ISSN: 2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Touchscreen Using Web Camera

Kuntal B. Adak[1], Adarsh N. Singh[2], Abhilash B. Kamble[3]
Department Of Computer Engineering,
University Of Pune.

Abstract—**In this paper we present a web camera based touchscreen system which uses a simple technique to detect and locate finger. We have used a camera and regular screen to achieve our goal. By capturing the video and calculating position of finger on the screen, we can determine the touch position and do some function on that location. Our method is very easy and simple to implement. Even our system requirement is less expensive compare to other techniques.**

Index Terms— Touchscreen, Webcamera, Computer input-output, H.C.I.

1 Introduction

At present, touch screen is the technology that is widely used for communication with electronic devices such as computers, laptops, mobile phones, tablets and many more. The advantage of touch screen is that it allows the user to directly interact with the screen without the help of any other intermediate devices such as a mouse. In our proposed system we will develop a less expensive web camera based touch screen system that will use simple finger detection technique. Only a screen and a single web camera is required to accomplish this goal. By using the geometrical area of the screen and the camera positions we may resolve the finger position on screen. Our system will work well under different surrounding environments and the detection method used will able to work with various finger colors and shapes.

There are many types of touch screen techniques. Our touch screen technique is very easy to use. Compairing to other technique the webcam based touch screen technique is less used due to imperfection i.e. the touch point accuracy is very low. To overcome most of the drawbacks of web camera based touchscreen we prefer our own technique in this paper. This paper shows how we can locate the accurate finger point on the screen with the help of a web camera. For performing the touch screen operation we used single web camera and a normal screen.

This paper is divided into multiple section viz introduction, Related work, detailed information on our proposed system, and our future work.

2 Related Work

Till today many touchscreen systems were developed. Some of these technologies are resistive, capacitive, infrared acrylic projection, optical imaging, glove-based and vision-based gesture technologies etc.

Resistive LCD touchscreen totally rely on touch overlay, which is composed of a flexible top layer and a rigid bottom layer separated by insulating dots, attached to a touchscreen controller.

The inside surface of each of the two layers is coated with a transparent metal oxide coating of Indium Tin Oxide (ITO) that facilitates a gradient across each layer when voltage is applied. Pressing the flexible top sheet creates electrical contact between the resistive layers, producing a switch closing in the circuit. The control electronics alternate voltage between the layers and pass the resulting X and Y touch coordinates to the touchscreen controller. The touchscreen controller data is then passed on to the computer operating system for processing. Resistive touch screen panels are generally more affordable but offer only 75% clarity and the layer can be damaged by sharp objects. Resistive touch screen panels are not affected by outside elements such as dust or water. Resistive Technology is divided into two broad categories: 1) 4 -Wire Resistive Touchscreen Technology and 2) 5-Wire Resistive Touchscreen Technology [2].

A capacitive touch screen panel is coated with a material that stores electrical charges. When the panel is touched, a small amount of charge is drawn to the point of contact. Circuits located at each corner of the panel measure the charge and send the information to the controller for processing. Capacitive touch screen panels must be touched with a finger unlike resistive and surface wave panels that can use fingers and stylus. Capacitive touch screens have excellent clarity, and there are no moving parts to wear out. Liquids, dirt, grease, or other contaminants do not affect them. Unfortunately, gloved fingers will not activate the system. It is divided into two broad categories as follows: 1) Surface capacitive technology and 2) Projected capacitive technology [2].

An infrared touchscreen uses an array of X-Y infrared LED and photo detector pairs around the edges of the screen to detect a disruption in the pattern of LED beams. A major benefit of such a system is that it can detect essentially any input including a finger, gloved finger, stylus or pen. It is generally used in outdoor applications and Point-Of-Sale systems which can't rely on a conductor (such as a bare finger) to activate the touchscreen. Unlike capacitive touchscreens, infrared touchscreens do not require any patterning on the glass which increases durability and optical clarity of the overall system [2].

The Surface Acoustic Wave (SAW) technology is one of the most advanced touch screen types. The technology is based on two transducers (transmitting and receiving) placed for the both of X and Y axis on the touch panel. The other important element of SAW is placed on the glass, called reflector. The controller sends electrical signal to the transmitting transducer, and transducer converts the signal into ultrasonic waves and emits to reflectors that are lined up along the edge of the panel. After reflectors refract waves to the receiving transducers, the receiving transducer converts the waves into an electrical signal and sends back to the controller. When a finger touches the screen, the waves are absorbed, causing a touch event to be detected at that point [2].

Longyu Zhang, Jamal Saboune and Abdulmotaleb El Saddik [1] developed a less expensive webcam-based touch screen system based on a simple and generic finger detection approach. Only one web camera and a screen are used to achieve their goal. By using the scene geometry and the camera model, they have determined the position of the finger on the screen. The technique they proposed is simple and works well under various surrounding environments while the detecting method suits many finger colors and shapes. Their approach can also be used for real-time applications.

R. Pradipa, Ms. S. Kavitha [3] discussed on various techniques, methods and algorithms related to the gesture recognition. Their survey mostly focused on two different techniques i.e. Glove-Based gesture and Vision-Based technique. Their proposed system consists of desktop or laptop interface, the hand gesture may be used by the users may need to wear any data glove, or may use the web camera for capturing the hand image. The initial step towards any hand gesture recognition is hand tracking and segmentation. Sensor devices are used in Data-Glove based methods for digitizing hand and finger motions into multi parametric data. The other sensors will collect hand configuration and hand movements. In contrast, the Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision by describing artificial vision systems that are implemented in software and/or hardware. The hand gesture is the most easy and natural way of communication. Hand gesture recognition has the various advantages of able to communicate with the Technology through basic sign language. The gesture will able to reduce the use of most prominent hardware devices which are used to control the activities of computer.But in Hand-Gesture recognition systems there were few limitations such as human body postures were restricted, few color recognitions, restricted Gestures, limited screen area etc.

Swapnesh Gandhi and Nikhil Digrase [4] presented a system that tracks the 2D position of the user's hands, using colour markers, on a table surface, allowing the user drawing, writing and manipulating virtual objects over this surface. Their implementation is based on a computer vision tracking system that processes the video stream of a single camera. The access of machine by mere movement of hands not only provides more flexible, natural and intuitive interaction possibilities, but also offers an economic and practical way of interaction.

In their system, the screen of a desktop or laptop is projected on a plane surface the camera processes the live video of this projected screen, in real time, where the system tries to find the presence of a coloured marker. If marker is present then mouse pointer is set at the location of marker and further a mouse click at that location can be initiated. Other mouse operations like double click and scrolling etc may also be implemented. User can use on-screen keyboard through his touch which effectively enables user to perform all tasks that he might want to do that too in a natural and intuitive way.

3 Proposed System

This section holds the detail description of the proposed system, it describes the webcam position and the advantages of that position. Then we explain the initialization calibration phase after that we explain how do we locate the finger in the captured image, after locating the finger we perform filtering operation on the image to get the noise free image for calculating real time touch position.

3.1. Webcam Position

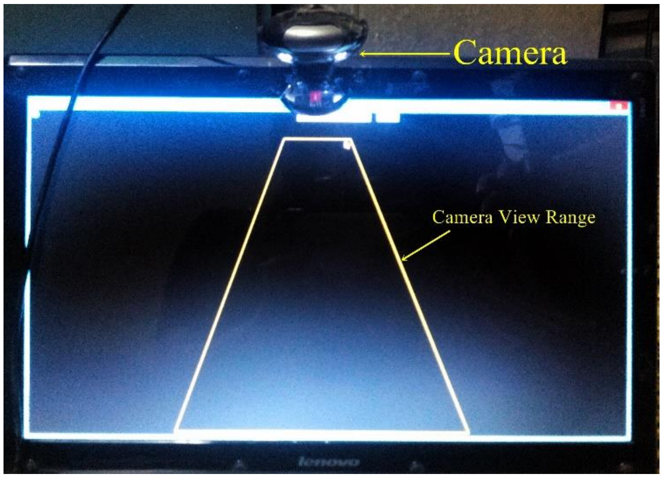


Fig 1. Webcam position and its field of view.

In our system we used a single webcam. The webcam position is important point in the system which uses it as an input devise, the web camera should be placed in such a position from where we can get good quality input, here good quality input is the input image which do not contain the noise or unwanted objects which may cause difficulty in calculating the finger position. To avoid noise as much as possible we placed our camera at the top of the screen refer fig 1. Due to this position the webcam covers the triangular region of the central part of the screen with the minimum reflection & no interference of the object in the surrounding. This helps to improve the system efficiency and makes our system to work in most of the surrounding conditions.

3.2. INITIALIZATION CALIBRATION

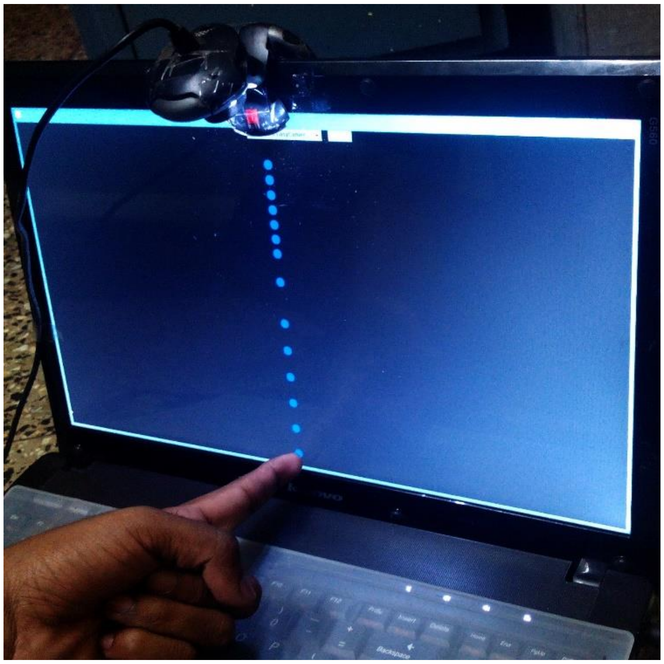


Fig 2. Calibration of the system.

The main objective of our system is to give an appropriate contact point between the screen and the finger to accomplish this objective we need to do Initialization and Calibration phase which will help the system to calculate the appropriate touch position. In our Initialization and Calibration phase we take 14 different points on the screen with their original values known to us refer fig 2 & with the help of our program we get the value of the points in the captured image.

3.3. FINGER DETECTION

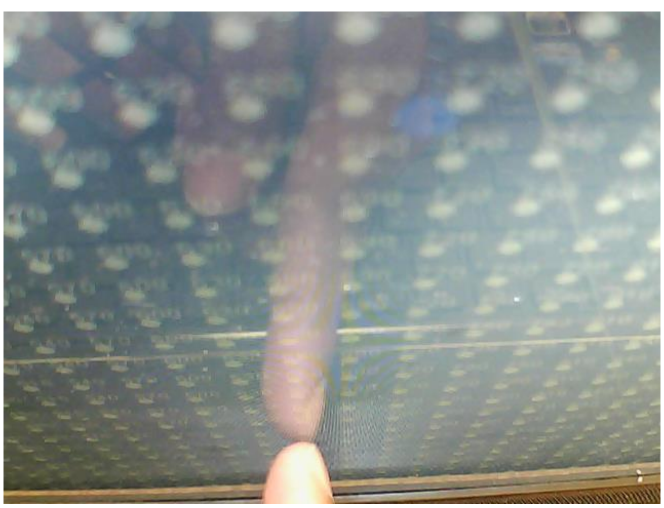


Fig 3. Captured Image.

The captured image by the webcam is send to the system refer fig 3 & the system does processing on that image to detect the finger in the image. Our system accomplish the finger detection task by executing some steps they are, firstly our system detects the skin-color pixel, then it perform filtering on the image after that noise is removed and then the finger tip is obtained in the image.

Detecting skin-color pixel:- The captured image from webcam is send to the system and the system firstly search for skin-color in the image. Our method of detecting the skin color is combination of three technique they are

I) Detecting the skin-color pixel in RGB color space. Skin color cluster in RGB color space (R,G,B) is classified as skin if [5]:
   $R > 95$ and $G > 40$ and $B > 20$ and
   $\max(R,G,B) - \min(R,G,B) > 15$ and
   $|R-G| > 15$ and $R > G$ and $R > B$
   Where, R is Red, G is Green, B is Blue.

II) Detecting the skin-color pixel in YCrCb color space. First we need to convert RGB color values to YCrCb color values. To convert RGB color values to YCrCb color values we can use following formula [9]:
   $Y = (0.257*R) + (0.504*G) + (0.098*B) + 16$
   $Cr = (0.439*R) - (0.368*G) - (0.071*B) + 128$
   $Cb = -(0.148*R) - (0.291*G) + (0.439*B) + 128.$
   Using these values we can classify (Y, Cr, Cb) as skin if [7]:
   $Cr <= 1.5862 \times Cb + 20$ AND
   $Cr <= 0.3448 \times Cb + 76.2069$ AND
   $Cr <= -4.5652 \times Cb + 234.5652$ AND
   $Cr >= -1.15 \times Cb + 301.75$ AND
   $Cr >= -2.2857 \times Cb + 432.85$ AND
   Where, Y is luma, Cr is red difference chroma component, Cb is blue difference chroma component.

III) Detecting the skin-color pixel in HIS color space. For calculating HSI color values, we can use following method [8]:
   First we need to normalize (R,G,B):
   $r = R / R + G + B$
   $g = G / R + G + B$
   $b = B / R + G + B$
   Where, R is Red, G is Green, B is Blue.

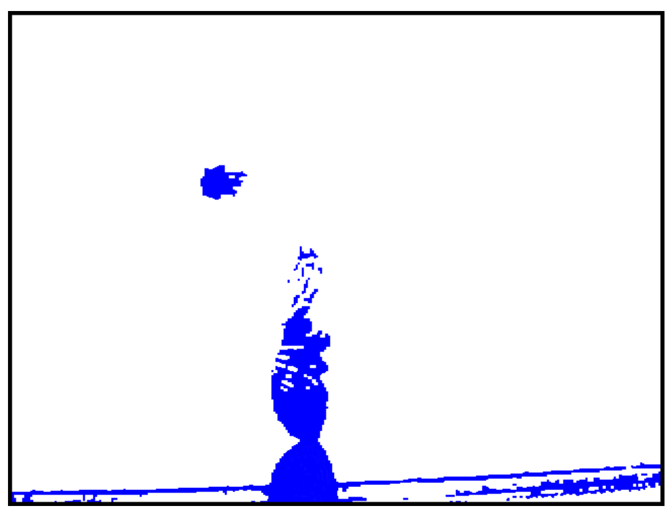


Fig 4. Binarised Image

Then we can use the values:

$h = \cos^{-1}((R-G)+(R-B)/2*\text{sqrt}((R-G)^2+(R-B)(G-B)))$

$s = 1-3*\min(r,g,b) \quad : \quad 0 <= s <= 1$
$i = (R + G + B) / (3*255) \quad : \quad 0 <= i <= 1$
$h = 360 - h, \quad \text{if} \quad (B>G)$
where, h is hue, i is intensity, s is saturation.
The skin color cluster in HSI space is simply:
$h < 25$ & $h > 230$

The image is passed through the above three skin-color detection technique and when it satisfies all the three technique then only the system recognise the skin-color refer fig 4.

3.4. NOISE-REMOVAL

After detecting the skin-color pixel in the image, then the system is converting to binary image. The binary image contain the finger in the binary form and also contain scattered grains or noise in the image with the finger. The scattered grains or noise must b reduce to make our binary image more efficient to perform further operations on it. To reduce this unwanted scattered grains we perform certain morphological operations on the image such as: Erosion and Dilation.

I) Erosion:-

Erosion is one of the two basic operators in the area of mathematical morphology. It is typically applied to binary images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels Thus areas of foreground pixels shrink in size, and holes within those areas become larger. By using this technique we are able to remove the scattered grains in the binary image.

II) Dilation:-

Dilation is one of the basic operations in mathematical morphology. Originally developed for binary images, the dilation operation usually uses a structuring element for probing and expanding the shapes contained in the input image. By using erosion we are able to remove the scattered grains but it also affect the finger size, finger size get reduce as the finger is shrinked. To repair our shrinked finger we perform dilation on the binary image.

III) Reflection removal

After removing the scattered grains we now need to remove the reflection of the finger which was detected by the skin color detection techniques, this reflection also survive our morphological operation. To remove this reflection we use our own technique. The binary image is scanned from left to right image and when the first color pixel is detected the system notes that point. Then the system start scanning from right to left and when it finds the first color pixel while scanning it notes its position of that point and then the difference between both the points is calculated that difference is our finger width. The system continues this procedure of calculation of width of the finger while moving up word in the image. At certain point the width of the finger will start decreasing as our fingers get narrower as we move towards the tip. After reaching a certain minimum value of the finger width the width will start increasing due to reflation. The system will observe the increase in the width size after reaching to certain minimum value. The system will remove all the color pixels above that minimum value of the width. This is how we get a reflection free image to perform further operations on it refer fig 5.

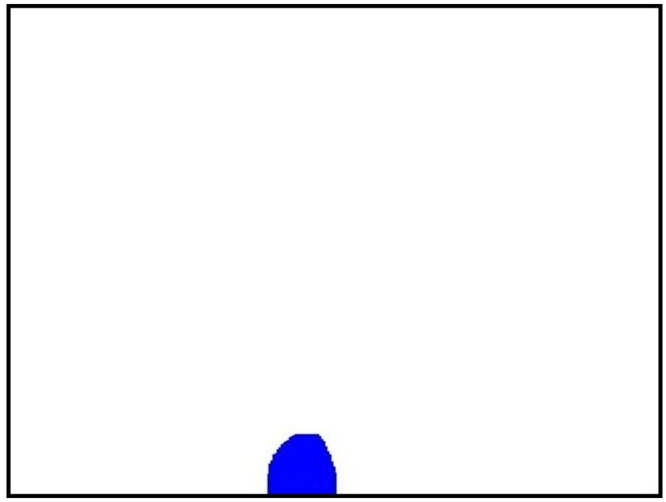


Fig 5. Noise-Free Image.

3.5. MATHEMATICAL CALCULATIONS

To calculate the actual touch point we performed some mathematical operation on our noise free binary image. To get the original touch point from the points obtained from that image, we calculate the image points and process it using our developed formula and we get the original screen points.
We developed the formula by using the points calculated from the calibration phase and the points obtained from the noise free image given by our system.

I. Calculating Y Coordinate

The formula to calculate Y-coordinate of any point on the screen works in a particular rang of pixel, the formula is as follows:

$$Y= ((PCM + ((IY - CV) * SPM)) * IY)$$

Where,

a) CV, is Calibration Value, it is the value of previous point from the current calibrated point.
b) IY, is the Y coordinate of the fingertip obtained from image generated by our system.
c) PCM, is Previous Calibration Value Multiplier, it is the multiplier for the Previous Calibration Value which maps the fingertip pixel of image to the original pixel on the screen. The multiplier of the calibration point is obtained by dividing the image pixel value by the original, screen pixel value.
d) SPM, is the Single Pixel Multiplier, it is the multiplier value for every single pixel in the specified range, which will map our fingertip pixel in the image with the original screen pixel. To calculate SPM firstly we take the difference between two calibration points(i.e the range of pixel for which this formula will work)let us name the difference as “DC” and then take the difference between the multiplier of those two calibration

point and name it is “DCM”. Now to obtain SPM by dividing DCM by DC.

II. Calculating X Coordinate

Once the Y-coordinate is calculated, the X-coordinate can be calculated by applying the X coordinate formula to the same pixel range which was taken while calculating Y coordinate. The formula to calculate X coordinate is as follows:

$$X = ((IX * MX) + SP)$$

Where,

a) IX, is the X coordinate of the fingertip obtained from image generated by our system.
b) SP, is the Pre Define Screen Point Value, Our camera has a triangular view so SP is the starting point of the camera’s view in the given range of pixels specified while calculating Y coordinate.
c) MX, is the X coordinate multiplier, to obtained MX we first need to find camera’s view range on the display, it is found by taking the difference of starting point and end point of the camera’s view on the display in the given range of pixels specified while calculating Y coordinate. Let us denote this difference by “D”. Now by dividing this D by the width of the image generated by our system will give you MX.

3.6. ALGORITHM.

This algorithm is stepwise representation of our proposed system’s flow

1. Capture frame using the camera.
2. Rotate, Resize according to standard image resolution.
3. Detect Skin Cluster in the image.
4. Convert the Image Into Binary Image.
5. Remove the noise if exist.
6. Find the fingertip in the image and obtain the Location of it.
7. Perform Mathematical Calculation and Obtain actual location of the finger tip on the screen.
8. Plot the location to show touch point.

3.7. ARCHITECTURE

The Explaination of our architecture refer fig. 6 is as follows. The user interacts with the program by touching his finger on the screen, webcam captures frame and this frame is given as input to the program and the program performs Resizing , Rotation, Binarization , Noise Removal, Mathematical calculations and finally it gives the touched point as output on the screen.

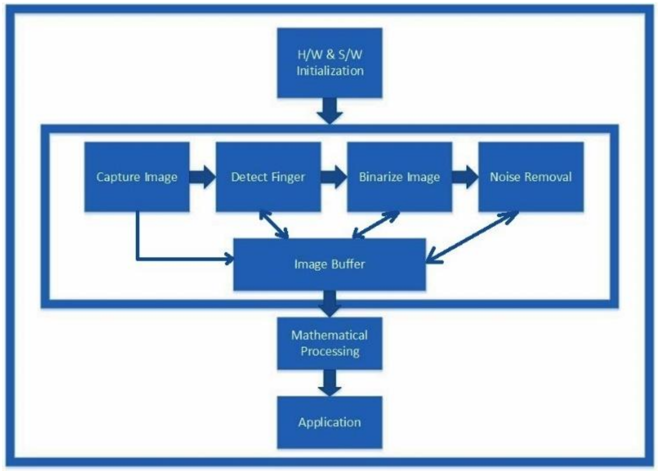


Fig 6. Architecture of the System.

3.8. APPLICATION

To show the implementation of our system we have developed an application called Paint application refer fig. 7(a) and 7(b). Our system could be used in applications such as drawing we can draw with our system in the triangular area of the centre of the screen. By adding few buttons such as “start” “stop” “clear background” could make the application more interactive & various types of small games could also be played on our system.

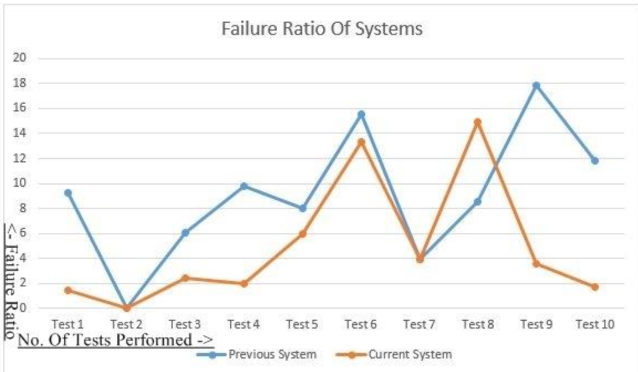
3.9. RESULTS

User No.	No. Of Test Points	Surrounding Light Condition	Successful Test Points	Unsuccessful Test Points	Detection Ratio Failure (%)	X-Coordinate Accuracy (%)	Y-Coordinate Accuracy (%)
1	151	Dim	149	2	1.32	98	95
2	60	Dim	60	0	0	99	95
3	33	Bright	25	8	2.42	85	80
4	51	Dim	50	1	1.96	97	96
5	50	Dark	47	3	6	97	94
6	45	Bright	39	6	13.33	88	82
7	51	Dim	49	2	3.92	90	94
8	47	Bright	40	7	14.89	84	83
9	56	Dark	54	2	3.57	89	97
10	59	Dim	58	1	1.69	93	95

Table 1. Accuracy of calculated touch positions.

This Section shows results while execution the system in different conditions and calculating the accuracy of the system and comparing it with previous system refer Table 1 and Graph 1.
To test the accuracy of our touch screen system, we asked 10 users to touch the screen and check if it is giving accurate output or not. By calculating X and Y coordinates of touch point’s position on the screen, we draw them on the same screen using blue points and we also tested on different light conditions like dim light,

bright light and in dark also.



Graph 1. Comparative study of two systems.

3.10. Evaluation Of Work

Our system works on various applications but we can improve our system in following fields:

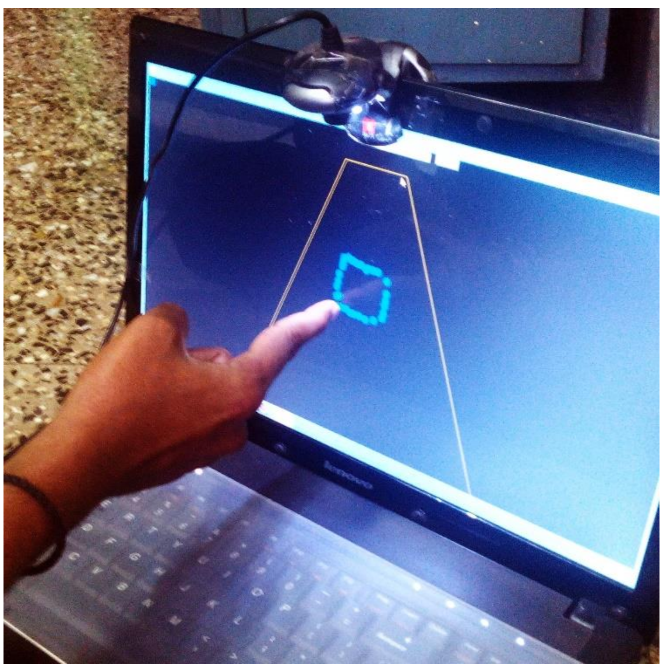


Fig 7(a). Paint Application.

1) The view of our webcam covers only the triangle area instead of the whole screen. Raising position of the webcam will able to solve this problem, or using two cameras sideways but extra efforts will be required to fix the webcam, and little changes in the program which is against our idea of proposing a simple and generic system.

2) Currently our proposed system works only with a single finger, supporting multi-touch will make it more interactive but the main problem is that as we use just a single webcam, if fingers overlap from the field view of the webcam, and then actions will not be detected because our method detects the finger's position using skin colour properties.

3) Compared with existing touchscreen systems, our system's accuracy and process time can still be improved, as well as minimizing restrictions for finger poses, but its low cost highly compensates the drawbacks, and the ability to transform normal screens also increases its competitive power. Despite the limitations mentioned above, our proposed system is still in practical use with applications which requires just a part of the monitors, acceptable update rate and accuracy e.g paint applications.

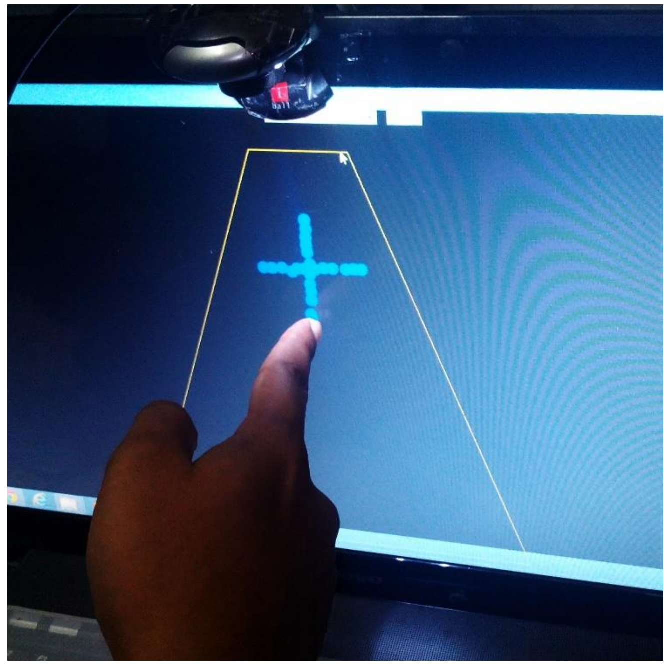


Fig 7(b). Paint Application.

3.11. Hardware Used

- Processor: Intel® Core™ i7-3610QM.
- Ram: 6GB.
- Hard Disk: 750 GB.
- Monitor: Regular monitor.
- Camera: iBall C8.0 Face2Face.

3.12. Software Used

- Language: Java.
- Tools: Eclipse with OpenCV.
- Platform: Windows 8.1.
- Documentation: MS Office.

4 Conclusion

In this paper, we have demonstrated a touch screen system with a single webcamera and a normal screen, and developed a practical application like paint application with this touch screen system. Our system works well with different backgrounds due to the camera position. Furthermore, the finger detection method we adopt suits many finger colours and shapes, and do not need any learning or classification step.

5 FUTURE WORK

Our proposed system is able to convert normal screen into touchscreen our future work will focus on improving screen area coverage and the accuracy in locating the exact position on the screen irrespective of the surrounding environment and also will focus on improvemet in the processing speed of our system and reducing the calibration phase as much as possible.

6 ACKNOWLEDGEMENT

We are thankful to Prof. Deepak B. Kanthale for his guidance in our project development and we are also thankful to the management of our college as they have provided us the best environment for the development process.

REFERENCES

[1]. Longyu Zhang ; Saboune, J. ; El Saddik, A. "Transforming a Regular Screen Into a Touch Screen Using a Single Webcam" in Journal of display technology, vol. 10, no. 8, august 2014 pp.1551-319X © 2014 IEEE.

[2]. Mudit Ratana Bhalla, Anand Vardhan Bhalla"Comparative Study of Various Touchscreen Technologies"IJCA'10.

[3]. R.Pradipa, Ms S. Kavitha -"Hand Gesture Recognition-Analysis of various techniques, methods and algorithms".- IJIRSET 2014 ISSN: 2319-8753 ICIET'14.

[4]. Swapnesh Gandhi, Nikhil Digrase -"Interactive Human Machine Interface through Virtual Touch Screen" publishresearch'12

[5]. Jure Kovac, Peter Peer, and Franc Solina -"Human Skin Colour Clustering for Face Detection"

[6]. Vladimir Vezhnevets, Vasili Sazonov, Alla Andreeva – "A Survey on Pixel-Based Skin Color Detection Techniques"

[7]. Nusirwan Anwar bin Abdul Rahman, Kit Chong Wei and John See – "RGB-H-CbCr Skin Colour Model for Human Face Detection"

[8]. Conversion from RGB to HIS Ruye Wang at http://fourier.eng.hmc.edu/e161/lectures/color_processing/node3.html

[9]. RGB to YCbCr http://www.fourcc.org/fccyvrgb.php

[10]. Java [Online] Available: http://www.tutorialspoint.com/java_technology_tutorials.htm.

[11]. MS Office [Online] Available: http://office.microsoft.com/en-in/visio-help/.

AUTHOR'S BIOGRAPHY

1. Kuntal B. Adak
recieved the Bachelor of Engineering in the Department Of Computer from University of Pune, India, in 2015. Email ID– koontal@live.in.

2. Adarsh N. Singh
recieved the Bachelor of Engineering in the Department Of Computer from University of Pune, India, in 2015. Email ID– singhadarsh027@gmail.com.

3. Abhilash B. Kamble
recieved the Bachelor of Engineering in the Department Of Computer from University of Pune, India, in 2015. Email ID– abhilashbk24@yahoo.com.