

Hand Gesture Movement Tracking System for Human Computer Interaction

Shital M. Chavan¹, Smitha Raveendran²

¹ M.E. Student, Electronics, Ramrao Adik Institute of Technology, Maharashtra, India

² Assistant Professor, Electronics, Ramrao Adik Institute of Technology, Maharashtra, India

Abstract- Hand gesture recognition is a natural and easy way to interact with the computer, since interactions with the computer can be increased through multidimensional use of hand gestures as compare to other input methods. This paper represents an approach for Human computer Interaction (HCI), where we introduce a new application to control the mouse cursor movement and click events using various hand gestures. Direct use of hands as an input device is an innovative method for providing natural Human Computer Interaction. The objective of this paper is making a system prototype that can track the movements of the hand as well as to recognize the meaning of simple hand gestures command as a replacement of the mouse device. It focuses on the three main phases of hand gesture recognition i.e. detection, tracking and recognition. The system presented here uses a web camera and algorithms which are developed using computer vision, image and the video processing toolboxes of MATLAB. In this paper skin detections method is used to create a segmented hand image and to differentiate with the background. Segmented hand image is further processed and used to track mouse pointer with the help of JAVA class. This technology has wide applications in the fields of computer graphics, computer gaming, prosthetics, robot navigation, augmented reality and biomedical instrumentation. Moreover, this hand gesture recognition technique is build to substitute the use of mouse for interaction with the virtual objects. In general, the system can detect and follow the hand movement so that can be used as user interface in real time.

Key Words: Human computer Interaction (HCI), Hand gesture, virtual environment, tracking, recognition.

1. INTRODUCTION

Hand gesture recognition is one of the most active areas of research in computer vision. It provides an easy way to interact with a machine without using any extra devices. If the user does not have much technical knowledge about the system then human-computer interaction (HCI) enables the user to use the system without any difficulty

with their normal hands [1]. The development of user interfaces has increased rapidly, especially in consumer devices. The use of the touch interface (e.g. touch screen) has common in today devices such as computer tablets, Smart Phones, and other control interface on an electronic device. The development of user interface are tends to the use of human body gesture, especially human hand as the main interface, which is more natural and intuitive.

The use of hand gestures provides an attractive and natural alternative to these cumbersome interface devices for human computer interaction [2]. Using hands as a device can help people communicate with computers in a more intuitive way. When we interact with other people, our hand movements play an important role and the information they convey is very rich in many ways. We use our hands for pointing at a person or at an object, conveying information about space, shape and temporal characteristics. We constantly use our hands to interact with objects: move them, modify them, and transform them. In the same unconscious way, we gesticulate while speaking to communicate ideas (stop, come closer, no, etc). Hand movements are thus a mean of non-verbal communication, ranging from simple actions (pointing at objects for example) to more complex ones (such as expressing feelings or communicating with others) [2]. In this sense, gestures are not only an ornament of spoken language, but are essential components of the language generation process itself. A gesture can be defined as a physical movement of the hands, arms, face and body with the intent to convey information or meaning [2].

Gesture is nothing but the meaning of statement said by the human being. They come naturally with the words to help the receiver to understand the communication [3]. Recent researches 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 is to create a system which can identify specific human gestures and use them to convey information or for device control. [4]

Our application presents a more effective and user friendly methods of human computer interaction intelligently with the usage of hand gestures. In this work, the hand gesture motion detection and recognition will be

implemented to replace the mouse function [5]. We use a camera, MATLAB Image Processing Toolbox to control the mouse movement and implement the finger tracking. To get the proper gesture recognition various processes can be performed on each frame of the video such as segmentation, Background removal, Noise removal and Thresholding etc.

2. LITERATURE SURVEY

Two approaches are commonly used to interpret gestures for Human Computer interaction. They are

- *Contact based*: This method employs sensors (mechanical or optical) attached to a glove that transduces finger flexions into electrical signals for determining the hand posture [4]. Contact based devices are based on physical interaction of user with the interfacing device i.e. the user needs to be accustomed with the usage of these devices. These devices are usually based on technologies like data glove, accelerometers, multi-touch screen etc which uses several detectors [2].
- *Vision Based*: Computer vision based techniques are non invasive and based on the way human beings perceive information about their surroundings. Although it is difficult to design a vision based interface for generic usage, yet it is feasible to design such an interface for a controlled environment. This approach is easier as the person doesn't need to wear hardware [4].

2.1 Vision based Hand Gesture Representations

In earlier days hand gesture detection was done using mechanical devices to obtain information of the hand gesture. One of the most widely used and accepted examples for hand gestures recognition is data glove. Enhancement of gesture recognition has replaced the role of data gloves to non wearable devices due to its naturalness without using any device. This is quite user friendly in human computer interaction. One of the major drawbacks of data glove is that it is cumbersome with the limitation of hand movement [6]. Also due to contact based devices it is the health hazards, which are caused by its devices like mechanical sensor material which raises symptoms of allergy, magnetic devices which raises risk of cancer etc. Whereas vision based devices have initial challenge of complex configuration and implementations but are more user friendly and hence more privileged for usage in long run [2].

Further the vision based hand gesture recognition system is modeled as

- 3D Model based
- Appearance based techniques

Figure 1 shows the classification of Vision based hand gesture recognition system. The 3D model based hand gesture recognition has different techniques for gesture representation namely 3D textured volumetric, 3D geometric model and 3D skeleton model. Appearance based hand gesture representation include color based model, silhouette geometry model, deformable gabarit model and motion based model.

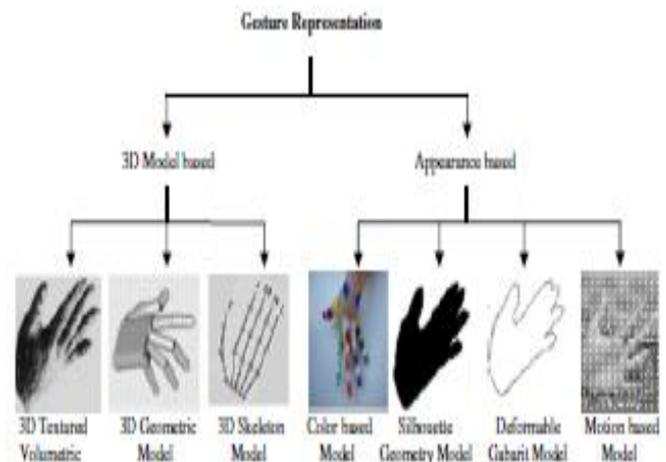


Fig-1: Vision based Hand Gesture Representations [2]

2.1.1 3D model based

3D model based methods make use of 3D information of key elements of the body parts. Using this information, several important parameters, like palm position, joint angles etc., can be obtained. This approach uses volumetric or skeletal models, or a combination of the two. In computer animation industry and for computer vision purposes, volumetric approach is better suited. This approach is very computational intensive and also, systems for live analysis are still to be developed [4].

2.1.2 Appearance based

Appearance-based systems use images or videos as inputs. They directly interpret from these videos/images. They don't use a spatial representation of the body. The parameters are derived directly from the images or videos using a template database. Some templates are deformable 2D templates of the human parts of the body, particularly hands. Deformable templates are sets of points on the outline of an object, used as approximation [4].

- *Color based model* uses body markers to track the motion of body or body part. The hand gesture recognition employs multiscale color features, hierarchal models and particle filtering [4].
- *Silhouette geometry based models* include several geometric properties of the silhouette such as

perimeter, convexity surface, bounding box/ellipse, elongation, rectangularity, centroid and orientation. The geometric properties of the bounding box of the hand skin were used to recognize hand gestures [4].

- *Deformable gabarit based models*: they are generally based on deformable active contours (i.e. snake parameterized with motion and their variants used snakes for the analysis of gestures and actions in technical talks for video indexing [4].
- *Motion based models* are used for recognition of an object or its motion based on the motion of object in an image sequence. Local motion histogram was introduced which uses an Ad boost framework for learning action models. Recognizing gesture is a complex task which involves many aspects such as object detection, object description, motion modeling, motion analysis, pattern recognition and machine learning even psycholinguistic studies also required [4].

- Detection of hand and segmentation of image.
- Hand Tracking.
- Gesture recognition.

2.2 Vision based hand gesture recognition techniques.

Vision based hand gesture recognition techniques by categorized under the three fundamental phases of detection tracking and recognition as shown in Fig.2.

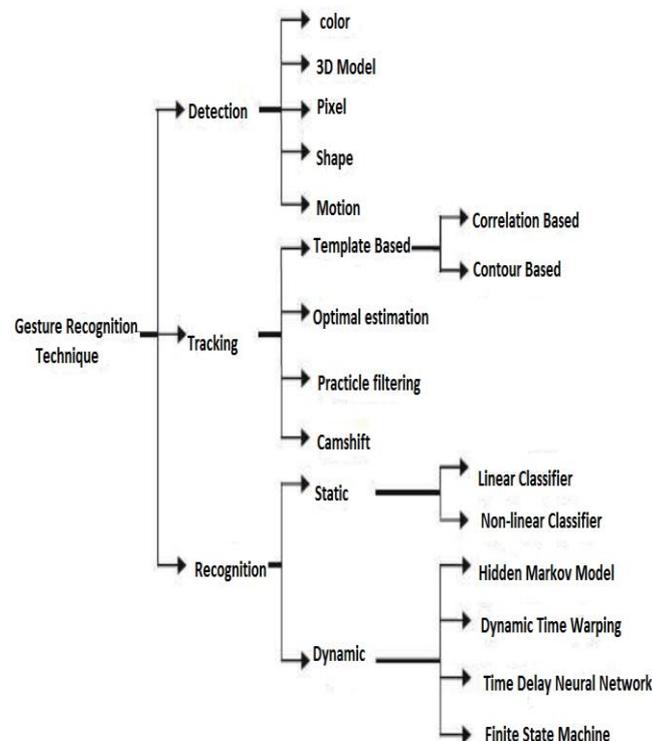


Fig-2: Vision based hand gesture recognition techniques [2]

2.2.1 Detection of hand and segmentation of image

The primary step in hand gesture recognition systems is the detection of hands and the segmentation of the corresponding image regions. This segmentation is crucial because it isolates the task-relevant data from the image background, before passing them to the subsequent tracking and recognition stages. Detection of hand gestures and segmentation is done on the basis of Color, Shape, Motion, Pixel and 3D model [2].

2.2.2 Hand Tracking

If the detection method is fast enough to operate at image acquisition frame rate, it can be used for tracking as well. However, tracking hands is difficult since they can move very fast and their appearance can change vastly within a few frames. Tracking can be defined as the frame-to-frame correspondence of the segmented hand regions or features towards understanding the observed hand movements [2].

2.2.3 Gesture recognition

The overall goal of hand gesture recognition is the interpretation of the semantics that the hand location, posture, or gesture conveys. Vision based hand gesture recognition techniques can be further classified under static and dynamic gestures. To detect static gestures (i.e. postures), a general classifier or a template-matcher can be used. However, dynamic hand gestures require techniques that handle this dimension like Hidden Markov Models [2].

3. APPLICATIONS

Hand Gesture recognition has a wide range of application in real life through computer vision.

1. Some advanced applications include tablet PC, games, medicine environment, and augmented reality [4].
2. In desktop applications, hand gestures can offer a substitute interaction medium for mouse and keyboard. Many hand gestures for desktop computing tasks involve manipulating graphic objects or annotating and editing documents using pen-based gestures. Also use of pen gestures, make marking menu selections using stroke gestures [4].
3. Interacting with a virtual environment: The user feels, he/she is actually interacting with the real object, while interacting with the digital objects inside computer system. With the help of Virtual environment, we create

virtual games, driving simulation, animation movies etc [7].

4. Robotics and Tele-presence: Tele-presence and tele-robotic applications are typically situated within the domain of space exploration and military-based research projects [4].

5. In Sign Language Recognition [4].

4. ANALYSIS PARAMETERS

In order to find out the performance and viability of the proposed gesture recognition system following testing and analysis parameters could be considered

- **Robustness:** In the real-world, visual information could be very rich, noisy, and incomplete, due to changing illumination, clutter and dynamic backgrounds, occlusion, etc. Vision-based systems should be user independent and robust against all these factors [6].
- **Scalability:** The Vision-based interaction system should be easily adapted to different scales of applications. For e.g. the core of Vision-based interaction should be the same for desktop environments, Sign Language Recognition, robot navigation [6].
- **Computational Efficiency:** Generally, Vision based interaction often requires real time systems. The vision and learning techniques/algorithms used in Vision-based interaction should be effective as well as cost efficient [6].
- **Users Tolerance:** The malfunctions or mistakes of Vision-based interaction should be tolerated. When a mistake is made, it should not incur much loss. Users can be asked to repeat some actions, instead of letting the computer make more wrong decisions [6].

5. SYSTEM MODEL

The Basic block diagram of the system is as shown in fig.3 starting by capturing real time video using webcam, converting the video into image frames and processing them. The detection of hand image is accomplished by extraction of skin colors from image and determines the hand center point. Performing the different mouse actions by tracking the mouse pointer using the coordinates obtained from the centroid.

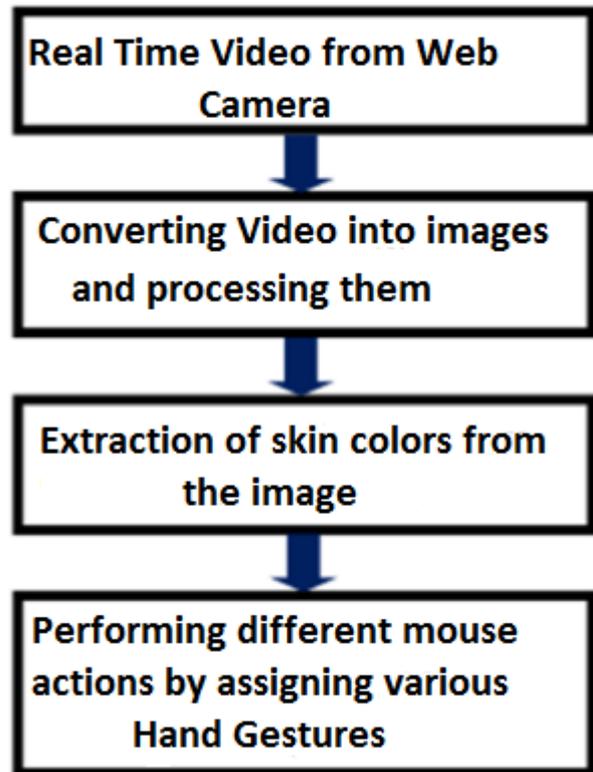


Fig-3: Basic block diagram of the system

Following are the steps in our approach:

1. Capturing real time video using Web-Camera.
2. Processing the individual image frame.
3. Conversion of each frame to a grey scale image.
4. Color detection and skin color extraction.
5. Apply median filter to remove noise.
6. Conversion of the detected image into a binary scale image.
7. Removing all the small areas.
8. Finding the largest region of the image and calculating its centroid.
9. Tracking the mouse pointer using the coordinates obtained from the centroid.
10. Simulating the mouse click events.

5.1 Capturing the video Frame

For the system to work we need a sensor to detect the hand movements of the user. The webcam of the computer is used as a sensor. The webcam captures the real time video at a fixed frame rate and resolution which is determined by the hardware of the camera. The frame rate and resolution can be changed in the system if required [8].

In order to initialize the system user have to wave his/her hand in front of webcam. The image acquisition toolbox of MATLAB acquires the output of the webcam in terms of

frames. The function getsnapshot is used for grabbing the frame.

- frame = getsnapshot(obj)

The video is divided into Image frames based on the FPS (Frames per second) of the camera. Each frame is processed individually.

5.2 Conversion of RGB Image into Gray Image

The main reason why grayscale representations are used, for extracting descriptors instead of operating on color images directly is that grayscale simplifies the algorithm and reduces computational requirements. Indeed, color may be of limited benefit in many applications and introducing unnecessary information could increase the amount of training data required to achieve good performance [9].

As compared to a colored image, computational complexity is reduced in a gray scale image. Thus the RGB image is converted into a gray scale image. All the necessary operations were performed after converting the image into gray scale [8]. Fig.4 shows original image and converted gray image.



Fig-4: (a) RGB image (b) gray image

5.3 Color Detection and Skin color Extraction

Skin color is considered to be a useful and discriminating image feature for tracking. In addition, the color of skin may vary among people. Furthermore, for the same person the skin color differs significantly both in place (i.e. face versus hands) and in time (i.e. after long sun exposure). Numerous techniques for skin color modeling and detection have been proposed [11].

Skin color segmentation is a technique of discrimination between skin and non-skin pixels of an image. In order to segment human skin regions from non-skin regions, a reliable skin model is needed who is adaptable to different colors and light conditions. Here implementation and extraction of skin pixels in RGB color model is being presented. The presence of light, shadows, noise etc can affect the appearance of the skin color. However an effective skin segmentation algorithm should be capable to

detect skin pixels efficiently by overriding these effects. In this work, a RGB based skin segmentation technique is used for extraction of skin pixels.

The elimination of background from hand image is obtained using skin color segmentation algorithm. This is the most important step in the whole process. The red, green and blue color object is detected and extracted from the Gray-Scale Image. This creates an image which contains the detected object as a patch of grey surrounded by black space.

5.4 Applying median filter to remove noise

The Median filter is non-linear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing [10]. In median filtering each output pixel contains the median value in the m-by-n neighborhood around the corresponding pixel in the input image. Fig.5 shows Filtered image of gray scale image.



Fig-5: Filtered image

5.5 Conversion of gray scale Image into Binary scale Image

To convert the gray scale image to binary scale image MATLAB's built in "im2bw" function has been used. Function can be used as;

BW = im2bw(I, level)

Where; it converts the grayscale image I to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). Specify level in the range [0,1]. This range is relative to the signal levels possible for the image's class. Therefore, a level value of 0.5 is midway between black and white, regardless of class [12].

Thus the resultant image obtained is a monochromatic image consisting of only black and white colors. The conversion to binary is required because MATLAB can only find the properties of a monochromatic image [8]. Fig.6 shows conversion of Gray scale image to Binary image.



Fig-6: Binary image

5.6 Finding Centroid of an object

For the user to control the mouse pointer it is necessary to determine a point whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. An in built function in MATLAB is used to find the centroid of the detected region [8].

'Centroid': 1-by-Q vector that specifies the center of mass of the region. The first element of Centroid is the horizontal coordinate (or x-coordinate) of the center of mass, and the second element is the vertical coordinate (or y-coordinate). All other elements of Centroid are in order of dimension [13]. These coordinates change with time as the object moves across the screen.

To get the properties of the region such as center point or bounding box etc., MATLAB's built in "regionprops" function can be used as;

```
STATS = regionprops(BW, properties)
```

Where; it measures a set of properties for each connected component (object) in the binary image, BW. The image BW is a logical array; it can have any dimension. In our application since we need to get the center of the object, we used only this property [13].

In order to find the fingertips of hand we crop the image, so the image is divided into different objects. After removing all the connected components (objects) other than the pointer, using MATLAB's "bwlabel" function the pointer can be labeled. In other words the region can be detected and we can recognize number of fingers.

5.7 Tracking the Mouse pointer

Once the coordinates has been determined, the mouse driver is accessed and the coordinates are sent to the cursor. With these coordinates, the cursor places itself in the required position. It is assumed that the object moves continuously, each time a new centroid is determined and for each frame the cursor obtains a new position, thus creating an effect of tracking. So as the user moves his hands across the field of view of the camera, the mouse moves proportionally across the screen. There is no in built function in MATLAB which can directly access the

mouse drivers of the computer. But MATLAB code supports integration with other languages like C, C++, and JAVA. Since java is a machine independent language so it is preferred over the others. A java object is created and it is linked with the mouse drivers [8].

The ability to control the mouse pointer position, motion and clicks is not available in MATLAB. To work around this issue, we use the java class java.awt.Robot which has this ability. Based on detection of hand system performs the clicking events of the mouse. Since the resolution of the camera is smaller than the resolution of the computer monitor, the (X, Y) coordinates are needed to be scaled. In our application the resolution of the input image was 640x480 and the resolution of the computer monitor was 1280x800. So, we scaled the (X, Y) coordinates of the pointer as to be able to move the cursor along all window.

5.8 Performing Clicking Actions

To control mouse position and motion, Centroid tracking is used. The centroid of binary hand is passed to class mouse.mouseMove(x,y). Hence the mouse pointer is moved according to the centroid motion. Other tasks are performed by identifying the different gestures in the current frame. If the system does not detect any finger (fist), it will only track the hand, and when it detects fingers it performs click function [1].

The control actions of the mouse are performed by controlling the flags associated with the mouse buttons. JAVA is used to access these flags. The user has to perform hand gestures in order to create the control actions [8]. First we import the java class to MATLAB, create an object of this type, and then execute the mouseMove method in a loop to simulate motion. Again, import the required java classes, create an object of this type, and then use the mousePress and mouseRelease functions to simulate a click [14].

The class MouseEvent has access getX() and getY() that get us the coordinates of where the event occurred.

- mousePressed(), which is triggered when the mouse is on the component.
- mouseReleased(), which is triggered when the mouse is released on the component, (mousePressed() must have been called first)

To implement mouse move, right and left click, zoom in and zoom out function, we use the following mouse event functions.

1. Move; is implemented using mouse.mouseMove.
2. Right click and Left click; are implemented using mouse.mousePress and with mouse.mouseRelease, where mouse.mousePress is call first.

3. Zoom in and zoom out; are implemented using mouse.mouseWheel.

The applied mouse function according to the number of fingers detected as shown in Fig.8.

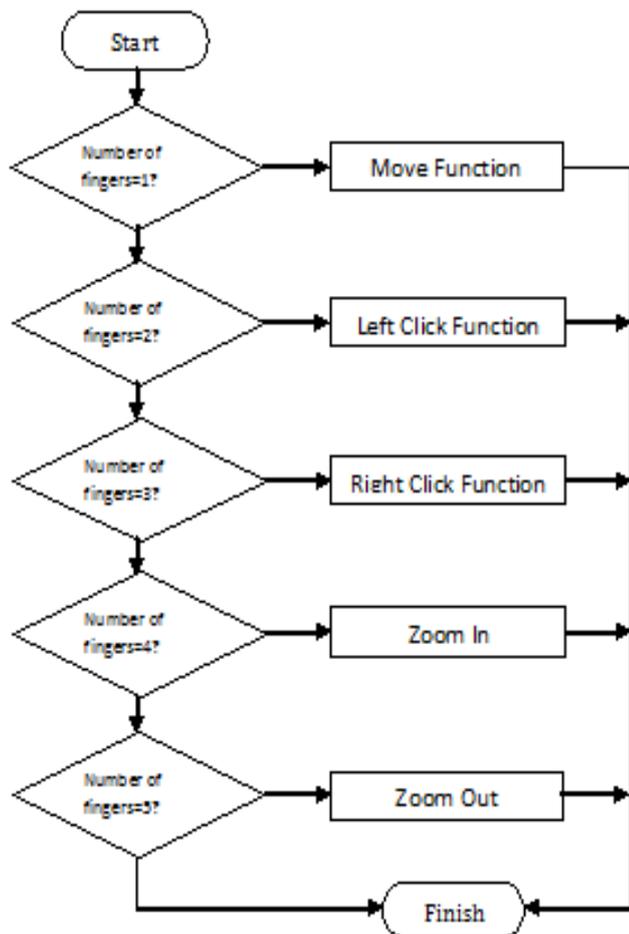


Fig-8: Mouse function implementation.

8. RESULT

The system is developed in MATLAB which has ability to support JAVA. So we import the JAVA class to MATLAB to access the mouse drivers. Fig.9 shows all processes of hand detection. After that finally detected hand image is used for tracking. This system is able to detect number of fingers shown by users from one to five. From the number of fingers indicated by users we implement mouse functions, like for finger one mouse move function, for two fingers left click, for three fingers right click, for four fingers zoom in, and for five fingers zoom out function.

We examined all the mouse tasks such that left click, right click, zoom in, and zoom out on windows. We could not compare with the mouse device because this hand gesture system always shows lower performance real mouse

device. Instead of comparing with a real mouse, we allowed to use this system to know how it can be adapted easily [15].



Fig-9: Result showing the all processes of Hand detection

CONCLUSION

Human computer Interaction (HCI) through computer vision is more convenient way to interact with the device. In this work, hand tracking based virtual mouse application has been developed and implemented using a webcam. The system has been implemented in MATLAB environment using MATLAB Image Processing Toolbox. The result shows that, with this system we can interact with computer using our hand gesture instead using a mouse. The system can recognize and track the hand movement and can replace the mouse function to move the mouse cursor and the mouse click function. In general, the system can detect and track the hand movement so that it can be used as user interface in real time.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my guide, Prof. Smitha Raveendran of Electronics Engineering Department, Ramrao Adik Institute of technology, Navi Mumbai, for guiding me through this project work. She devoted her valuable time and helped me in all possible ways towards successful completion of this work. I am extremely thankful to her for giving me an opportunity to work on a very interesting area of image processing and for her excellent guidance and support.

REFERENCES

- [1] M. B. Ali, J. Farooq, 'Real Time Hand Gesture Recognition for Computer Interaction', *International Conference on Robotics and Emerging Allied Technologies in*

Engineering (iCREATE), Islamabad, Pakistan, April 22-24, 2014.

[2] S. S. Rautaray, A. Agrawal, "Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey", *Springer*, 06 November 2012.

[3] A. Chaudhary, J. L. Raheja, K. Das, S. Raheja , "Intelligent Approaches to Interact with Machines using Hand Gesture Recognition in Natural way: A Survey", *International Journal of Computer Science and Engineering Survey (IJCES)*, Vol.2, No.1, Feb 2011.

[4] R. R. Itkarkar, A. K. Nandy, 'A Study of Vision Based Hand Gesture Recognition for Human Machine Interaction', *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, Vol. 1, Issue 12, December, 2014.

[5] R. Hartanto, A. Susanto, P. I. Santosa, "Real Time Hand Gesture Movements Tracking and Recognizing System", *Electrical Power, Electronics, Communications, Controls, and Informatics International Seminar* , 2014.

[6] S. S. Rautaray, A. Agrawal, "Real Time Hand Gesture Recognition System for Dynamic Applications", *International Journal of UbiComp (IJU)*, Vol.3, No.1, January 2012.

[7] K. N. Shah, K. R. Rathod, S. J. Agravat , "A Survey on Human Computer Interaction Mechanism Using Finger Tracking", *International Journal of computer trends and Technology (IJCTT)*, Vol.7, No.3, January 2014.

[8] A. Banerjee, A. Ghosh, K. Bharadwaj, H. Saikia, "Mouse Control using a Web Camera based on Colour Detection ", *International Journal of Computer Trends and Technology (IJCTT)*, volume 9, number 1, Mar 2014.

[9] C. Kanan, G. W. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?", *www.plosone.org*, Volume 7, Issue 1, January 2012.

[10] http://en.m.wikipedia.org/wiki/Median_filter.

[11] A. Diplaros, T. Gevers, N. Vlassis, "Skin Detection Using the EM Algorithm with Spatial Constraints", *ieeexplore.ieee.org*, 2004.

[12] <http://www.mathworks.in/help/images/ref/im2bw.html>

[13] <http://www.mathworks.in/help/images/ref/regionprops.html>.

[14] How can I programmatically control mouse motion and clicks with MATLAB?

Retrieved from:

<http://www.mathworks.com/support/solutions/en/data/1-2X10AT/index.html>.

[15] D. C. Gope, "Hand Tracking and Hand Gesture Recognition for Human Computer Interaction", *American Academic & Scholarly Research Journal*, Vol. 4, No. 6, Nov. 2012.