



# Mouse Movement through Finger by Image Grabbing

G Nagendra Babu, Anees Ahamed Baig, Chaganti Sai Sree, Eedhi Akhilandeswari

**Abstract:** Touch less Interactive screen or system is an upcoming technology in the real life world, which will be another biggest realization followed by the technologies based on the monitors cover as PC monitor, TV Screens without a stroke or touch. The aim of this project is to develop a touch less interactive screen which helps to ease the controlling of any kind of monitors. Project is based on MATLAB platform. In this work, the finger gestures which are captured by a webcam and that will be converted into mouse movements to control the screen. Various Analysis result has included in the report to show the feasibility of the application of this development. A Graphical Interface User (GUI) is developed to let the user to choose perform calibration and number of times it's required to perform calibration and to trigger the program upon user's request.

**Keywords:** Digital image processing, object detection, windowscreensynchronization, GUI analysis, gesture control, and image grabbing

## I. INTRODUCTION

### Background:

Interactive touch less screen or system is an undergoing developing technology in the real life. It is a new evasion after the newly developed touch screen technologies such as iPads and iPhones. As what it says screen or system will be controlled via touch less features. With the help of touch less technology anyone would be able to control the screen or system without a touch or stroke. A touch less gesture is the next frontier of the touch technology. The conceptual idea of touch less gestures has the potential to change a lot of what we know about touch. At Present touch less technologies available and named as flash scanning of finger Prints, vision – based input interface, touch less control technology

### Motivation:

Touch less technology helps people, who work in production plants, engineering sectors, research centers and especially those disabled ones who don't have to touch on any kind of hardware. Existing touch less technologies are developed on measuring temperatures, sensors, flash scanning of fingerprints and finger gestures via image processing method.

Revised Manuscript Received on April 21, 2020.

\* Correspondence Author

**Nagendra Babu Gugulothu\***, Asst. Professor, Department of Electronics and Communications, PSCMR College of engineering and technology At Vijayawada.

**Anees Ahamed Baig**, Student, B.Tech in Potti Sriramulu Chalavadi Mallikharjuna Rao College of engineering and technology, Vijayawada

**Sai Sree Chaganti**, Student, PSCMR CET, Electronics and Communication Engineering

**Akhilandeswari Eedhi**, Student, PSCMR CET, Electronics and Communication Engineering

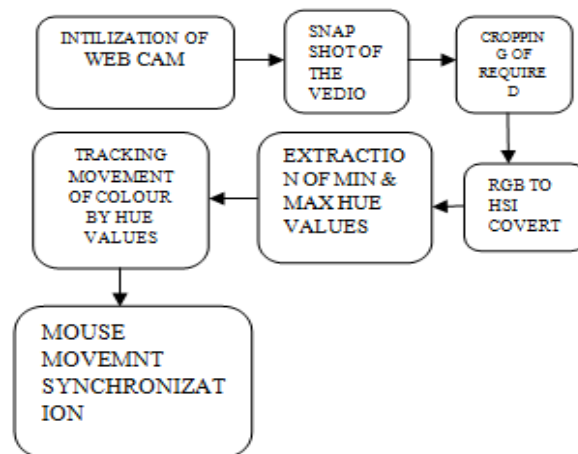
© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

In our work, screen can be controlled by using additive primary colors such as Red, Green, and Blue not only that system is been enhanced to handle any kind of colors beside additive primary colors. The system is enabled to perform calibration of any used in subsequent process of Image processing. System will be ready to perform touch less interaction once after the designated color has been identified via; image processing method

### Project Scope:

- To acquire the minimum and maximum values of the captured frames as part of calibration process using image acquisition toolbox in MATLAB
- To convert the captured RGB (additive primary colors) images values via webcam into HIS (Hue, saturation, intensity) values to not to be affected by the variation of light intensity as part of pre-processed image data.
- To enhance the software codes to filter the noise and to remove the unwanted components from the pre-processed image data.
- To perform the synchronization of the gesture or color movements into mouse movements on the system screen. To test and study the implementation java class functionality on MATLAB platform to perform the desired mouse clicks on the screen.

### Block diagram of the process:



## II. DESIGN AND METHODOLOGY

### Project Approach and method:

Project objective is to develop interactive touch less screen, which helps the humans to navigate the mouse on the screen or system without having any touch or stroke

using hardware items such as mouse or keyboard. There are a lot of interactive technologies are available in the market and most of them are under development stage based on various image processing methods such as measuring temperatures, sensors, flash scanning of finger prints and finger gestures via image processing method.

In this project, interactivity of the touch less screen will be developed based on digital image processing methods on MATLAB Platform. There are two processes has to be performed in order to achieve the project scope.

1. Perform Color code Calibration – A function to perform color code calibration via webcam to decide the object that user suppose to use for tracking purpose. User may choose any additive primary colored object for tracking purpose. Calibration codes detect the object color in RGB color model and convert into HIS color space then extract and pass down the HUE value to the object tracking program for further process. Since the calibration takes the HUE value of the color code tracker, the light intensity variation will not be affected deeply.
2. Perform object tracking and mouse interactivity on screen – A real live video will be continually captured through webcam and captured data will be pre-processed based on the pass down HUE value from the above function. Program starts to detect the object and the movement of the object will be synchronized with mouse movement on screen once the given HUW value conditions are satisfied in object tracking program. Interactivity enables the user to navigate the mouse on screen and performs mouse functionalities such as open, close, save and edit.

### Software:

Software design will be performed on MATLAB platform version R2009b because of its powerful toolbox's called as image acquisition and processing toolbox. There are various functionalities are available in MATLAB which helps to perform smooth image acquisition and pre-processing of image enhancements and analysis.

## III. RESEARCH AND METHODOLOGY

### Calibration Of Color Code Tracker

Calibration of color code tracker consists of various image processing phases like capturing image from a real live video and cropping of the color code tracker. A real live video will be continually captured using webcam for an unlimited period. Calibration allows users to choose the total number of calibration is required to perform. User may opt to start perform calibration and required calibration number via Graphical User Interface.

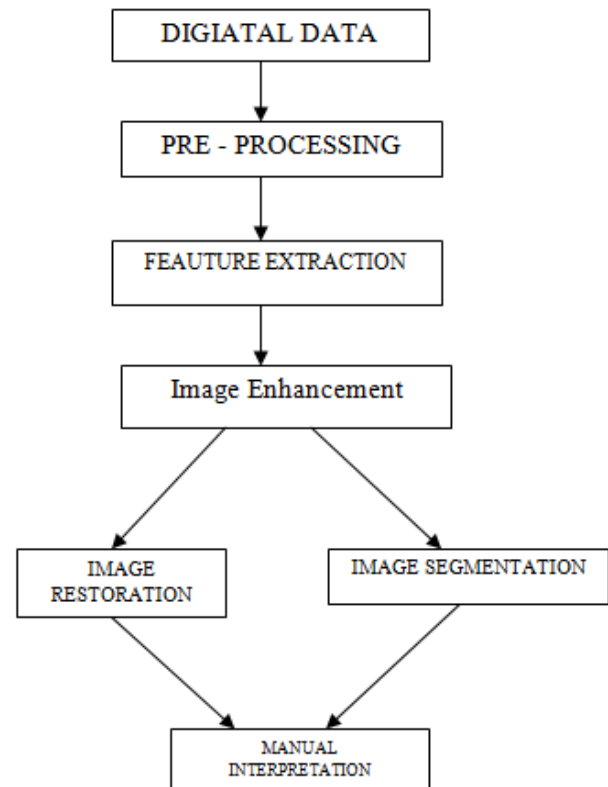
### Webcam Initialization

Webcam Initialization will be performed differently on various PC's or Laptops. It has to be configured based on the Webcam configuration in each system, type Matlab code image on Mat lab command window as shown below figure 3.1(a) Mat lab Command Window. System shall return the hardware information cached by the toolbox.

### Digital Image Processing:

Digital image processing refers to the processing of digital images by means of digital computer and its being composed with a finite number of elements,

each value has a particular location and elements referred as pixels an image may be described as a 2-D function,  $f(x, y)$



Where  $x, y$  are spatial coordinates and the amplitude of  $f$  at any pair of coordinates  $(x, y)$  is called the intensity or grey level of the image at that point. In image processing one of the hypothesis is to consider three types of computerized process in the continue 'low-mid-high-level process'. Low level process is image pre processing the primitive operation such as to reduce noise and trust enhancement and image shaping. Mid-level process involves image segmentation, categorization of each object and to convert and to convert into a suitable form for computer processing. Finally high level process involves the ensemble of the distinguished objects.

### Image Cropping

Image cropping functionality helps to crop the desired color code object tracker on the captured frames from real live video. The Mat lab Code 'get snapshot' can be ;used for to capture the image from the video and code "micro" helps to enable the cropping functionality of the image as shown in figure 3.2 Image Cropping. The cropped data will be converted into data type double for further image pre-processing. Below Mat lab code does the image cropping process.

### Simulation code:

```

Data = get snapshot (vide); % retrieves frames mishap (data);
Data Cropped = in crop;
Data Cropped 1 = double (data cropped);
RGB to HIS Conversion Codes
    
```

The captured and then cropped image in RGB color model as shown in Figure 3.3(a) Cropped Image, need to be converted into HIS color space to extract the HUE value from the capture images. The HUE value of each color will not be deeply affected by the light intensity variation on the background. Following codes will be used for to perform the RGB to HIS conversion and performs HUE value normalization before passing the value for the subsequent process as shown in Figure 3.3(b) Normalized HUE Value.

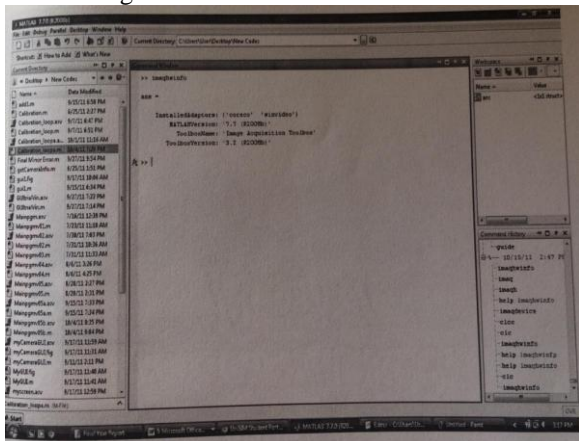
```
Code:
R=dataCropped1 (: 1); g=dataCropped1 (: 2);
b=dataCropped1 (: 3);
S=1-3.*(min(min(rag),b))./(r+g+b+eps); I=(r+g+b)/3;
Th=acos((0.5*((r-g)+(r-b)))/((sqrt((r-g).^2+(r-b).*(g-
b))))+eps));%H value conversion H=th;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
%performs normalization
```

The minimum and maximum of the above red color as shown below and these values will be saved into a TXT file and for to be retrieved by the object tracking program on later stage.

$$Z=[0.9005 \ 0.9976]$$

**Image Enhancement**

This process includes enhancements of the image that results more appropriate to certain applications or experiments and it is improve o the quality of the images using software. Filtration is one of the task in image enhancement algorithm



**Non Spatial Filteration Of Image**

Commonly used order-statistic filters are called non linear spatial filters. It responses based on ranking result which does the replacement of the central pixel in neighborhood with the value based on the ranking result.



Figure 2.2 Noised Image before filtering

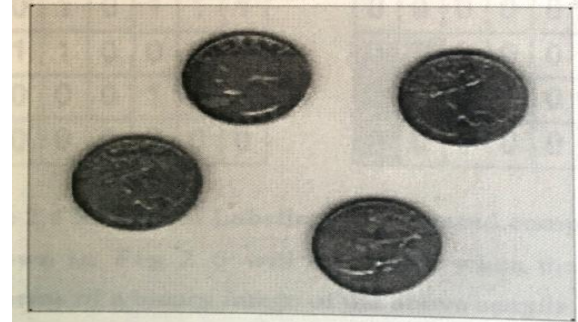


Figure 2.3 Image after median filtering

**Object Tracking And Screen Interactivity**

Object tracking codes continuously tracks the movement of the color code tracker through webcam and the movement will be synchronized with mouse movements on the screen. As per explained in section 3.3 Calibration of color code tracker, Object reaches the level of HIS to Binary Image Conversion. Object tracking codes have been enhanced to perform the mouse functions like right click or left click which helps to perform open/close, save, edit functions.

**Object Detection**

The converted binary image will go through morphological process for to perform the object detection as shown in Figure 3.4. First process is to remove pixels, Matlab code “bwareaopen” helps to remove the unwanted pixels from the connected components and produces another binary image. Second process is to label the connected objects, Matlab code “bwlabelm” always returns a label matrix those contains label for the connected components in the binary image and it also returns the total number of connected objects in found in the binary image and it also return the objects detection those having same HUE value. Last process will be to find the properties of the binary image such as Area, Filled Area, and Perimeter. Matlab code “region props” returns the properties of each labeled region ion the label matrix. Below Matlab code does the object detection process

**Simulation code:**

```
% Remove all those pixels less than 300
pixelsH_live_up = bwareaopen(H_live_up, 300);
% Label all the connected components in the image.
%Restrict multiple detection via num
[Lanus]=bwlabeln(H_liv
e_up);
```

```

Obi=regionprops(L,'BoundingBox',
'Centroid','Area','FilledArea','Perimeter','Orientation')
;
Len = length(obj);
Bb = obj(object). Centroid;
Rectangle('Position',bb,'EdgeColor','r','LineWdth',2)
plot(bc(1),bc(2),'m+');%Determine
X and Y co-ordinate a=text(bc(1)+15,bc(2),
strcat('X:',num2str(round(bc(1))),'Y:');
Num2str(round(bc(2))));
Set(a,'FontName','Arial','FontWeight','bold','FontSiz
e',12,'Color','Yellow');
Set(0,'pointerLocation',[(bc(1)+250)(bc(2)+250)])

```

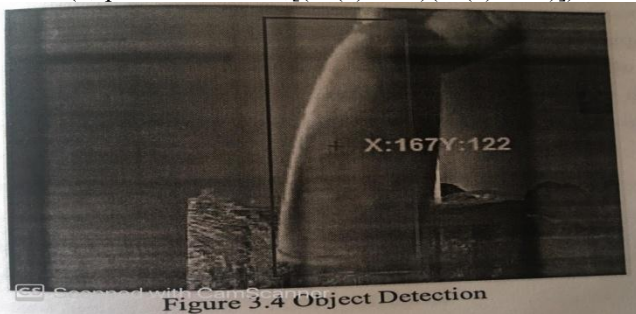


Figure 3.4 Object Detection

**Window Screen Synchronization**

Based on the co-ordinates of the location of the identified color which is calibrated earlier, the window screen mouse co-ordinates (x, y) are to be mapped accordingly. This is done by co-relating the Matlab video frame co-ordinates to Window screen co-ordinates mathematically. Below Matlab codes perform the window screen synchronization to move the mouse pointer based on this co-relation % Calculation of X clue:-

**Simulation code:**

```

If
Bc(1)==0,t_x=0;
Elset_x=bc(1)*4.666;
End
Round(t_x);
%Calculation of Y value:-
Id bc(2)==300,
T_y=0;
Else
T_y=(bc(2)-300)*-1*2.66;
End round(t_y); set(0,'pinterlocation',[t_x,t_y]);
Mouse controls and functions

```

Java class function is used to perform mouse controls and functions. Function “import package name “adds the specified package name to the current import list.

In this work only 2 packages are used for the mouse controls and functions. Those are “java.Awt.Robot” and “java.awt.event”. Below Matlab code does the mouse controls and functions, calling java function to perform left click and right click import fava.awt.Robot

```

Import java.awt.event.*; mouse=Robot;
Mouse, mouse Release (inputEvent.BUTTON3_MASK);
Mouse, mouse Release (inputEvent.BUTTON3_MASK);
Mouse.mousePress (inputEvent.BUTTON1_MASK);
Mouse.mouseRelease(inputEvent.BUTTON1_MASK);

```

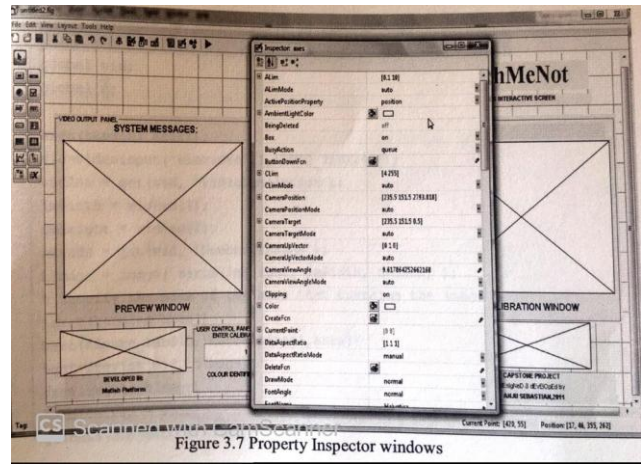


Figure 3.7 Property Inspector windows

**Matlab Gui Design**

Matlab GUI will be developed using MATLAB graphical User Interface design window type “guide” opens the GUI design environment. GUIDE initiates tools that allows GUI to be created or edited interactively from FIG-files or handle(s) to figure. Calling GUIDE by itself will open the GUIDE Quick Start Dialog where can choose to open a p0reviously created GUI or create a new one from one of the provided GUI templates as shown in Figure 3.5 Guide Quick Starts.

GUIDE (filename) opens the FIG-file named ‘filename’ for editing if it is on the MATLAB path. GUIDE (full path) opens the FIG-file at ‘full path’ even if it is not on the MATLAB path.

Here we started t o create a new GUI from default blank template as shown in Figure 2.10 a Blank GUI. Using the palette on the left side of the GUI adding static text box, Axes pushbuttons are used to create a Matlab GUI as shown in Figure 3.6 initial Draft of GUI.

Each component has its own default properties and its editable using the property Inspector. Select the component which needs to be edited and when perform tight click on that a new pop-up window will be up to edit the properties as shown in Figure 3.7 Property Inspector windows of Matlab GUI components.

Once after the layout of the GUI completed then the next step is need to integrate the program in each component. An m.file will be auto generated upon save the GUI. Each component has callback functions and it can be edited using the m.file. For example, in the m.file, codes have been added to trigger and preview video when pushbutton2 START PREVIEW is pressed as shown in Figure 3.8 Example of m.file coding on GUI.



IV. RESULT AND DISCUSSION

GUI Analysis

The Graphical user interface allows users to perform color code tracker calibration and user may trigger the program from the GUI. Illustrated below on step by step that how to perform calibration for example using red color and GUI Instruction guide helps the user to perform various steps and how to trigger the tracker program.

Step 1: Start-up GUI

When the GUI started, user could see the following screen and allows the user to perform calibration of the designated color code tracker as shown I Figure 4.2

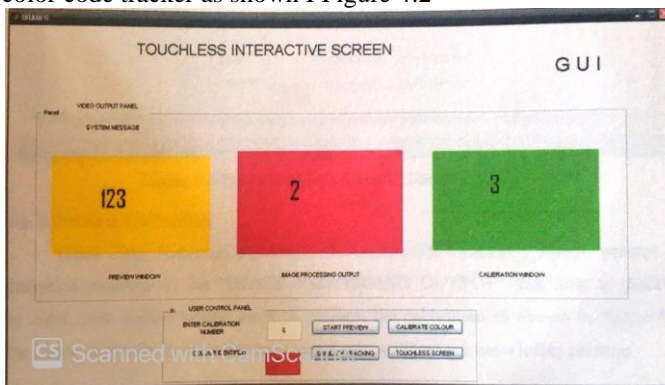


Fig (a)

Step 2: Required Calibration Number

Real live video capturing will be performed once upon click the pushbutton “START PREVIEW” and it will be shown in the “PREVIEW WINDOW”. When the user is decided the total number of calibration is required to be performed, and this number can be entered into the free text column “Enter calibration NO” for example ‘1’ as shown in Figure 4.3. This will allows the system to decide the total number of calibration to be performed.

Step 3: perform Calibration

When User clicks on pushbutton “CALIBRATE COLOR”, system prompt an instruction message in the “IMAGE PROCESSING OUTPUT” that how to position the color code tracker before starts to perform the calibration as shown in Figure 4.4. The screen also tells the user that how many more calibrations have left to perform.

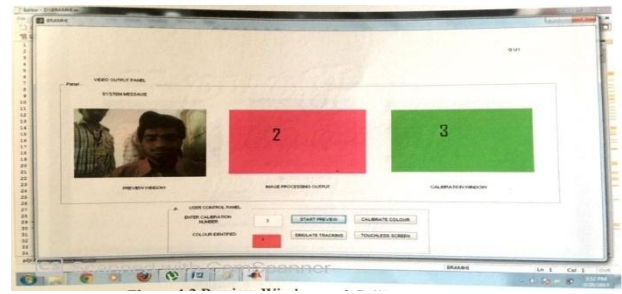


Fig (b)

Step 4: Image Cropping

Align the color code tracker in the top left corner of the window and press enter to continue the calibration, so then snapshot of the real live video will be performed and right click to choose the cropping functionality as shown in Figure4.5

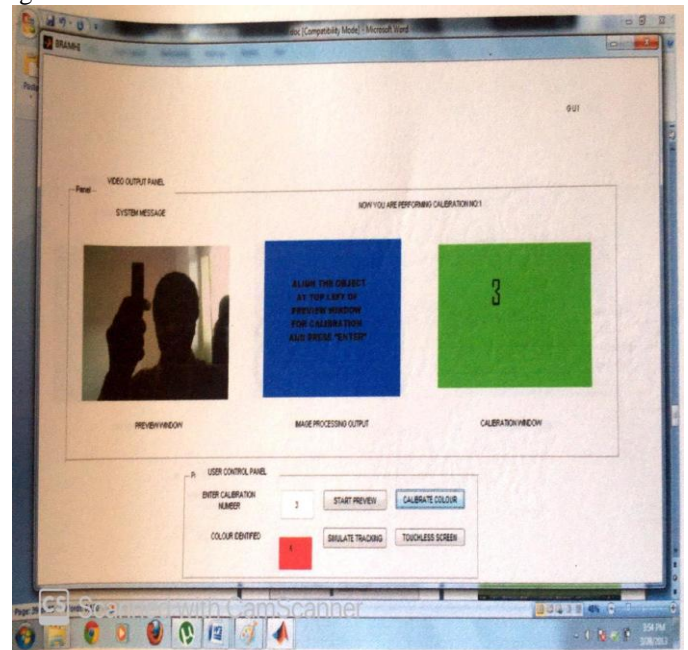


Fig (c)

Step 5: Simulate Tracking

System prompts message upon successful completion of calibrations. Instruction panel shows that, click on pushbutton “SIMULATE TRACKING” to view the color code tracking and GUI shows the identified color in the screen as shown in Figure 4.6.

Table 4.3 summarizes test results conducted under incandescent light mode and the simulation attempt was to track the mouse, to open a folder at windows desktop. Time taken for individual successful trials were recorded down and compared with actual mouse response time for the same action

Average time taken to open a folder at windows desktop by using an external hardware mouse is found to be 2.5 to 3 seconds. Same action was simulated using the touch less interactive screen for 10 times to measure the responsively of the system. Test results are shown below and responsively of the mouse action via touch less interactive screen was calculated.

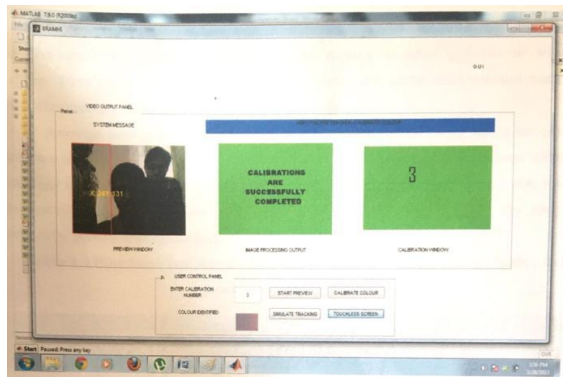


Figure 4.7 Simulate Tracking



Figure 4.8 Touchless Interactions on Screen

Based on the above test results, highest variance is observed at trial number 6 with a variance of 2.1 seconds compared to the relative value of 3 and the least out of 10 trials made, 3 trails were with a better responsively compared to external hardware mouse.

Rest of the 7 trials shown slower response than the hardware mouse, with the largest responsively deviation of 1.6 seconds than the external hardware mouse.

This is to set the camera mode to nullify the effect of light intensity due to outside environment like Night mode, bedroom mode, sunny mode gloomy mode and incandescent mode etc...

### Advantages

- 1) It is easier and satisfactory experience
- 2) No conventional input devices are required.
- 3) It protects against vandalism.
- 4) It is suitable for large screen displays.
- 5) It overcomes contamination problem unlike touch technologies.

### 5.3 Applications

- 1) Can be used in smart phones and smart screens.
- 2) We can also used with virtual keypads.
- 3) Used for disabled persons.

## V. CONCLUSION

Project objectives achieved using Image Processing technologies and its functionalities on Matlab platform. In this work we have analyzed and concluded that window screen can be controlled by gesture analysis and it's not purely developed based on gesture movement but on color code tracker though the mouse movements are controlled by gesture movements, in these work we analyzed window screen can be controlled by gesture analysis.

## REFERENCES

1. Alasdiar Mcandrew, 2004, Introduction to Digital Image Processing with MATLAB, Course Technology.
2. Charles A. Poynton (2003). Digital Video and HDTV: Algorithms and interfaces. Morgan Kaufmann. ISBN 1558607927.
3. Rafael C. Gonzalez, Richard Eugene Woods, Steven L. Eddins, 2004, Digital Image processing using MATLAB, Pearson, Upper Saddle River, New Jersey.
4. <http://blogs.mathworks.com/steve/2007/05/11/connected-component-labeling-part-5/>
5. [http://en.wikipedia.org/wiki/RGB\\_color\\_model](http://en.wikipedia.org/wiki/RGB_color_model)
6. <http://ewh.ieee.org/r8/uae/GUL.pdf>
7. <http://software.intel.com>
8. [http://web2.clarkson.edu/class/image\\_process/RGB\\_to\\_SHI.pdf](http://web2.clarkson.edu/class/image_process/RGB_to_SHI.pdf)
9. <http://www.mathworks.com/help/toolbox/images/f8-20792.html>
10. <http://www.mathworks.com/help/toolbox/images/f11-12251.html>
11. <http://www.scribd.com/doc/16617255/HSI-colour-space>
12. [www.mathworks.com/matlabcentral/fileexchange/30546-tracking-yellow-color](http://www.mathworks.com/matlabcentral/fileexchange/30546-tracking-yellow-color)
13. <http://www.mathworks.com/help/toolbox/images/ref/rgb2gray.html>

## AUTHORS PROFILE



**Nagendra Babu Gugulothu** is working as Asst. Professor, Department of Electronics and Communications, PSCMR College of engineering and technology at Vijayawada. He is specialized in handling Embedded and Iot stream research work and he had handled nearly 50 projects in IOT and EMBEDDED. He is good at research work terminology and published few papers



**Anees Ahamed Baig** is a second year student pursuing B.Tech in Potti Sriramulu Chalavadi Mallikharjuna Rao College of engineering and technology, Vijayawada. He attended many workshops, conferences and presented more reputed papers in the field of Electronics and technology Drives. He got a national level 1<sup>st</sup> prize in paper presentation at IISC Bangalore. His interest in the field of AI and NEURALNETWORKS. He is also perusing a Data science certified by IBM



**Sai Sree Chaganti** is perusing graduation 2nd year in PSCMR CET in the field of ELECTRONICS AND COMMUNICATION. She attended workshops in different domains and published a paper in IOT domain. She did some projects in IOT domain. She got some certificates and BADGES from IBM. She interests in the field of AI and ML



**Akhilandeswari Eedhi** is perusing graduation 2nd year in PSCMR CET in the domain of electronics and communication engineering. She published a paper in various journals and attended the workshops on EMBEDDED SYSTEMS and IOT. She got the certificates and badges in IBM. She interests in the field of AI and DL