for the motion of hand is as shown in Fig. 3.

3) *Color Identification:* The background subtracted image is scanned to get the color caps of a required color. The lower and upper ranges of a particular color in HSV(Hue, Saturation, Value) format are given as

Blue: Lower limit = [80, 70, 70]

Upper limit = [120, 255, 255]

Red: Lower limit = [0, 100, 100]

Upper limit = [20, 255, 255]

Green: Lower limit = [60, 100, 100]

Upper limit = [60, 255, 255]

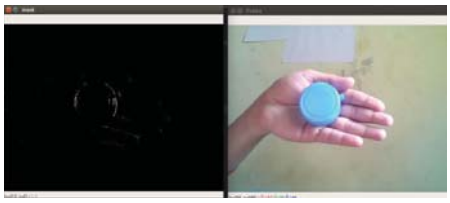

Fig. 2 Hand with no motion

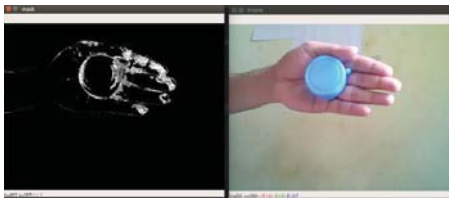

Fig. 3 Hand in motion

Using the 'in range' function in OpenCV library, a greyscale image with only the objects of the required color are extracted. Bit-wise 'and' of this greyscale image with the background subtracted frame image gives the image with only the objects of required color as shown in Fig. 4 for red color identification, Fig. 5 for blue color identification and Fig. 6 for green color identification from a group of colored circles. This is used to identify the color cap of required color among different colored caps.

4) *Circle Identification:* After the objects of a particular color that are in motion are identified, the shape of the objects that coincides with the color caps is identified. The color caps that are taken in the project are in circular shape. Hence, the objects that are circular in shape are identified. The Hough Circles function in OpenCV is used to identify the circular color caps. The radius of the circle is specified in a particular range so that color caps placed at optimum distance from the camera are identified. The range values lie between 50 to 200 pixels which corresponds to a distance of 0.5 m to 1 m. The Hough circle function gives the result in a two dimensional list that contains the radius and centers of all the colored circles present in the image and can be drawn according to the co-ordinates as shown in Fig. 7.

C. Gesture Recognition

Identification of fingers can be done through smoothening of frames using bilateral filter function, then subtracting the background of images. Each frame is then checked with the color range and there contours function is used to obtain bounding shape of the palm and convex hull is formed using convex hull function.

Fig. 4 Red color extracted image

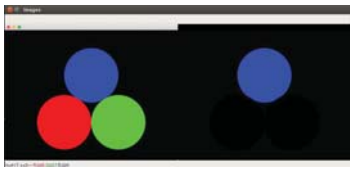

Fig. 5 Blue color extracted image

Fig. 6 Green color extracted image

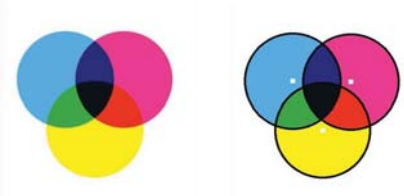

Fig. 7 Circle Identification

Contours function is used to get the outline along the palm of a hand. It can be done by checking the color of the palm with the mentioned color range and then an approximate boundary of palm is detected and drawn accordingly which gives an outline of the palm as contour. Convex envelope of a set A of points is the smallest convex that contains A. Convex hull technique is used to get the hull of the entire contour formed. In this, the points on the contour are checked such that all the points are inside the hull with the extreme, outer most points as some of the points on the hull.

Fig. 8(a) represents the normal image of a palm. Fig. 8(b) represents HSV image formed from the normal image. Fig. 8(c) represents the image for finger detection where green lines are contours along the palm and red lines are obtained after convex hull technique.

Different hand gestures are identified with the help of contour area and convex hull area formed by the palm. The area ratio gives an idea of a particular gesture. Number of fingers can be identified by the number of peaks formed in the contour.This represents the number of fingers opened freely.

The areacnt is the area of the hand which is palm and areahull is the area of the hull formed by the palm.

arearatio = ((areahull - areacnt) / areacnt) * 100

The above statement provides the percentage of area that is not covered by hand in convex hull. So gesture can be identified by checking the number of fingers and the area ratio which gives the name of finger that is used as shown in the TABLE 1.

'l' is the number of fingers identified.

D. *Mouse Control*

Mouse Control is performed with the help of pyautogui library. The fuctionalities of mouse that are performed in the project are cursor control, left click, left double click, right click, up scrolling and down scrolling.

Three fingers are used for the mouse control operations in case of color caps. Those three fingers are index, middle finger and ring finger. Blue colored cap is inserted on index finger. Green color cap is inserted on middle finger and red color cap is inserted on ring finger.

Fig. 8 (a) normal image (b)Skin mask image (c)Contour and hull formed.

TABLE 1. Gesture corresponding to area ratio

l	Area ratio (a)	Gesture / fingers opened	l	Area ratio (a)	Gesture / fingers opened	l	Area ratio (a)	Gesture / fingers opened
-	Hand area < 2000	No hand	1	a>18	Index finger	3	a>27	OK sign
-	a<=12	Fist	2	-	'V' sign	4	-	4 fingers
1	12<a<=18	Thumb	3	a<=27	3 middle fingers	5	-	palm

In case of hand gesture mouse control, various hand gestures that includes all five fingers of the bare hand are used for mouse control operations.

1) Cursor Control: In case of color caps, blue color alone is used for the movement of cursor. Green and red caps are closed. While using hand gesture recognition technique, index finger is used for cursor control.

2) left Click: Green color cap is used for left click operation with remaining color caps closed. In hand gesture recognition technique, index and middle fingers are used for single left click.

3) Double Click: Blue and Green color caps together perform the double click operation. In hand gesture recognition technique, index, middle finger, ring finger and little finger together gives left click operation.

4) *Right Click:* Red color cap performs right click operation. Index, middle finger and ring finger together gives right click operation in hand gesture recognition technique.

5) *Up-Scroll:* Red and blue colors identified together gives up scroll function. In hand gesture recognition technique, middle finger, ring finger and little finger together gives down scroll operation.

6) *Down-Scroll:* Red and green colors together gives down scroll function. In hand gesture recognition technique, middle finger and little finger together gives down scroll operation.

III. RESULTS

We have applied the above discussed algorithms for three different color caps and acquired the results for various mouse functionalities. The experimental figures are given in figures below for each mouse operation. A series of five result images are given for each mouse operation as shown in the following group of figures where Fig. 9 represents the set of images for cursor movement operation, Fig. 10 represents the group of images for left click operation, Fig. 11 shows the set of images for double click operation, Fig. 12 represents the series of images for right click operation and Fig. 13 represents the group of images for scrolling operation of a mouse.

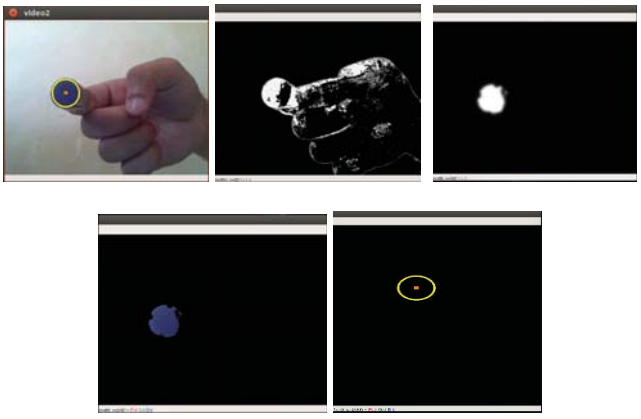

Fig. 9 Flow of Cursor Movement Operation

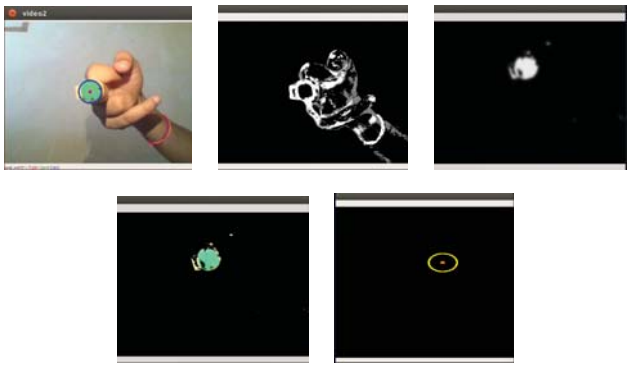

Fig. 10 Flow for Left Click Operation

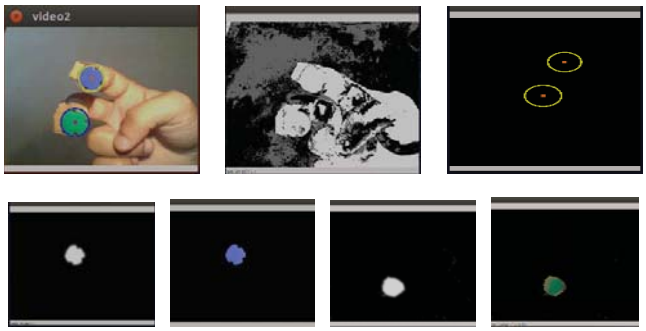

Fig. 11 Flow for Double Click Operation

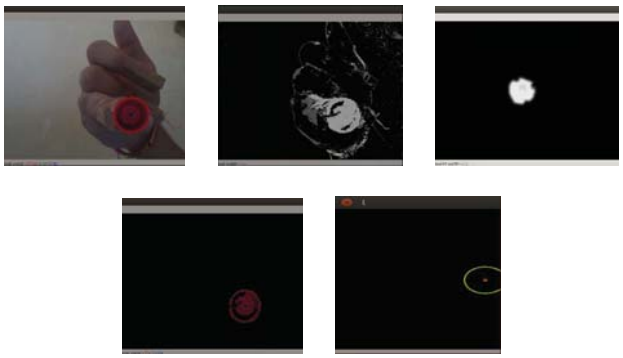

Fig. 12 Flow for Right Click Operation

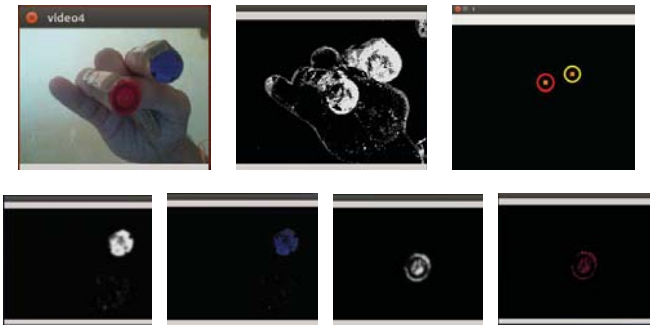

Fig. 13 Flow for Scrolling Operation

In this series, the first image is the original frame image with color caps on respective fingers. The color cap is has a center shown as a rectangular box in middle surrounded by a circle of radius equal to the radius of the color cap. The second image in the series is the background subtracted image. The third image in the series is the greyscale image after color identification. The fourth image is the image that is 'bit-wise and' of greyscale and frame image. The fifth image gives the circle that is identified with center in rectangular shape surrounded by circle. For this approach, the color detection accuracy variation with the brightness of the surroundings is observed. As shown in Fig. 14, all the three colours Red, Blue and Green have highest detection accuracy of > 90% when the brightness is around 500-600 lux, which is the average brightness in offices and well lit classrooms. In order to overcome this issue, hand gesture recognition technique is adapted where the hand shape contours is detected.

For hand gesture recognition, we use all the five fingers and take five different hand gestures for the different mouse operations. Those five operations are cursor movement, left click, double click, right click and scrolling. We have series of three images for each mouse operation as shown in the following group of figures where Fig. 15 represents the group of images for cursor movement operation, Fig 16 represents the group of images for left click operation of a mouse, Fig. 17 represents the flow for right click operation, Fig. 18 represents the set of images showing double click operation and Fig. 19 shows the group of images for scrolling operation. The first image is the normal frame image. The second image in the series gives the skin color identified, background subtracted greyscale image. The third image in the series gives the contour and convex hull identified image. In the third image, the contours of the hand gesture is enclosed by the green lines joining the coordinates given by contour function. These coordinates are used to form the convex hull which is shown by red lines enclosing the hand gesture. The different mouse operations are as follows.

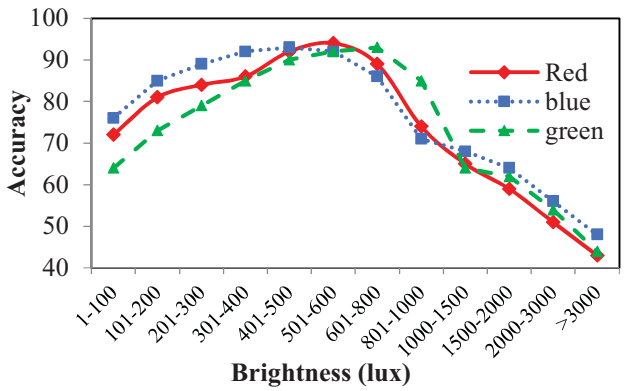

Fig. 14 Variation of accuracy with brightness

Fig. 15 Flow for Cursor Movement Operation

Fig. 16 Flow for Left Click Operation

Fig. 17 Flow for Right Click Operation

Fig. 18 Flow for Double Click Operation

Fig. 19 Flow for Scrolling Operation

IV. Conclusion

A method for on-screen cursor control without any physical connection to a sensor is presented. Identification of colored caps on the fingertips and their tracking is involved in this work. Different hand gestures can be replaced in place of colored caps for the same purpose. Different operations of mouse controlled are single left click, double left click, right click and scrolling. Various combinations of the colored caps are used for different operations. Range of skin color can be varied in the program in accordance with the person to be used, surrounding lightening conditions. An approximate area ratio that is not being used by the hand in the convex hull is taken after analyzing the program output at different gestures of the hand.

This work can be used in various real time applications like cursor control in a computer, android based smart televisions etc. Although there are devices like mouse and laser remotes for the same purpose, this work is so simple so that it reduces the usage of external hardware in such a way that the motion of fingers in front of a camera will result in the necessary operation on the screen.

References

[1] N. Shaker and M. A. Zliekha, "Real-time Finger Tracking for Interaction," 2007 5th International Symposium on Image and Signal Processing and Analysis, Istanbul, 2007, pp. 141-145.

[2] R. M. Prakash, T. Deepa, T. Gunasundari and N. Kasthuri, "Gesture recognition and fingertip detection for human computer interaction," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-4.

[3] A. Dekate, A. Kamal and K. S. Surekha, "Magic Glove - wireless hand gesture hardware controller," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2014, pp. 1-4.

[4] J. Suh, M. Amjadi, I. Park and H. Yoo, "Finger motion detection glove toward human-machine interface," 2015 IEEE SENSORS, Busan, 2015, pp. 1-4.

[5] R. S. Baştuğ, B. Yeşilkaya, M. Unay and A. Akan, "Virtual Mouse Control by Webcam for the Disabled," 2018 Medical Technologies National Congress (TIPTEKNO), Magusa, 2018, pp. 1-4.

[6] A. Mhetar, B. K. Sriroop, A. G. S. Kavya, R. Nayak, R. Javali and K. V. Suma, "Virtual mouse," International Conference on Circuits, Communication, Control and Computing, Bangalore, 2014, pp. 69-72.

[7] S. K. Kang, M. Y. Nam and P. K. Rhee, "Color Based Hand and Finger Detection Technology for User Interaction," 2008 International Conference on Convergence and Hybrid Information Technology, Daejeon, 2008, pp. 229-236.

[8] Y. Fang, K. Wang, J. Cheng and H. Lu, "A Real-Time Hand Gesture Recognition Method," in 2007 International Conference on Multimedia & Expo, Beijing, 2007 pp.

[9] M. Han, J. Chen, L. Li and Y. Chang, "Visual hand gesture recognition with convolution neural network," in 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai, China, 2016 pp. 287-291.

[10] T. D. Grove, K. D. Baker and T. N. Tan, "color based object tracking," Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170), Brisbane, Queensland, Australia, 1998, pp. 1442-1444 vol.2.

[11] N. Vo, Q. Tran, T. B. Dinh, T. B. Dinh and Q. M. Nguyen, "An Efficient Human-Computer Interaction Framework Using Skin Color Tracking and Gesture Recognition," 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), Hanoi, 2010, pp. 1-6.

[12] D. A. Barhate and K. P. Rane, "A Survey of Fingertip Character Identification in Open-Air Using Image Processing and HCI," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, 2018, pp. 1-4.