



---

## International Journal of Intellectual Advancements and Research in Engineering Computations

---

### Finger gesture mouse pointer

**Manikandaprabu.N<sup>1</sup>, Dhineshkumar.C<sup>2</sup>, Hariharan.S<sup>2</sup>, Gowtham.P<sup>2</sup>, Susma.S<sup>2</sup>,  
Dr.S.Vijayachitra<sup>2</sup>**

<sup>1</sup>Ass.Professor, Department of Electronic and Communication Engineering,  
Nandha Engineering College [Autonomous]-Erode.

<sup>2</sup>UG Scholar Student Department of Electronic and Communication Engineering,  
Nandha Engineering College [Autonomous]-Erode.

<sup>3</sup>Professor, Department of Electronics and Instrumentation Engineering, Kongu Engineering  
College, Perundurai

---

#### ABSTRACT

Human-Computer Interaction (HCI) uses of the interfaces between people (users) and computers. In this paper, I have tried to control the mouse cursor movement and clicks events (Left & Right click) of the computer mouse using finger motion and gestures. Finger gestures were acquired using a camera based on skin detection algorithm with different angle detected images. This method mainly focuses on the use of Web-Camera to create a cost effective virtual mouse using webcam with image processing.

**Keywords:** Finger Motion and Gesture, Web-Camera, Image Processing, Skin Detection Algorithm

---

#### INTRODUCTION

The GUI on Personal Computers (PCs) is quite developed, well defined and provide an efficient interface for a user to interact with the computers and access the various applications with the help of mice etc. In the present day scenario most of the mobile phones are using touch screen technology to interact with the user. But this technology is still not cheap to be used in desktop and laptops. Our objective was to create a cost-effective virtual mouse system using Web-Camera to interact with the computer in a more user friendly manner with image processing. . The webcam capture the real time video at a fixed frame rate and resolution which is determined by the hardware of camera. The frame rate and resolution can be changed [1-3].

#### INTRODUCTION OF THE SYSTEM

In my work, I have tried to control mouse cursor movement and clicks actions using a camera based. Here a real time video has been captured using a Web-Camera and captured image of the finger using skin detection algorithm. Individual image of the video are separately processed. The processing technology involves noise remove technique and detect different angle image. When angle of the finger detected then system perform various operations to track the cursor and perform control actions. No additional hardware is required by the system other than the standard webcam which is provided in every laptop computer.

#### IMAGE ACQUISITION

An image is analyzed as it is clicked. Then the user is given tools to discard that it considered noise .The image acquisition is done using a digital

---

#### Author for correspondence:

Department of Electronic and Communication Engineering, Nandha Engineering College [Autonomous]-Erode.

camera or web-camera. For the system to work we need a sensor to detect the finger movement of the user. The web-cam of the computer is used to as a sensor. The webcam capture the real time video at a fixed frame rate and resolution which is determined by the hardware of camera [4]. The frame rate and resolution can be changed in the system if required

- Computer Webcam is used to capture the Real Time Video
- Video is divided into Image frames based on the FPS (Frames per second) of the camera

- Processing of individual Frame Here the input image got is an RGB image.

## PROCESS

### RGB Image

RGB is one of the formats of color images. Here the input image is represented with three matrices of sizes regarding the image format. The three matrices in each image corresponds to the colors red, green and blue and also says that of how much of each of these colors a certain pixel should use.



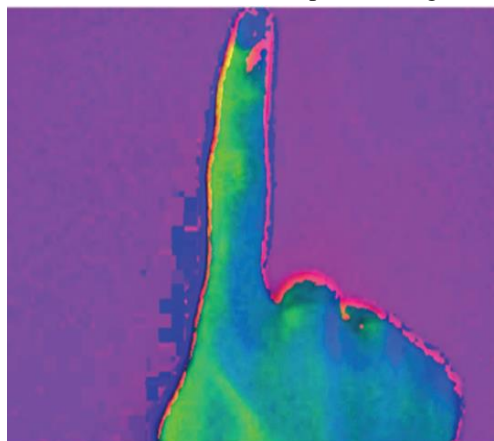
**Fig. A: RGB Image**

## FILTERING

The purpose of filtering is to smooth the image. This is done to reduce noise and improve the visual quality of the image. Often, smoothing is referred to as filtering. Here filtering is carried out by median filter since it is very useful in detecting edges [5, 6].

### HSV Image

Hue, Saturation, Value or HSV is a color model that describes colors (hue or tint) in terms of their shade (saturation or amount of gray) and their brightness (value or luminance) which is used to detect the skin color image. HSV is used to separate image luminance from color information.



**Fig. B: HSV Image**

### Y CbCr Image

Y is the luminance component and CB and CR are the blue difference and red-difference chroma

components. It is also use to detect skin in detect the skin color image.



Fig. 5: YCbCr Image

### BINARY IMAGE

A Binary Image is a digital image which has two assigned pixel values. Typically the two colors used for a binary image are black and white. The rgb filtered image of the finger converted into binary image this means that each pixel is stored as

a single bit (0 or 1). Thus the resultant image obtained is a monochromatic image consisting of only black and white colors. The conversion of binary is required because MATLAB can only find the properties of a monochromatic image. And then the outlined image got by binary image.

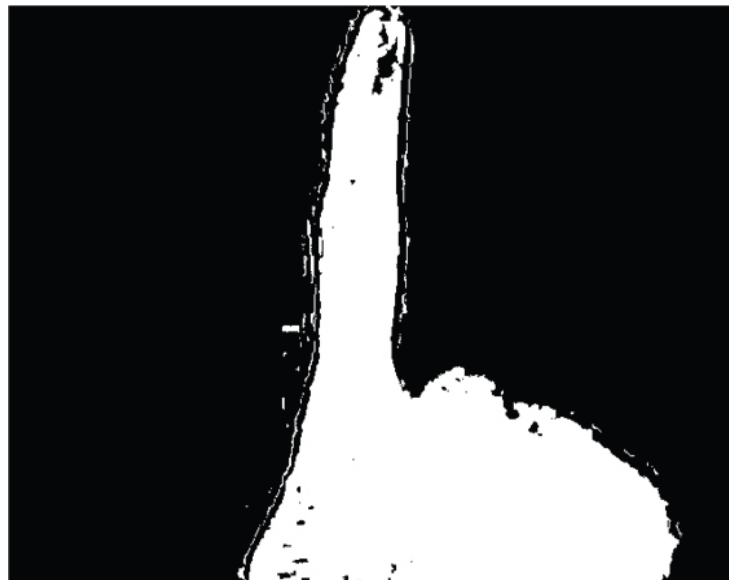
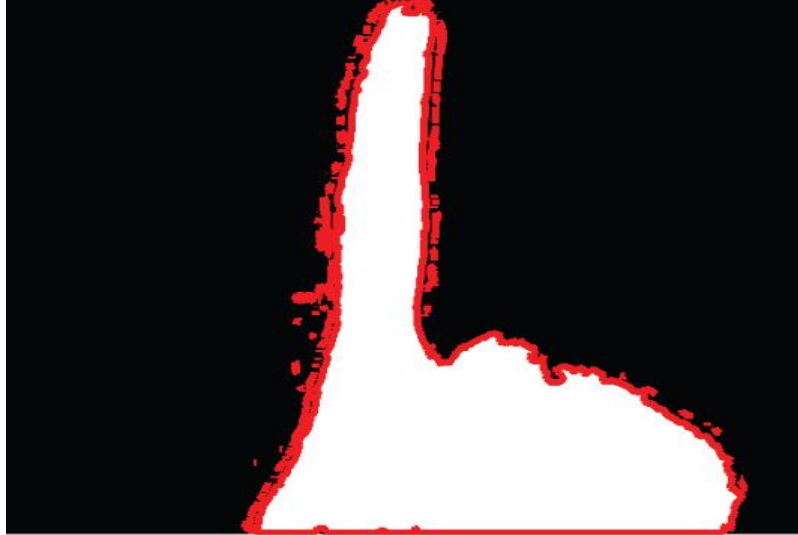


Fig. 6: Binary Image



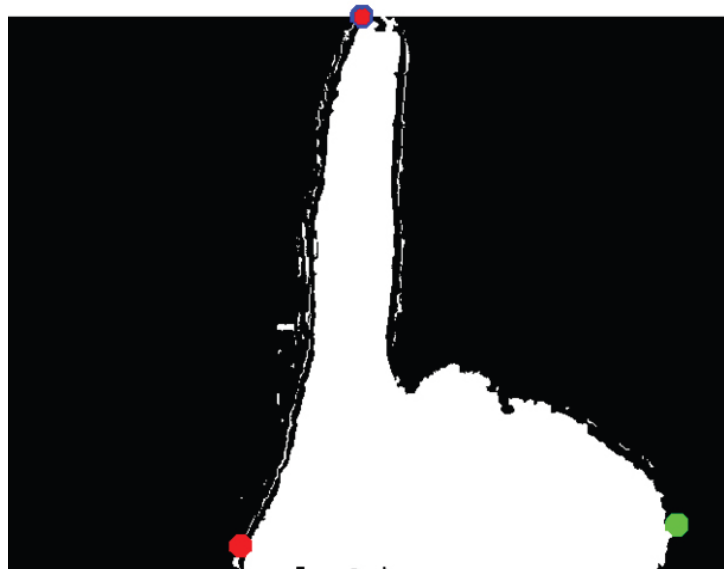
**Fig. 7: Binary image**

Finding Extreme point Centroid of the image and plotting: 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 inbuilt function in MATLAB is used to find the centroid and extreme point of the detected region. The output of function is a matrix

consisting of the X(horizontal) and Y (vertical) coordinates of the centroid.

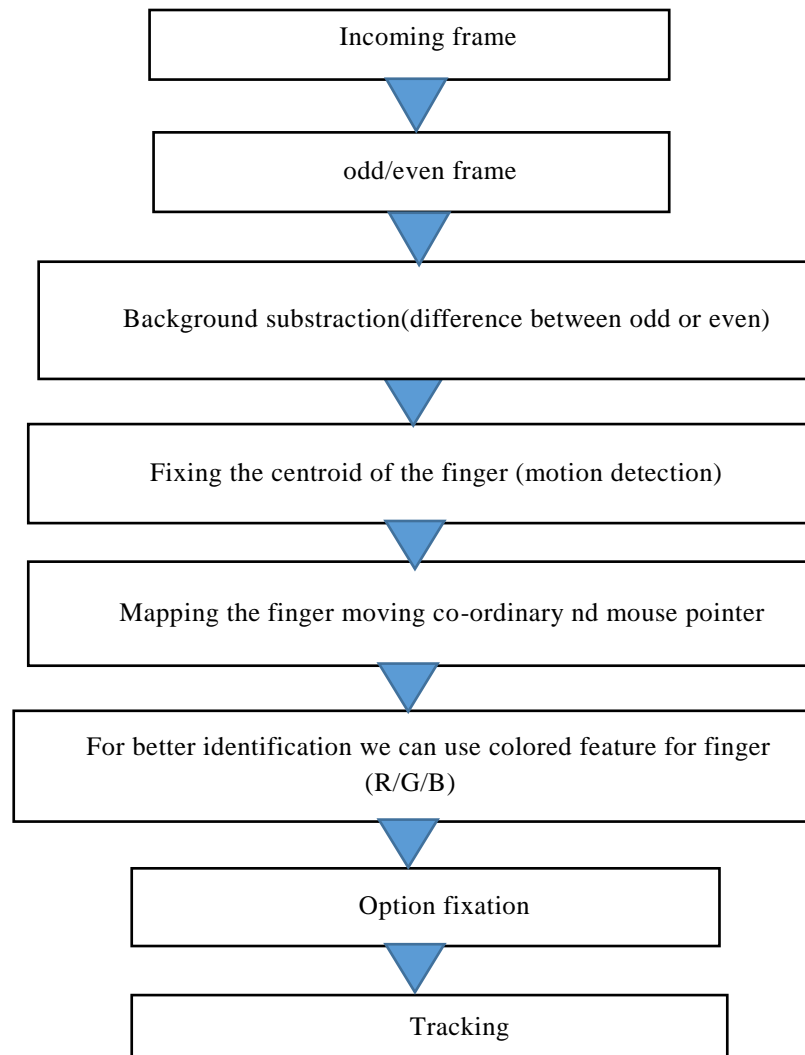
These coordinates change with time as the object moves across the screen.

- Centroid and extreme point of the image is detected.
- Its co-ordinates are located and stored in a variable.



**Fig.8: Binary image**

### Calculate angle at the centroid point



**Fig.9: Process methodology**

After find centroid and extreme point, calculate the angle at centroid point with two extreme point using three point formulas:

$$\text{Angle} = \text{acosd} \left( \frac{(x1^2 + x3^2 - x2^2)}{(2 * x1 * x3)} \right).$$

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 language like C, C++, and JAVA. so it is preferred over the others. 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. Based on the detection of

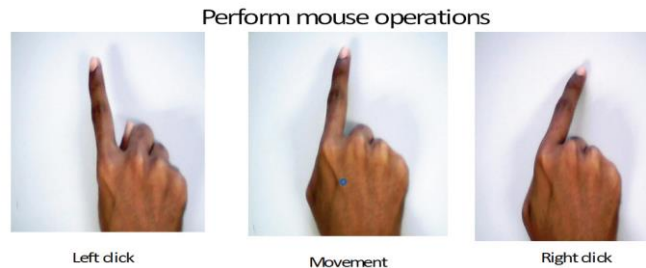
angle the system performs the movement and clicking operation.

To find the centroid of any triangle, construct line segments from the vertices of the interior angles of the triangle to the midpoints of their opposite sides. These line segments are the medians. Their intersection is the centroid.

Centroids may sound like big rocks from outer space, but they are actually important features of triangles. They also have applications to aeronautics, since they relate to the center of gravity (CG) of shapes.

## PERFORMING MOVEMENT AND CLICKING ACTION

The control actions of the mouse are performed by controlling the flags associated with the mouse buttons. JAVA is used to access these flags



**Fig:10**

User has to perform finger gestures in order to create the control actions. Movement and Clicking action is based on simultaneous detection of different angle at centroid point: If  $\text{Angle} > 110$  degree - Perform Left click  $110 < \text{Angle} < 80$  - Perform Mouse movement  $\text{Angle} < 80$  - Perform Right click.

## SYSTEM PROPOSAL AND GESTURE INTERFACE

The main goal of the system is to achieve an effective vision-based gesture interface that recognizes distinct gestures to enable the user to control the pan, tilt, and shutter of the camera. We must consider both intuitiveness of the gestures and the recognition rate, and ensure that they are comfortable to perform.

### Design of the Gesture Interface

To achieve objective (4), we designed a GUI with visual feedback on the area for gesture recognition. This area is shown using a circle, the inside of which corresponds to the tracking points that are used for gesture recognition.

When motion occurs, the tracking points move from the base position. Once the strength of the motion exceeds the threshold, an arrow appears, starting at the center and pointing to the edge; this arrow indicates the direction of the tracked motion, as shown in Figure 3, 4. The number shown below the circle indicates which gesture has been recognized.

### Detecting

We designed the system to respond to 20 tracking points on the circle (see Figure 3, Each tracking point was calculated by applying Lucas-Kanade optical flow [18] to subsequent pairs of frames, and the motion of the tracked point was summed to determine the overall motion.

If the strength of the motion exceeds a threshold, it is mapped to a gesture. In order to make more precise pointer of mouse finding centroid is necessary. Here „bwlable“ Matlab function can be used for cropping the genuine area. In other words the required region can be detected (see figure 10). To get the properties of the region such as center point or bounding box etc.,

Centroids may sound like big rocks from outer space, but they are actually important features of triangles. They also have applications to aeronautics, since they relate to the center of gravity (CG) of shapes.

During the period of time whereby the hand moves, the motion is calculated from the tracking points in the region. When the user's hand leaves this region, the resulting motion will be summed to calculate the mean direction of the gesture, which for the left/right sweep gesture will be motion to the left or right, respectively.

### Sweep Left/Right Gesture

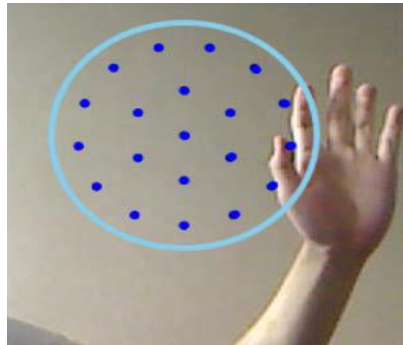
This gesture is shown in Figure 3, and can be performed from left to right or from right to left. During the period of time whereby the hand moves, the motion is calculated from the tracking

points in the region. When the user's hand leaves this region, the resulting motion will be summed to calculate the mean direction of the gesture, which for the left/right sweep gesture will be motion to the left or right, respectively. This gesture is, and can be performed from left to right or from right to left [8, 9]. During the period of time whereby the hand moves, the motion is calculated from the tracking points in the region. When the user's hand leaves this region, the resulting motion will be summed to calculate the mean direction of the

gesture, which for the left/right sweep gesture will be motion to the left or right, respectively.

### Waving Gesture

This gesture is shown in Figure 5. The waving gesture can be performed by moving hand left/right repeatedly over the area. To recognize this gesture, we detect whether the direction of motion of the hand has reversed over two subsequent frames. A reversal corresponds to a change in direction of at least 135.



**Fig:11**

Each such reversal of the direction of motion is processing. pixels. The interface was shown on the right side of the face, and participants were asked to use their right hand to perform the gestures. The participants tried to familiar with the gesture interface in about 5 minutes. The prototype system

for the experiment is shown in Figure 6. The self-portrait system, including a pan-and-tilt camera and notebook computer. Gestures are used to control camera functions, such as pan, tilt and shutter trigger.



**Fig:12**

## CONTROL METHOD OF HAND MOUSE FUNCTION

An intuitive UI was introduced in "Toward the Use of Gesture in Traditional User Interfaces" by Kjeldsen et al., in 1996. In this, a user's hand was recognized by a camera from the moment the user placed the mouse. Then, the user selected a virtual menu that was created on the real monitor using the hand [10].

This study was followed by several others on UIs and new methods using gestures, but each new interface had the burden of memorizing a gesture for a user command. Thus, these interfaces were not intuitive at all. In addition, only a user who was aware of more than 15 gestures could use the interface. The movement of the cursor was controlled by the tip of the index finger. In order to identify the tip of the index finger, the centre of the palm must first be found. The method used for finding the hand centre was adopted from and it

has the advantage of being simple and easy to implement.

The shortest distance between each point inside the inscribed circle to the contour was measured and the point with the largest distance was recorded as the hand centre. The distance between the hand centre and the hand contour was taken as the radius of the hand. Table 3 lists the results in three lighting conditions. The average precision of recognition was 1.0, and the average recall was 0.9 in normal lighting condition. It was 1.0 in dark conditions and in bright conditions, with average recall rates of 0.9 and 0.91, respectively. The experiment shows no obvious conflicts among the gestures. If the left hand moves in the virtual monitor, the left hand coordinate of the virtual monitor can be controlled by converting the coordinates of the mouse pointer on the real monitor.

TABLE III. THE GESTURES PERFORMANCE, WHERE LIGHTNESS = 150 / 100 / 200 ± 10.

Gestures	Correct	Wrong	Lost	Precise	Recall	Time(s)
Sweep to left	16/16/16	0/0/0	3/4/3	1.0/1.0/ 1.0	0.84/0.80 /0.84	29.1 ( $\sigma = 1.7$ )
Sweep to right	16/16/16	0/0/0	2/3/2	1.0/1.0/ 1.0	0.89/0.84 /0.89	
Circling CCW	16/16/16	0/0/0	1/1/1	1.0/1.0/ 1.0	0.94/0.94 /0.94	22.7 ( $\sigma = 1.9$ )
Circling Cw	16/16/16	0/0/0	2/0/2	1.0/1.0/ 1.0	0.89/1.0/ 0.89	
Waving	16/16/16	0/0/0	1/1/0	1.0/1.0/ 1.0	0.94/0.94 /1.0	22.5 ( $\sigma = 1.0$ )

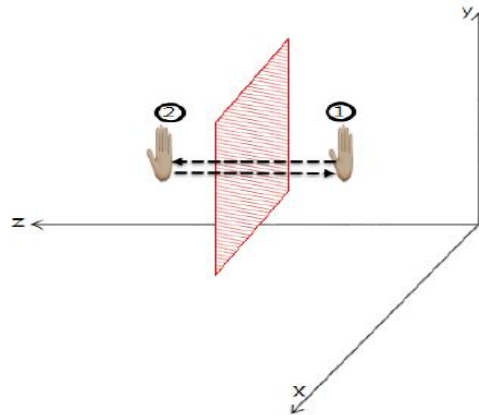
The time to finish the test in three conditions was 24.8 s on average, giving a time of 2.5 s to perform each gesture. The one-way analysis of variance (ANOVA) of the recall result in these three lighting conditions, shown no significant effects,  $F(2, 12) = 0.04$  with  $p = 0.96$ . Therefore, we conclude the overall gesture recognition accuracy was 0.9 in the lightness range 100-200. hand mouse interface reminiscent of the most commonly and intuitively used mouse actions. The actions that a user takes to use a mouse include mouse movements and button clicks.

Therefore, actions that may be reminiscent of the mouse can be controlled by simply moving (left hand) and pressing (right hand) the mouse. The use of a hand-mouse interface with a user's

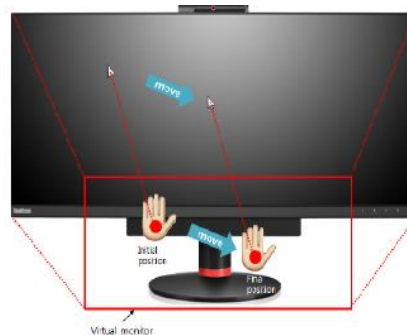
hand is very simple and intuitive. The left hand controls the movement of the mouse, and the right hand controls a mouse click. If the left hand moves in the virtual monitor, the left hand coordinate of the virtual monitor can be controlled by converting the coordinates of the mouse pointer on the real monitor. Fig. 8 shows the movement of the mouse pointer by the movement of the left hand in the virtual monitor. As can be seen from Fig. 8, because the movement of the mouse pointer can be controlled in the same way as the left hand movement on the virtual monitor, the user can move the mouse pointer exactly to the desired position on the real monitor. Simultaneously, it is possible to provide the user with intuitive feedback.

While the left hand can control the movement of the mouse pointer, the right hand can control the mouse click function. As shown in Fig. 9 the movement of the right hand to control the click function on the virtual monitor, which can be re-organized by stretching out and bending the user's right hand from position 1 to position 2. The

behavior of the right hand can be understood in the sense of touching the touch-screen with the right hand. By merely using the depth information of the right hand and virtual monitor to control the movement of the right hand, the click function can be controlled.



**Fig.13: Movement of right hand to control click function**



**Fig.14: Movement of mouse pointer by movement of left hand in virtual monitor**

## PROBLEMS

Since the system is based on image capture through a webcam, it is dependent on illumination to a certain extent. The system might run slower on certain computers with low computational capabilities because it involves a lot of complex calculations in a very small amount of time. However a standard pc or laptop has the required computational power or optimum performance of the system. Another fact is that if the resolution of the camera is too high then the system might run

slow. However this problem can be solved by reducing the resolution of the image by making changes in the system.

## CONCLUSION

In this paper, a virtual mouse application has been developed and implemented using a webcam. The system has been implemented in MATLAB environment using MATLAB Image Processing Toolbox. This technology has wide applications in

the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. This technology can be used to help patients who don't have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a

person's motions are tracked and interpreted as commands. Most of the applications require additional hardware which is often very costly. Our motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system.

## REFERENCES

- [1]. A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall of India, First Edition, 1989.
- [2]. Gajanand Gupta, "Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter", International Journal of Soft Computing and Engineering (IJSCE), 2011, 1(5).
- [3]. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Pearson Education, Second Edition, 2005.
- [4]. Hojoon Park, "A Method for Controlling the Mouse Movement using a Real Time Camera", Brown University, Providence, RI, USA, Department of computer science, 2008.
- [5]. Shahzad Malik, "Real-time Hand Tracking and Finger Tracking for Interaction", CSC2503F Project Report, 2003.
- [6]. 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 7(3), 174-177, 2014.
- [7]. Delores M. Etter, "Introduction to Matlab", 2010.
- [8]. A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E. "Computer vision based mouse", Acoustics, Speech, and Signal Processing, Proceedings (ICASS) IEEE International Conference, 2002.
- [9]. Shaalini, R., et al. "Human motion detection and tracking for real-time security system." International Journal of Advanced Research in Computer Science and Software Engineering, 3(12), 2013.
- [10]. Roopa, V. G. "Real Time Visual Tracking of the People using Video Camera with Reduced Time Complexity." Unique Journal of Engineering and Advanced Sciences 2, 2014.