Gesture Recognition and Hand Movement Tracking in Real Time

Syed Alfran Ali, Shams Ali and Jai Pal Singh and Ashish Sharma
Indian Institute of Information Technology, Kota

Abstract

As technology has become one of the most important part of the day to day human life, so in order to make the best use of those technologies the study of the relationship and interaction between human and computer i.e. called human computer interaction (HCI) is important. In this paper, we are trying to provide a non-tangible way of HCI and that is with the help of hand gesture recognition to control different operations of mouse. To create a virtual mouse, we do object detection using colour segmentation and find contours. Using these contours, we map different mouse operations with different gestures and hence, get virtual mouse.

Keywords

Human Computer Interaction, Hand Gesture Recognition, Object Detection, Mouse Tracking, Object Tracking

1 Introduction

In the present world, the interaction with the computing devices has advanced to such an extent that as humans it has become our necessity and we cannot live without it. The technology has become so embedded into our daily lives that we use it to work, shop, communicate and even entertain our self. It has been widely believed that the computing, communication and display technologies progress further. To efficiently use them, most computer applications require more and more interaction. For that reason, human-computer interaction (HCI) has been a lively field of research in the last few years. Recognizing hand gestures for interaction can help in achieving the ease and naturalness desired for human computer interaction.

Gesture recognition is a type of perceptual computing user interface that allows computers to capture and interpret human gestures to do the certain key-operation binding and perform certain commands. The general definition of gesture recognition is the ability of a computer to understand gestures and execute commands based on those gestures. Hand gesture recognition is considered as one of the best interaction style and non-tangible technique for human computer interaction. In this technique different hand gestures are recognized to solve certain problems like convex-hull problem via jarvis's algorithm etc. and certain actions are being performed based on those gestures. Hand gesture can be categorized into two types of hand gesture representation:

a) Contact Based: In contact-based hand gesture representation, the hand must be in contact with some of the external device as like gloves for the gesture to be recognized. Without the external device contact-based hand gesture representation is meaningless.

b) Vision Based: In vision-based hand gesture representation the hand need not to be in contact with some external device to recognize the gesture. The only requirement for this type of hand gesture representation is that you need a computer inbuilt with camera or computer system with camera.

In computer vision, **object detection** is scanning and searching for an object in an image or a video (which is just sequence of images) and **object tracking** is like you are spying on someone and following it. Done in motion images like in animated gifs or videos, we want to track how an object is moving, where is it going, or its speed. Tracking hand movements and then binding them with mouse operations can be very useful. This virtual mouse can be used in gaming, virtual reality, corporate sector, banking and even in medical field too.

2 Related Work and Background

In this section, we give a brief survey of existing literature that we went through to gain knowledge about the approaches used in developing algorithm of the hand movement tracking, gesture recognition and user interaction system. There are various techniques for implementing the detection and tracking of the dynamic hand gestures such as Haar-like features [1], convex hull [2], contour matching [3], and skin colour [4].

Gesture Recognition and fingertip detection for HCI [5] proposes a novel gesture recognition and fingertip detection algorithm for Human Computer Interaction in particular mouse control operations using real time camera. The hand gestures are captured using real time camera. It uses region growing algorithm followed by morphological operations to segment hand region. The centroid of the palm region is calculated and the finger tips are then detected using the convex hull [2] algorithm. But this algorithm is computationally expensive and is sensitive to noise.

The Vision-Based Hand Gesture Recognition Using Blob Analysis [6] proposes a dynamic hand gesture recognition algorithm. The algorithm implements vision-based hand gesture recognition using optical flow and blob analysis to track six dynamic hand gestures and classify their meanings. But this technique requires clear background-foreground relation and high pixel-precision. Real-Time Marker-Based Finger Tracking with Neural Networks(NN) [7] provides a real-time algorithm as an alternative to inverse kinematics. It uses Keras, Theano and NN techniques.

Real Time Hand Gesture Movements Tracking and Recognizing System [8] uses Convex Hull for gesture recognition and Lucas-Kanade for motion gesture recognition. An Efficient Fast Hand Tracking Approach based on Segmentation [9] considers hand as a complete region instead as a combination of feature descriptors and uses histogram based segmentation algorithm. The Application of Improved Camshift Algorithm in Hand Tracking [10] uses background subtraction algorithm based on mixed gaussian model to extract the movement information of hand, then combine the color feature with movement feature by adopting an adaptive method and the Kalman filter algorithm is used to predict the location of hand. Tracking Hand Movements and Detecting Grasp [11] uses Microsoft Kinect to measure hand movements and a SVM [12] classifier to detect grasp gesture.

3 Proposed Method

In this paper, we propose a method to create a virtual mouse using colour segmentation and gesture recognition. This can be done in two steps. First we will do simple colour detection to detect some colored objects and mark them in live camera view. Secondly, we will do gesture recognition and mouse function binding on top of that.

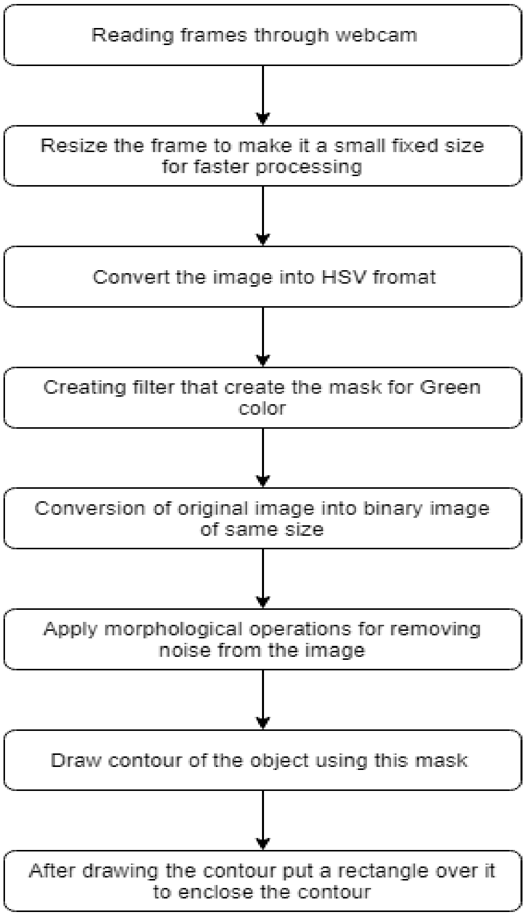
Reading frames through webcam
Resize the frame to make it a small fixed size for faster processing
Convert the image into HSV fromat
Creating filter that create the mask for Green color
Conversion of original image into binary image of same size
Apply morphological operations for removing noise from the image
Draw contour of the object using this mask
After drawing the contour put a rectangle over it to enclose the contour

Figure 1: Colour Segmentation and Object Detection

Object detection using colour segmentation consists of a series of steps. These steps are shown in figure 1. We start from reading frame from the camera and then we will resize it to make it a small fixed size for faster processing. Now we will convert this image to hsv format. We only want green colour object to be detected, rest of the colors we are not interested in. To do that we need to decide a range for hsv value for coloured object. So we declared some limits for the hsv values of each pixels. Now we will create a new binary image of same size a original image, we will call it mask. Mask will contain only those pixels that are in this hsv range.

In the raw mask, there can be some false-positives or noises which are not good for object tracking. To make the tracker work, we need to clean the mask using some morphological operations namely:

a) Opening: to remove all the dots randomly popping here and there in the mask.

b) Closing: to close the small holes that are present in the actual object.

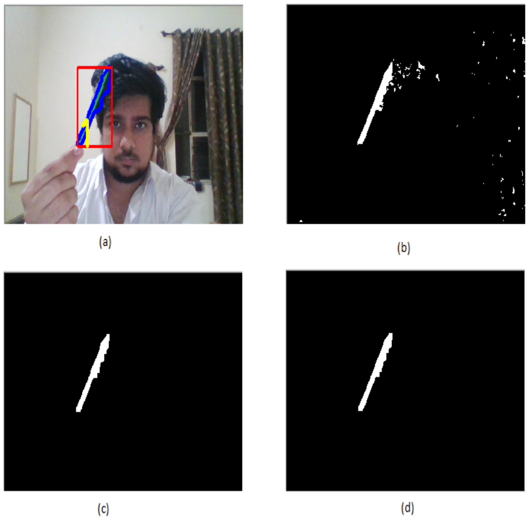

Figure 2: (a)Webcam image, (b)raw mask, (c)open mask and (d)close mask for single object detection

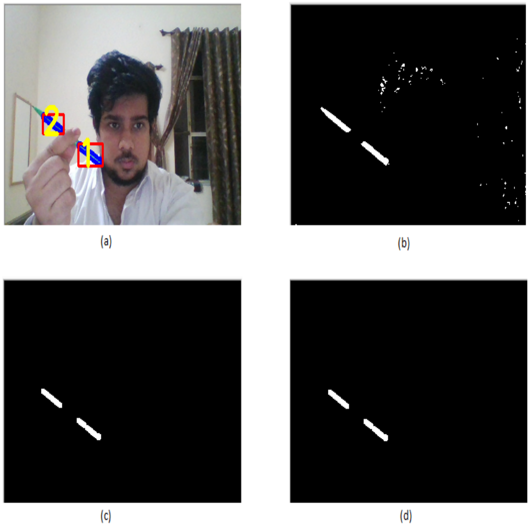

Figure 3: (a)Webcam image, (b)raw mask, (c)open mask and (d)close mask for multiple object detection

Figure 2 and Figure 3 shows web-cam image, raw mask, mask after open morphological operation and mask after close morphological operation for one and two objects. We are using the result of close mask as the final form of image. Now we know exactly where the object is so we can draw a contours from this mask. We will keep count of number of contours and draw a rectangle around each contour for easy detection. This is all to be done in first step.

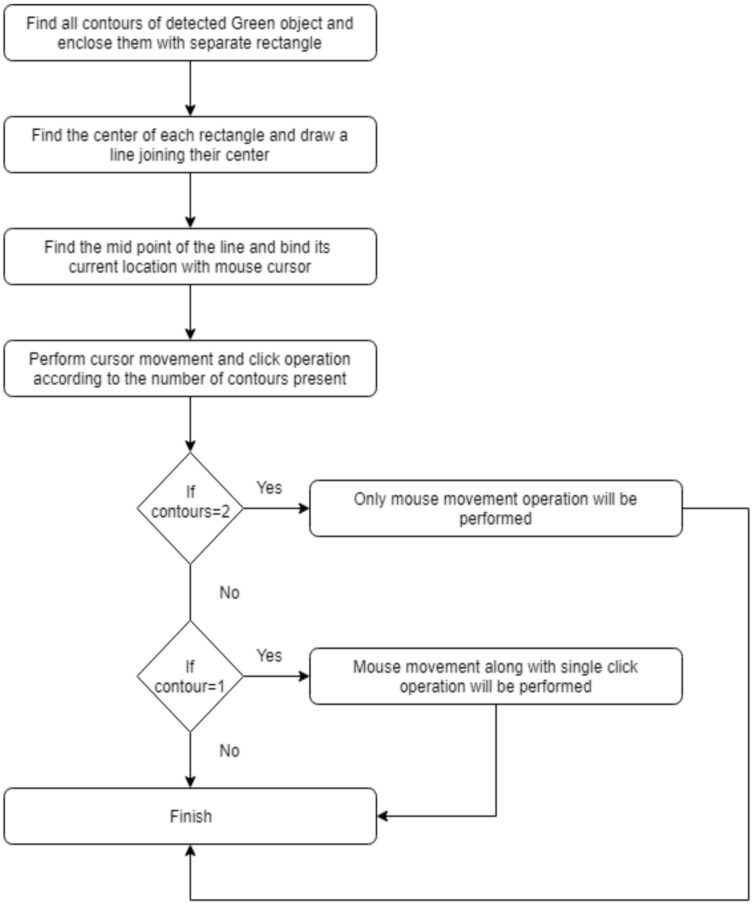
Find all contours of detected Green object and enclose them with separate rectangle
Find the center of each rectangle and draw a line joining their center
Find the mid point of the line and bind its current location with mouse cursor
Perform cursor movement and click operation according to the number of contours present
If contours=2
Yes
Only mouse movement operation will be performed
No
If contour=1
Yes
Mouse movement along with single click operation will be performed
No
Finish

Figure 4: Object tracking and Mouse binding

In second step to create a virtual mouse with gesture recognition, we first need screen resolution and captured image resolution. All the steps for this are shown in figure 4. From the number of contours detected, we have to decide the gesture for each number of contours. We are implementing open gesture if two contours are detected and close gesture if only one contour is detected.

In the contours detected, we have to find a single point that we can bind with the mouse position. There are various ways to find that point. We are using the centre point of the detected contours to draw a line and the mid-point of that line is used to bind with the current location of mouse. We are using the mid-point as it gives least fluctuation in mouse pointer location.

Open Gesture Operation: Open Gesture is mainly used to simulate mouse drag operation on the screen. To implement the open gesture we need to do some calculation to find some coordinates. Firstly, we have to calculate the centre of both detected green objects. We can do this by taking the average of maximum and minimum points of the bounding boxes . Consider x1, y1, w1, h1 and x2, y2, w2, h2 be the x-coordinate, y-coordinate, width and height of first and second detected object respectively. So the centre coordinate of the first object will be

$$cx1 = \frac{x1 + w1}{2} \text{ and } cy1 = \frac{y1 + h1}{2}$$

Similarly, the centre coordinate of the second object will be

$$cx2 = \frac{x2 + w2}{2} \text{ and } cy2 = \frac{y2 + h2}{2}$$

Then, we draw line through the centers of the detected objects. The centre of the line drawn is the average of centre coordinate of these objects i.e.

$$cx = \frac{cx1 + cx2}{2} \text{ and } cy = \frac{cy1 + cy2}{2}$$

These centre coordinates of the line will act as the position for the mouse cursor. To do this, first we need to convert the detected coordinate from camera resolution to the actual screen resolution and after that we set the position as the mouse location. We are doing a mouse release to ensure the mouse left button is not pressed. This is all for the open gesture.

Close Gesture Operation: Close Gesture is used to simulate the mouse left click operation on the screen. The implementation of close gesture is very similar to the open gesture , but the difference is we only have one object here so we only need to calculate the centre of it.

$$cx = \frac{x + w}{2} \text{ and } cy = \frac{y + h}{2}$$

And that will be where we will position our mouse pointer. Also we are performing a mouse press operation instead of mouse release operation.

4 Experimental Results

To verify the effectiveness of the proposed approach, we have done some testing using I7 1.7 GHz machine and the image sequences are captured at about 30 frames per second.

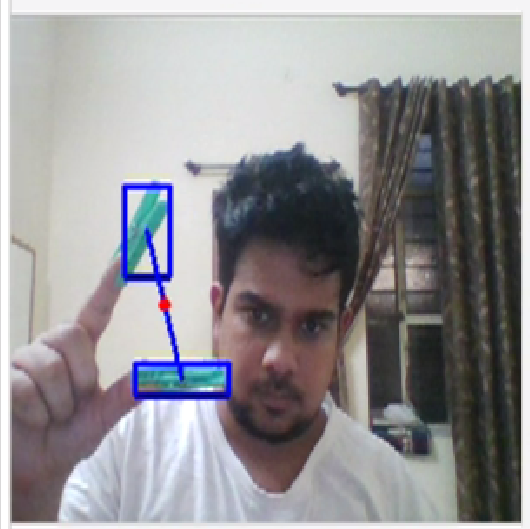

Figure 5: Open gesture recognition

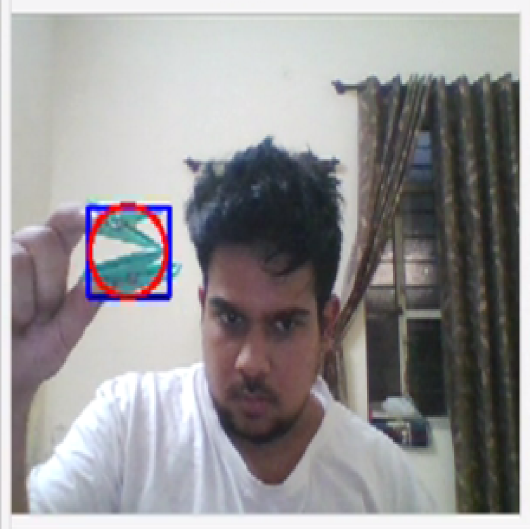

Figure 6: Close gesture recognition

To evaluate the performance of recognition proposed, we examine the system to operate the windows application using hand gesture to replace mouse as user application interface. Testing is done under normal illumination, system can recognize, and track the movement of user hand. The center of gesture represented by a small circle had successfully follow the user's hand movement.

The segmentation process is capable in detecting the user's hand gestures and counting the contours. Figure 5 shows open gesture operation. System is correctly detecting the object on user's hand and is accurately enclosing the detected object in rectangular region. The mouse pointer is moving precisely as the small red dot is moving. Figure 6 shows the close gesture operation. Mouse left click operation is working properly with this operation. We can easily perform the basic window operations such as drag, scroll, select etc.

However, there are some limitations of the program, i.e. the change of lighting environment is still influential in the process of segmentation, in particular on the process of the removal of the background.

5 Applications

Most tasks on a computer can be accomplished solely by using the left mouse button. Left-click is often referred to as normal-click or regular-click. Pressing the left mouse button simulate the mouse cursor being pressed down on the screen which is used to directly interact with files, links and screen items for clicking and dragging.

This method of HCI can be applied in various areas including medical system which is valuable for the elderly people who are not able to talk or express their feelings by words. The use of a monitoring system with virtual mouse can be beneficial for both elderly and caregivers. With very minute modifications, it can be used in gaming, banking and other areas.

6 Conclusion

Based on the evaluation and discussion above, it can be concluded that by using the above proposed methodology the system can recognize and track the hand movement and can replace the mouse function to move the mouse cursor and the mouse left click function as well. The segmentation algorithm using color detection is good enough to detect particular colour in the image and separate it with its background. In general, the system can detect and follow the hand movement and can be used as user interface in real time.

7 Future Work

The future work is to improve segmentation process in eliminating the effect of illumination changes. It is also possible to make the mouse right click function work using a different object detection technique.

References

[1] Chen-Chiung Hsieh and Dung-Hua Liou. Novel haar features for real-time hand gesture recognition using svm. *Journal of Real-Time Image Processing*, 10, 06 2012.

[2] Pablo Barros, Nestor T. Maciel-Junior, Bruno J.T. Fernandes, Byron Bezerra, and Sergio M.M. Fernandes. A dynamic gesture recognition and prediction system using the convexity approach. *Computer Vision and Image Understanding*, 155, 10 2016.

[3] Yiyi Ren, Xiang Xie, Guolin Li, and Zhihua Wang. Hand gesture recognition with multi-scale weighted histogram of contour direction (mswhcd) normalization for wearable applications. *IEEE Transactions on Circuits and Systems for Video Technology*, PP:1–1, 09 2016.

[4] Wei Wang and Jing Pan. Hand segmentation using skin color and background information. In *2012 International Conference on Machine Learning and Cybernetics*, volume 4, pages 1487–1492, July 2012.

[5] R. M. Prakash, T. Deepa, T. Gunasundari, and N. Kasthuri. Gesture recognition and finger tip detection for human computer interaction. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–4, March 2017.

[6] T. Ganokratanaa and S. Pumrin. The vision-based hand gesture recognition using blob analysis. In *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, pages 336–341, March 2017.

[7] Thibault Herbelin Bruno Boulic Ronan Pavllo, Dario Porssut. (real-time marker-based finger tracking with neural networks. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 456–461, March 2018.

[8] R. Hartanto, A. Susanto, and P. I. Santosa. Real time hand gesture movements tracking and recognizing system. In *2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS)*, pages 137–141, Aug 2014.

[9] L. Xu, L. Zhang, K. Wang, and K. Mei. An efficient fast hand tracking approach based on segmentation. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 19–24, Nov 2016.

[10] W. Wei, L. Zhijun, C. Zhiping, M. Xiaomeng, Z. Rui, and Y. Ruo. The application of improved camshift algorithm in hand tracking. In *2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, pages 87–90, Dec 2017.

[11] R. S. Chegani and C. Menon. Tracking hand movements and detecting grasp. In *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, May 2016.

[12] M. E. Mavroforakis and S. Theodoridis. A geometric approach to support vector machine (svm) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, May 2006.