Proceedings of the 13th INDIACom; INDIACom-2019
6th 2019 International Conference on "Computing for Sustainable Global Development", 13th – 15th March, 2019
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

Hand Gesture Recognition using Computer Vision

Preetika Chhotray
GGSIPU, Bharati Vidyapeeth's College of Engineering
New Delhi, India
preetikachhotray@gmail.com

Write author(s) name
Affiliation, Name of organization
City, Country
Email Id: Write author mail id

Abstract – The very essence of Human Computer Interaction (HCI) lies in the way in which humans successfully interact with computers. In order to do that, gesture recognition is essential. Hand gesture recognition serves as a method to interact with computers. It provides a natural, innovative and a modern way of non-verbal communication. This paper proposes a method that is efficacious to recognize hand gestures for HCI using computer vision and image processing techniques. Implementation of this particular method would successfully replace hardware devices (eg. keyboard and mouse) for interacting with the computer. This method uses a webcam that takes in the input feed, tracks skin color using histogram approach, detects the hand and then detects gestures.

Keywords – Human Computer Interaction (HCI), Hand recognition, Gesture recognition, Convex Hull, Skin color tracking, Computer Vision, Image Processing

NOMENCLATURE

HCI – Human Computer Interaction
HSV – Hue Saturation Value

I. INTRODUCTION

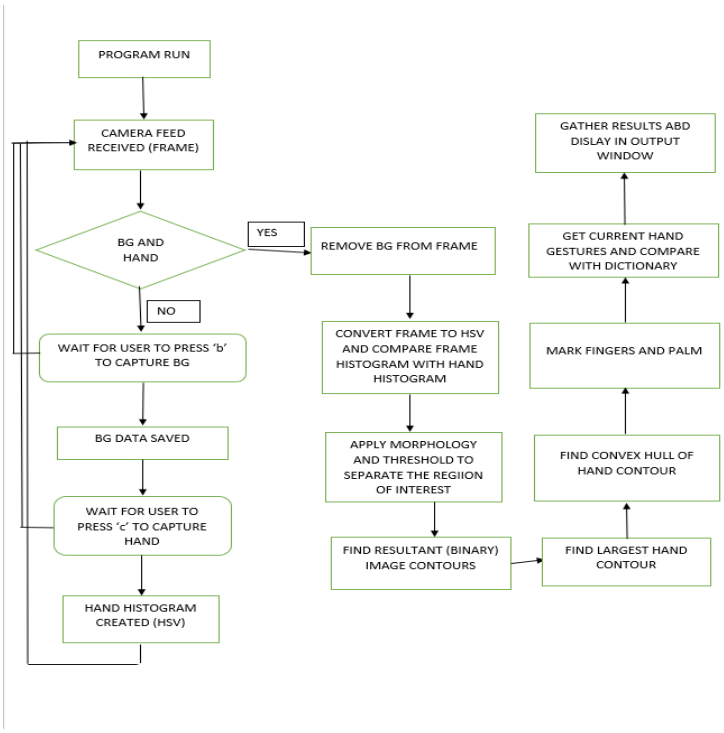
The vision-based technology of Hand Gesture Recognition is a fascinating topic especially in the field of HCI. It is described as method of allowing computers to understand human gestures. There are a number of applications that implement hand gesture recognition for example 3-D gaming, television control, mouse control, making computers more accessible to physically disabled people, virtual reality etc. Most of the hand interactive systems include 3 basic layers – detection, tracking and recognition.

- Detection – It refers to detecting the desired objects on the screen. This can be gathering information about the object features (skin color or shape). Skin color segmentation is one of the most common and effective approaches. This is done by considering any color space from HSV, RGB, YCrCb etc. Though many techniques were proposed, skin detection is still one of the most challenging problems. [1]
- Tracking – Tracking refers to understanding the movements of the object in motion. It helps us to keep a track of the object as well as gives us the knowledge to update the parameters. There are many Data Association techniques that help us to track multiple objects over time entirely based on detection result [2,3].
- Recognition – Here, the information about the hand features are extracted depending on the specific applications.

In the past few decades since the advent of HCI came into picture, hand gesture recognition has become extremely valuable and influential. Many hand gesture recognition techniques were developed and each one of them has their own pros and cons. An example of an older version is wired technology. A lot of work has been done on hand gesture and sign language recognition [4,5]. A sign language recognition has been developed using the Hidden Markov Model in [6].
Another approach is hand driven control, which usually focuses on the hands and fingertips. Kenji Oka proposed a tracking of hands for application in augmented desk interface systems [7]. Another system was proposed which was sophisticated enough to extract hand parameters from monocular image sequences in order to retrieve meaningful information about the hand constellation and 3-D hand posture [8].

II. METHOD DESCRIPTION

The system's method operates as follows. First the skin pixels are detected and a histogram-based approach is used to separate out a hand from the background image. Background cancellation techniques are used to separate out a hand from the background image. The detected hand is then processed and modelled by finding contours and convex hull to recognize finger and palm positions and dimensions. Finally, a gesture object is created from the recognized pattern which is compared to a defined gesture dictionary.



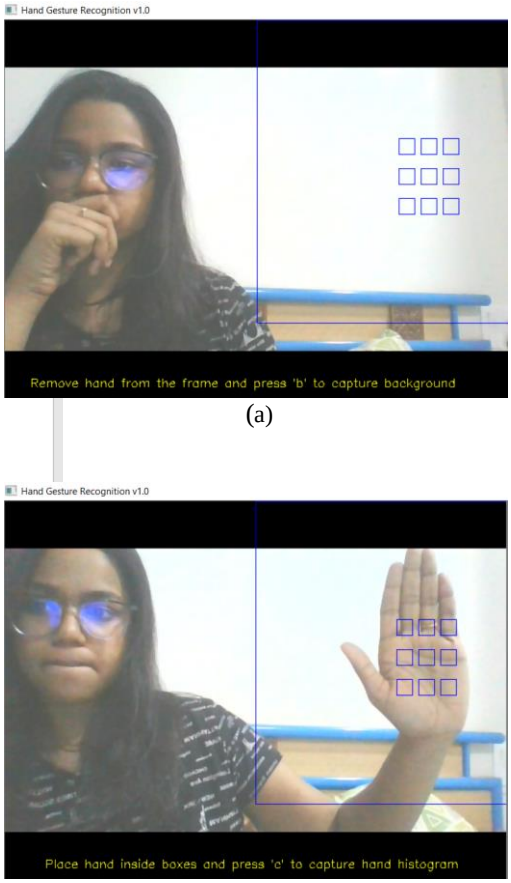
First, the program captures the background. This helps in reducing the detection of false objects which are around the same histogram range as our hand.

A. Skin Detection

Skin detection is one of the most essential features that help us distinguish a person's hand from the background but the skin color varies from person to person and is strongly sensitive to lighting conditions. In this paper, skin color is learned via the histogram-based approach. A histogram is the histogram of it's pixel intensity values. It displays a graph showing the number of pixels in an image at different intensities.

First, the camera input frame is saved to a NumPy array. The program captures the background which helps in reducing the detection of false stationary objects which are around the same histogram range as our hand. The background can be captured by pressing 'b' on the keyboard (a). The next step is to capture the histogram of our hand. In order to detect the pixels and do that, we place our hand inside the 9 square boxes in order to extract the pixels and convert it to HSV color space (b). Here, we are choosing HSV instead of RGB because Hue and Saturation channels are not affected by lighting and other image parameters. Thus, extracting the color information is easier. A background mask is created and applied on the input frame containing only the foreground.

For background subtraction, we use Gaussian-mixture based background/foreground segmentation algorithm.



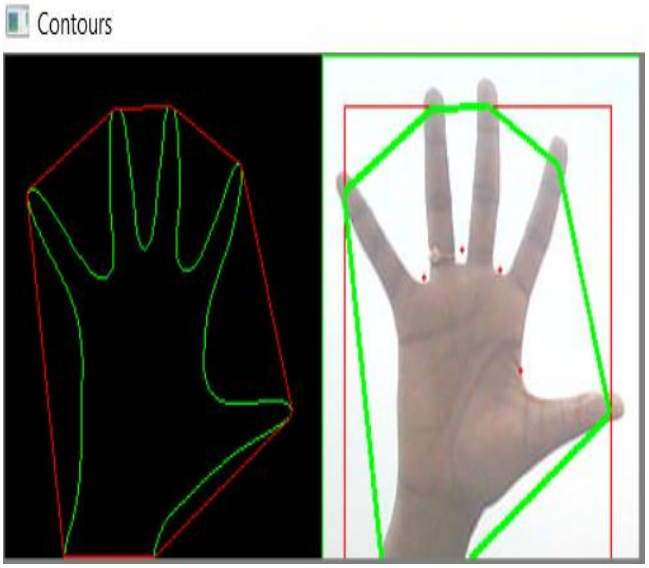
(a) Background capture (b) Histogram capture

B. Contouring and False detection

Once the background is subtracted the rest of the area, back projection is calculated. The back projection is what separated the hand from the rest of the area. We apply a threshold to create a binary image of the back projection. This threshold helps us to separate out the hand from the rest of the frame (c). The next step is to use morphology and blurring techniques to create a proper shape of the hand. Once this step is completed, the process of contouring is executed. This step is crucial because no matter how perfect the histogram is, or how efficient the back-projection algorithm is, there will always be false detection and noise. For this, all the contours are found out for all the images and the largest contour is validated to verify if it matches the profile of a hand or not.

Once the largest contour is found out, the program works only on the largest contour. The next step is to find out the convex hull of the contour. Further, we detect the center of the hands and fingers associated (d).

(c)

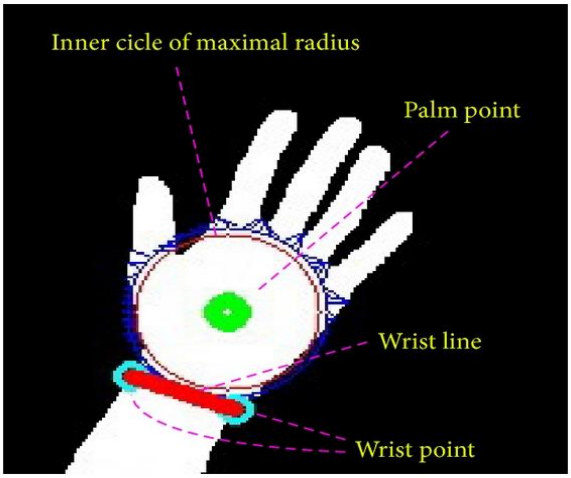


(d)

B. HAND GESTURE RECOGNITION

(I) Hand Palm Segmentation:

The range spanning the largest contour is found out. In the range determined thus, we take every point in the rectangular range and measure the distance from that point to the nearest point on the contour. The point with the largest distance is found out. The point is the center of the largest circle that can be inscribed inside the contour and the distance is the radius of the circle. The point is the center of our hand and the radius is the dimension of our palm. Thus, we have found out the palm center coordinates and radius. The next step is to find the fingers of our hand. For this, we go in a counter clockwise direction and eliminate all the convex hull points that are very near to each other (e).



(e)

(c) Threshold capture (d) Finding the contour (e) Detection of the palm and fingers

(II) Gesture Recognition:

We create a gesture class that has the following attributes –

TABLE I. ATTRIBUTES

S.No	*TYPE*	*Task*
1.	name	A name given to the gesture
2.	Hand_center	Center of the palm
3.	Hand_radius	Radius of the palm
4.	Finger_pos	Array of position of fingers
5.	Finger_count	Total number of fingers
6.	Angle	Angle between each finger and x-axis

A dictionary containing these attributes is used to compare our gestures with.

Since this paper is aimed for hand-driven control, it does not need to recognize a big vocabulary of hand postures. We do not employ any special technique for this particular purpose. Rather, we deduce the hand gesture from extracted information about hand palm, fingertips. An example could be

a hand object with five detected fingers can be classified as the "opening" gesture.

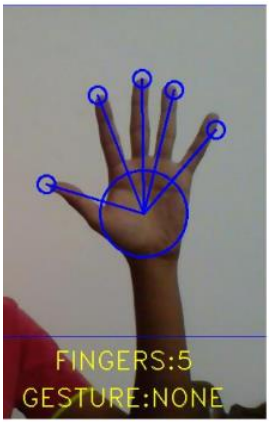
III. EXPERIMENTS

A. METHOD ANALYSIS

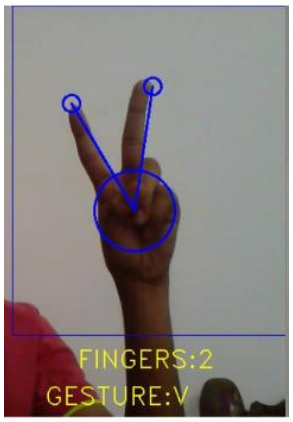
A prototype of the proposed framework is implemented using OpenCV and Python. It can run on a computer starting from 1.86 GHz at a rate of 20 Hz. Upon doing practical experiments, it shows that the framework works well in an environment with little noise and balanced lighting condition.

First, a window appears that shows the camera feed. A rectangular frame on the right side of the window will be noticed. That's the frame where all the detection and recognition work. To begin, the hand and body is kept outside the frame, so as to capture just the background environment, by pressing 'b'. This will capture the background and create a model of it. This model will be used to remove background from every frame captured once the program setup is complete. Now, the hand histogram is captured. The hand is placed over the 9 small boxes in the frame so as to capture the maximum range of shades of the hand. For best results, there must be no shadow or air gap show on the boxed areas. The hand is captured and a histogram is generated by pressing 'c'. The setup is now completed. By keeping the hand inside the rectangular frame, it will get detected and a circle inside the palm area will be noticed, with lines projecting out from it towards your fingers. The hand is moved by hiding a few fingers or giving it one of the sample gestures implemented in the program.

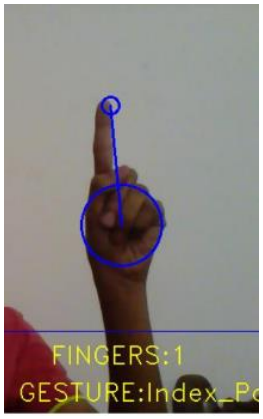
Number of fingers	**Meaning**
1	Index pointing
2	L_right, None, Victory
3	Three
4	Four
5	None



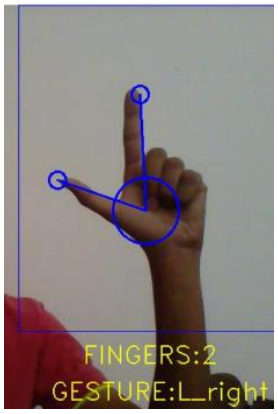
(f)



(g)



(h)



(i)

(j)

B. APPLICATIONS

By using this approach and extracting information, user's gestures are recognized. Different applications can be developed by using the suitable gestures. It comes in very handy as visually impaired people can make use of hand gestures for writing text on MS Office, notepad etc. The strength of this approach includes its simplicity, ease of implementation, and it does not require any significant amount of training or post processing as it provides us with the higher recognition rate with minimum computation time. The weakness of this method is that we define certain parameters and threshold values experimentally and it might not detect properly in improper lighting conditions since it's extremely light sensitive.

Playing games

There are certain racing games such as Need for Speed, that if integrated with hand gesture recognition, would bring a more realistic feel to the experience.

It's a car racing game where the essential part of the game is controlling the steering wheel which can be done by holding a virtual wheel using this technique. The steering wheel can be turned left or right as per the requirement of the gamer. This is done by calculating the angle between two hands with respect to the horizontal axis.

First person shooter games can also be played. Gestures for the following can be set –

1) Look and fire
2) Reloading the bullets
3) Moving forward

Controlling computer cursor

The hand can be detected and its motion is retrieved and interpreted as cursor movement, index finger moving up gesture is interpreted as a mouse left button down event. This interface allows the users to control many programs for example playing games like solitaire, opening and closing files etc.

IX. CONCLUSION AND FUTURE SCOPE

An efficient Human-Computer Interaction framework using skin color detection, tracking and gesture extraction has been presented. In real time, this system is able to interpret many human gestures. We also demonstrate several practical applications as a basic step in developing a real system. In the future, there is likelihood of performance and application implementation using any game. Also, more techniques can be employed, such as hand posture recognition, eye detection and tracking, etc to extract more features helping to deploy more gestures. It will allow us to fully control a computer with more various complicated commands.

REFERENCES

[1] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," Proc. Graphicon, pp. 85–92, Moscow, Russia, September 2003.
[2] C. Huang, B. Wu, and R. Nevatia,"Robust object tracking by hierarchical association detection responses", ECCV 2008, vol. 2, pp. 788–801.
[3] F. Yan, W. Christmas, and J. Kittler , "Layered data association using graph-theoretic formulation with applications to tennis ball tracking in monocular sequence," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, pp. 1814 – 1830, New York, 2008.
[4] T. Starner, A. Pentland , J. Weaver, "Real-time American sign language recognition using desk and wearable computer based video," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20 issue 12, pp. 1371–1375, December 1998.
[5] H. Fillbrandt, S. Akyol, K. F. Kraiss, "Extraction of 3D hand shape and posture from image sequences for sign language recognition." IEEE International Workshop on Analysis and Modeling of Faces and Gestures, vol. 17, pp. 181–186, October 2003.
[6] T. Starner, A. Pentland , J. Weaver, "Real-time American sign language recognition using desk and wearable computer based video," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20 issue 12, pp. 1371–1375, December 1998.
[7] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," IEEE Computer Graphics and Ap
[8] H. Fillbrandt, S. Akyol, K. F. Kraiss, "Extraction of 3D hand shape and posture from image sequences for sign language recognition." IEEE International Workshop on Analysis and Modeling of Faces and Gestures, vol. 17, pp. 181–186, October 2003.