

# Wieldy Finger and Hand Motion Detection for Human Computer Interaction

Prerna Verma, Rohit Miri

**Abstract**---We have developed a gesture based interface for human computer interaction under the research field of computer vision. Earlier system have used the costlier system devices to make an effective interaction with systems, instead we have worked on the web cam based gesture input system. Our goal was to propound lesser cost, wieldy, object detection technique using blobs for detection of fingers. And to give number of count of the same. In addition, we have also implemented the hand gesture recognition.

**Index Terms**—Blob Detection, Center Of Gravity, Computer Vision, Finger Detection, Gesture Recognition, Human Computer Interaction, Web Cam Based.

## I. INTRODUCTION

With the development of information technology in our society, we have seen that computer systems got embedded to a larger extent into our environment. With the use of graphical interfaces like keyboard, mouse, scanners etc, we have evolved out from a text-based command line interfaces to graphical user interface. However, apart from evolution they are still inconvenient and unnatural. The use of hand gestures provides an attractive alternative to these cumbersome interface devices for human-computer interaction (HCI). Human beings are generally more comfortable with their natural hand gestures to express their feelings and intentions, of their thoughts. Human computer interaction basically needs naturalness as well as easiness, which is achievable by visual interpretation of hand gestures. Vision has the potential of carrying a wealth of information in a non-intrusive manner and at a low cost, therefore it constitutes a very attractive sensing modality for developing hand gestures recognition. Recent researches [2], [4], [10], in computer vision have established the importance of gesture recognition systems for the purpose of human computer interaction.

The primary goal of gesture recognition research was to create a system which can identify specific human gestures and use them to convey information or for device control. A gesture may be defined as a physical movement of the hands, arms, face, and body with the intent to convey information or meaning [10]. Gesture recognition, not only includes the recognising part but it also includes the interpretation of semantic meaning of those gestures. Two approaches by [14] were commonly used to interpret gestures for Human Computer interaction.

1) *Methods Which Use Data Gloves*: This method employs sensors (mechanical or optical) attached to a glove that transduces finger flexions into electrical signals for

Prerna Verma, Computer Science Department, Dr.C.V.Raman University, Kargi road, Kota, Bilaspur, India,

Mr. Rohit Miri, Head Of Computer Science Department, Dr.C.V.Raman University, Kargi road, Kota, Bilaspur, India

determining the hand posture. This approach forces the user to carry a load of cables which are connected to the computer and hinders the ease and naturalness of the user interaction.

2) *Methods Which are Vision Based*: Computer vision based techniques were non invasive and were more natural in terms of perceiving, as inputs were taken in similar manner as humans use to gather information by means of eyes. Although it was difficult to design a vision based interface for generic usage, yet it is feasible to design such an interface for a controlled environment.

Hand Gestures can be broadly classified using the following two approaches.

1) *Rule based Approaches*: Rule-based approaches consist of a set of manually encoded rules between feature inputs. Gesture were given as input to the system, which extracts desirable features from input. These features were compared to preencoded rules. The rule that matches the input is reflected as the gesture. As an example, in [14] predicates related to low-level features of the motion of the hands were defined for each of the actions under consideration. When a predicate of a gesture is satisfied over a fixed number of consecutive frames the gesture was returned. A major problem with rule based approaches was that they rely on the ability of a human to encode the rules.

2) *Machine Learning based Approaches*: A popular machine learning approach is to treat a gesture as the output of a stochastic process. Of this class of approach Hidden Markov Models (HMMs) [8] by far have received the most attention in the literature for classifying gestures.

This paper throws some light on previous work through section II. Following this section our main work of algorithm proposal was elaborated under section III. Followed by implementation section IV, then results in section V. And at last conclusion and future scope in section VI

## II. LITERATURE REVIEW

Many researchers have given a firm contribution towards it. It has been a fascinating area in the field of human computer interaction. [4] gave a novel approach to recognize hand gesture using usb webcam or Kinect camera. They divided whole system into three modules: Camera Module, Detection Module, Interface Module. They deployed hybrid hand and finger tracking method, mainly on convexity defects, K-curvature and maximum inscribed circle. While it was a good approach in characterizing hand shape, it was still not perfect because some non-hand objects might possess similar characteristics to the hand. They achieved accuracy upto 50%. [2] described a Kinect-based virtual keyboard using finger gestures. They used Microsoft Kinect motion controller as an input device. Concept was

based on position of the finger if they were on the actual key board. The system scored 75%. [1] discussed a vision-based technique of controlling a robotic hand which has human hand like joints in fingers. The user has to show a gesture to the system with bare hand without any limitation on hand direction and the robotic hand would mimic that gesture. The positions of human hand fingers were calculated using supervised Artificial Neural Network (ANN). They used Kinect camera as input device. The system performed above 90% accuracy.[10] described the advantages of hand gestures being easy to use, natural, and intuitive. They suggested various application of hand gesture input in various domains. They proposed three stages algorithm based on a hand tracking technique to recognize the actual beginning of a gesture using a Kalman filtering process, hidden Markov models and graph matching.[4] presented A gesture-based touch typing keyboard using the Kinect camera. The proposed Gestairboard extends the Gestyboard touchscreen keyboard concept that was suggested in 2011 to overcome exactly that problem, thus adapting it for devices without a touchscreen. It uses inexpensive hardware and to the best of our knowledge was the first implementation of a touch typing keyboard using Kinect. The prototype was compared to the Gestyboard and to a physical keyboard and evaluated in terms of performance, reliability, learnability and user acceptance.[5] presented Hand Tracking And Guesture Recognition. The aim of the project was to device a program that was able to detect out hands, track them in realtime and perform some guesture recognition. It was do be done with simple signal processing performed on images obtained from a regular laptop web-camera. Thus we can detect the no of fingers, the location of palm and its depth.Using this we can construct systems that detect gestures. Due to the heavy noise involved in regular camers, the location of the palm and the radius are very jittery and varies continuously, but the average is the same and is suitable for use. Thus we need to average it over a few frames. Thus rapid guestures were not suited for the system that is built. Also the background needs to be stationary (slight to moderate ) motion is tolerable due to the filtering techniques used). The hand needs to be facing straigt at the camera, as tilts cause the palm to obtain elliptical shape which was not well detected by the system. The system was very well suited for performing guestures like pause , play ,grab, drop, guestures based on finger location,etc.[10] designed a three phase gesture recognition systems. The phases were Image Processing ,Tracking,Recognition.The recognition was based on HAAR classifier. The system performed well with few limitation posed by noise and blur motion. [1] presented a new method for the detection of fingertips in a closed hand using the corner detection method and an advanced edge detection algorithm. The algorithm has five stage as following Input RGB Image,Color based Segmentation,Edge detection,Corner detection and Fingertip localization,Fingertips Detection.The accuracy wachieved upto 60%.[2] presented a novel method forfingertips detection and centres of palms detection distinctly for both hands using MS KINECT in 3D from the input image. The whole system was implemented in real time. The accuracy for fingertips detection, when all finger were open was near to 100% while in fully bended fingers case sometimes it was confused. In the case of centre of palm detection, the results were around 90% correct. [7] presented a vision-based

system for touch-free interaction with a display at a distance. A single camera is fixed on top of the screen and is pointing towards the user. An attention mechanism allows the user to start the interaction and control a screen pointer by moving their hand in a fist pose directed at the camera.[8] introduced two faster techniques to detect skin color. The skin color detection is important for background subtraction.[9] presented a color based method for hand location and fingertip location was done using contour analysis. The finger recognition gesture worked fine for a 99% of the cases.

### III. HAND GESTURE RECOGNITION

#### A. Architecture

initially Background image was captured.Then frames containing our palm region as an object were captured.Our proposed background elimination algorithm subtracts the background from foreground image frames, by comparison of the frames and setup of thershold.This image was then operated upon by gray scale conversion.After which blob with biggest area was searched by application of proposed blob detection algorithm.Distance between Center of gravity and edge contour of blob hence,gave us the maxima and minima values.By which number of fingers were counted upon.After these steps finally motion of hands were detected by localisation of the corner pixels of box bounding the hands region.The major architecture of the whole procedure can be shown as

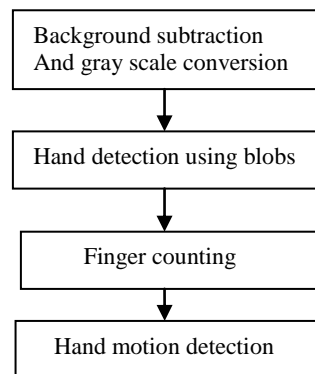


Figure 3.1-Methodology architecture

- 1) *Background Elimination:*Background are those objects which shows a gradual changes in pixels during human computer interaction.To mark those background we have to take image frames at regular interval and matching operation was performed on them to find out, if there pixel values got changed or not.If change in pixel values were recorded considerably, then it was identified as foreground otherwise it was considered to be the background.Two images  $I_1$  and  $I_2$  taken at some interval of time,could be defined as a set of pixels  $I_1 = \{p_{11}, p_{12}, p_{13}, \dots, p_{1n}, \}$  and  $I_2 = \{p_{21}, p_{22}, p_{23}, \dots, p_{2n}, \}$  where  $p_{1i}$  is  $i$ th pixel in the image  $I_1$ ,  $p_{2i}$  is  $i$ th pixel in the image  $I_2$ ,  $i = 1, 2, 3, \dots, n$  and  $n$  is the number of pixels in the image.Then algorithm of background elimination is as follows-

//BackgroundElimination( $I_1, I_2$ )

- 1)  $n \leftarrow \text{lengthOf}(I_1)$

- 2) threshold  $\leftarrow \alpha$  is bench mark to determine if pixel value has changed considerably
- 3)  $B[n] \leftarrow \phi$
- 4) **for**  $i \leftarrow 1$  to  $n$
- 5)     **if**  $|p_{1i} - p_{2i}| > \text{threshold}$  then
- 6)          $B[i] \leftarrow 0$
- 7)     **else**
- 8)          $B[i] \leftarrow 1$
- 9)     **end if**
- 10) **end for**
- 11) return B.

An Array of output values B(i), indicating 0 for foreground and 1 for background.

- 2) *Blob Detection Of Hand Region:* After the background subtraction extracted image contains several images in foreground regions among which all the objects were not our palm region. They could be either our face or any other object in motion. But the only biggest blob was of the palm. Hence, we needed to search only for the biggest blob inside the image frame. This was accomplished by proposed algorithm. Set of Blobs  $B = \{b_1, b_2, b_3, \dots, b_n\}$  were taken as input, where  $b_i$  was  $i^{\text{th}}$  blob in the set and  $i = 1, 2, 3, \dots, n$  where  $n$  was the number of blobs in set. Algorithm for finding biggest blob of hand is as follows-

//HandBlob( B[ ] )

- 1)  $n \leftarrow \text{lengthOf}(B)$
- 2) hand blob  $\leftarrow \text{null}$
- 3) biggest blob size  $\leftarrow 0$
- 4) **for**  $i \leftarrow 1$  to  $n$
- 5)     current blob size  $\leftarrow \text{sizeOf}(B[i])$
- 6)     **if** biggest blob size  $<$  current blob size
- 7)         **then**
- 8)             hand blob  $\leftarrow B[i]$
- 9)             biggest blob size  $\leftarrow$  current blob size
- 10)     **end if**
- 11) **end for**
- 12) return hand\_blob

After the analysis of all the blobs largest blob was predicted as output of the proposed algorithm. Hence hand\_blob corresponding to hand region is the output.

- 3) *Number Of Fingers Detection:* For counting purpose of number of fingers from hand blob. First center of gravity of the hand was calculated on the blob after application of above algorithm. From this center of gravity, distances of maxima and minima in the edge contour of the hand blob were evaluated. Maxima was the highest point and Minima was the lowest point on the edge of finger from the center of gravity of hand. Algorithm for counting number of fingers is as follows:

//FingerTipDetection( H )

- 1) Edge of the blob  $E$  is defined as set of pixel in edge,  $E[ ] \leftarrow \text{edgeOfBlob}(H)$
- 2)  $n \leftarrow \text{lengthOf}(E)$
- 3)  $\text{max} \leftarrow 0$
- 4)  $\text{min} \leftarrow 0$
- 5) threshold  $\leftarrow \alpha$
- 6)  $N$  number of finger  $\leftarrow 0$
- 7)  $F \leftarrow \phi$
- 8)  $c \leftarrow \text{centerOfGravity}(H)$
- 9)  $\text{max} \leftarrow \text{distance}(c, H[1])$
- 10) **for**  $i \leftarrow 2$  to  $n$
- 11)      $d \leftarrow \text{distance}(c, H[i])$
- 12)     **if**  $|\text{max} - d| > \text{threshold}$  AND  $\text{max} > d$  **then**
- 13)          $\text{min} \leftarrow d$
- 14)          $F \cap H[i]$
- 15)     **else if**  $|\text{min} - d| > \text{threshold}$  AND  $\text{min} < d$  **then**
- 16)          $\text{max} \leftarrow d$
- 17)     **end if**
- 18) **end for**
- 19) return  $N$  and  $F$ .

$N$  is the Number Number of finger detected and  $F$  is the set of co-ordinate of finger tip =  $\{f_1, f_2, f_3, \dots, f_N\}$  where,  $f_i$  is coordinate of tip of  $i^{\text{th}}$  finger detected, and  $i = 1, 2, 3, \dots, N$ .

- 4) *Hand Motion Detection :* To detect motion of the hand we monitor corner pixels of bounding box which encapsulates the hand blob. And motion of hand was monitored by direction of motion of the corner diagonal coordinates of bounding box. Using the detection of hand motion algorithm we have developed an implementation for panning of system screen to left right and so on. Hand blobs taken at different interval  $H_i$  and  $H_j$  where,  $H_i$  is hand blob recorded at time  $i$  and  $H_j$  is hand blob recorded at time  $j$ . These were the inputs for our proposed algorithm. Origin of coordinate is situated at leftmost bottom corner of the window.

//HandMotion(  $H_1, H_2$  )

- 1)  $B_1 \leftarrow \text{boundinBoxOf}(H_1)$
- 2)  $x_{l1}$  x coordinate of left bottom most corner of  $B_1$
- 3)  $y_{l1}$  y coordinate of left bottom most corner of  $B_1$
- 4)  $x_{r1}$  x coordinate of right top most corner of  $B_1$
- 5)  $y_{r1}$  y coordinate of right top most corner of  $B_1$
- 6)  $B_2 \leftarrow \text{boundinBoxOf}(H_2)$
- 7)  $x_{l2}$  x coordinate of left bottom most corner of  $B_2$
- 8)  $y_{l2}$  y coordinate of left bottom most corner of  $B_2$
- 9)  $x_{r2}$  x coordinate of right top most corner of  $B_2$
- 10)  $y_{r2}$  y coordinate of right top most corner of  $B_2$
- 11) threshold  $\leftarrow \alpha$
- 12)  $D \leftarrow N$  //Motion of hands were initialised to // no hand motion  $N$ .
- 13) **if**  $x_{l1} > x_{l2}$  And  $|x_{l1} - x_{l2}| > \text{threshold}$  And  $|y_{l1} - y_{l2}| \leq \text{threshold}$  **then**
- 14)      $D \leftarrow L$  // motion towards left
- 15) **end if**

```

16) if  $x_{11} > x_{12}$  And  $|x_{11} - x_{12}| > \text{threshold}$  And
     $|y_{11} - y_{12}| < \text{threshold}$  then
17)     D ← R // motion towards right
18) end if
19) if  $|x_{11} - x_{12}| < \text{threshold}$  And  $y_{11} < y_{12}$  And
     $|y_{11} - y_{12}| > \text{threshold}$  then
20)     D ← U // motion towards up
21) end if
22) if  $|x_{11} - x_{12}| < \text{threshold}$  And  $y_{11} > y_{12}$  And
     $|y_{11} - y_{12}| > \text{threshold}$  then
23)     D ← D // motion towards down
24) end if
25) if  $x_{11} > x_{12}$  And  $|x_{11} - x_{12}| > \text{threshold}$  And  $y_{11} < y_{12}$ 
    And  $|y_{11} - y_{12}| > \text{threshold}$  then
26)     D ← LU // motion towards left up
27) end if
28) if  $x_{11} > x_{12}$  And  $|x_{11} - x_{12}| > \text{threshold}$  And  $y_{11} > y_{12}$ 
    And  $|y_{11} - y_{12}| > \text{threshold}$  then
29)     D ← LD // motion towards left down
30) end if
31) if  $x_{11} < x_{12}$  And  $|x_{11} - x_{12}| > \text{threshold}$  And  $y_{11} < y_{12}$ 
    And  $|y_{11} - y_{12}| > \text{threshold}$  then
32)     D ← RU //motion towards right up
33) end if
34) if  $x_{11} < x_{12}$  And  $|x_{11} - x_{12}| > \text{threshold}$  And  $y_{11} > y_{12}$ 
    And  $|y_{11} - y_{12}| > \text{threshold}$  then
35)     D ← RD //motion towards right down
36) end if
37) return .
    
```

Motion of hands in the directions proposed by our algorithm was output D, resulting output indicating L,R,U,D,LU,LD,RU,RD,N for left,right,up,down,left up,left down,right up,right down,no motion respectively.

#### IV. IMPLEMENTATION

The implementation of hand recognition and finger counting for human computer interaction was developed on our Lenovo G 580 laptop having Processor Intel Core i3-2348M CPU@2.30GHzX4, RAM :3.8 GiB, OS type: 32-bit. It contains Linux,with built-in webcam was used for input and processing vision interface module.

Frame images were captured from built-in webcam.Vision module was developed using Open Processing ver.2.2.1 (Processing, <https://www.processing.org>) library.Mainly vedio and blobscanner libraries were used.We have used java based coding to develop the whole program.

The constant parameters used in this paper are as follows. The resolution of camera input is 640x480 pixels.Frame rate was 30 frames/sec.The thershold value used was 190.Tip\_Mass and Hand\_Mass were 45 and 600 respectively.

#### V. RESULTS

Image processed by the proposed algorithms are shown by below figures:

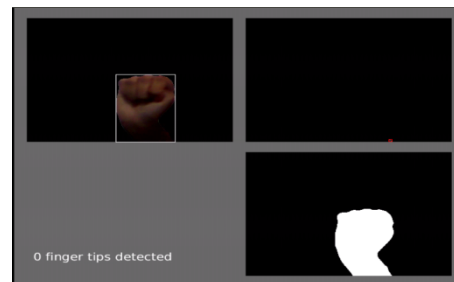


Fig.5.1:No fingers zero count by developed program

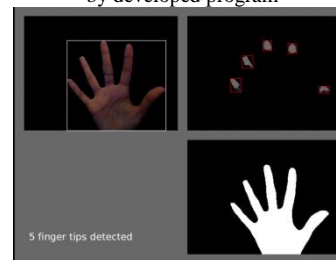


Fig.5.2:Count of Five displayed by developed program

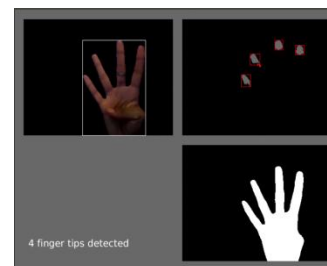


Fig.5.3:Program gave exact count of four fingers

In the figure 5.1,figure 5.2,figure 5.3,leftmost sub window displays the number of count of fingers.Leftmost Top sub-window showing the current frame bieng captured by web cam.Rightmost top window represents implementation of finger detection algorithm and rightmost below subwindow shows the hand blobs after gray scale image conversion.When Hand motion detection algorithm implemented for window paning of demo web pages it resulted to run correctly.Hence, after application of above proposed algorithms meticulousness rate of the program developed by proposed algorithm was about to 89% in sufficient light and proper background.

The system worked accurately under sufficient lighting condition.While for dimmer lights due to low resolution of webcam, system performance was low.In addition to this, a clear monocolour plain background was expected because presence of too many blobs in background can make process of hand blob detection difficult.

#### VI. CONCLUSIONS AND FUTURE WORK

In this paper, a cost reduced,wieldy, and effective module, which can work with low resolution cameras of systems was developed.The proposed algorithmic modules were able to detect motion of hand and count the number of fingers in sufficient expected environment.The proposed module was able to work accurately for object placed at position of about 30-50 cm from webcam.Using the program by following the

proposed algorithms, simple application of panning window screen has been deployed successfully.

For future research, Depth calculation of object from capturing devices with more robust module for calculation of interest of region could be developed, which could work accurately with inferior light intensity for human computer interaction. In addition, project can be deployed in large scenario by making personal APIs and libraries.

## REFERENCES

- [1] Ankit Chaudhary , J.L. Raheja , Karen Das , and Sonia Raheja, A Survey on Hand Gesture Recognition in Context of Soft Computing ,2011.
- [2] Ankit Chaudhary, J. L. Raheja, Karen Das, Sonia Raheja, Intelligent Approaches to interact with Machines using Hand Gesture Recognition in Natural way: A Survey, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.1, Feb 2011
- [3] Hasup Le Lee, Yoshisuke Tateyama, and Tetsuro Ogi , Hand Gesture Recognition using Blob Detection for Immersive Projection Display System ,2012.
- [4] Hasan, M. M., & Mishra, P. K., "Gesture recognition using modified HSV segmentation", IEEE International Conference on Communication Systems and Network Technologies, 2011.
- [5] Johannes Roith, Tayfur Coskun, Teodora Velikova, Sebastian Klingenberg, Gudrun Klinker, Gestairboard: A gesture-based touch typing keyboard using the Kinect camera
- [6] Lakshminarayanan (s-lin.in), Hand Tracking And Gesture Recognition .
- [7] M. S. Yang and N. Ahuja, "Detecting human faces in color images," in Proc. IEEE Int. Conf. Image Process., IL, Oct. 1998, pp. 127–130.
- [8] Pedro Trindade, Jorge Lobo, and Joao P. Barreto, Hand gesture recognition using color and depth images enhanced with hand angular pose data, 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)
- [9] Reza Oji, AN AUTOMATIC ALGORITHM FOR OBJECT RECOGNITION AND DETECTION BASED ON ASIFT .KEYPOINTS, Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.5, October 2012.
- [10] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, 2014
- [11] Rama B. Dan , P. S. Mohod, Survey on Hand Gesture Recognition Approaches .
- [12] Sung Kwan Kang, Mi Young Nam , Phill Kyu Rhee, Color Based Hand and Finger Detection Technology for User Interaction , International Conference on Convergence and Hybrid Information Technology 2008
- [13] Sushmita Mitra, Tinku Acharya, Gesture Recognition: A Survey , IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, VOL. 37, NO. 3, MAY 2007
- [14] S. Phung, A. Bouzerdoum and D. Chai, "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.27, No. 1, 2005, pp. 148-154.
- [15] Vishal Nayakwadi, N.B. Pokale, Dynamic Hand Gestures Recognition System with Natural Hand, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 4, April 2014.
- [16] V. Vezhnevets, V. Sazonov and A. Andreeva, " A Survey on Pixel-Based Skin color Detection Techniques", In Proceedings of Graphicon, 2003, pp. 85-92.
- [17] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Trans. PAMI, 24, 2002.