

# Virtual Mouse

Dikshant Rathod, Kanchan Kadlag, Savita Anarase, Poonam Gavade

**Abstract:** In this paper we present an approach for transforming normal LCD into touch screen. This will be done by using single Web Cam which will be filming the LCD and detecting motion. That motion will then be transformed into adequate action like pressing left mouse button or emulating double left click. Application works by mapping Web Cam coordinates into LCD coordinates. This means that when Web Cam detects motion it knows its coordinates inside the image it has taken. Those coordinates are then transformed into LCD coordinates by using set of pre-calculated coordinates of the dots which are displayed during initialization phase. Motion is detected by using ski detection on normalized RGB colour space. This method mainly focuses on the use of a Web Camera to develop a virtual mouse in a cost effective manner.

**Keywords:** Color Detection, Web Camera, Touch Screen.

## I. INTRODUCTION

To transform the normal LCD into touch screen, this will be done by using single Web Cam which will be filming the LCD and detecting motion. Initialization is done by displaying three pictures on LCD. LCD will first fill with single color. This will be later used to distinguish area which should be turned into touch screen from the rest of the LCD. This step presumes that there is no. of such color behind LCD. Then area is filled with lines. These lines will be later used to detect dots. Each line is used to define which dots belong to the same vertical coordinate. Then area is filled with dots. This picture is center element in initialization process since the goal of initialization is to determine Web Cam & LCD coordinates of these dots. After all initialization screens were displayed initialization part of application minimizes and LCD then shows whatever was on it before starting initialization.

## II. DESIGN

### Algorithm

#### PIXELS\_TRESHOLD\_FILLGAPS

This function will set each empty pixel if number of set pixels in its neighborhood is greater or equal to minNeighbours.

```
static public int[][] pixels_Treshold_FillGaps(int[][]
pixels, int minNeighbours)
{
    int filledNeighbours = 0;
    int lines = pixels .length;
    int rows = pixels[0].length;

    int[][] pixelsFilled = new int[lines][rows];

    for(int l=1;l<lines-1;l++)
    {
        for(int r=1;r<rows-1;r++)
```

```
{
    if (pixels[l][r]==1) { pixelsFilled[l][r] = 1; }
    else
    {
        filledNeighbours=0;
        if (pixels[l-1][r-1]==1) { filledNeighbours++; }
        if (pixels[l-1][r ]==1) { filledNeighbours++; }
        if (pixels[l-1][r+1]==1) { filledNeighbours++; }
        if (pixels[l ][r-1]==1) { filledNeighbours++; }
        if (pixels[l ][r+1]==1) { filledNeighbours++; }
        if (pixels[l+1][r-1]==1) { filledNeighbours++; }
        if (pixels[l+1][r ]==1) { filledNeighbours++; }
        if (pixels[l+1][r+1]==1) { filledNeighbours++; }
        if (filledNeighbours>=minNeighbours)
        { pixelsFilled[l][r] = 1; }
        else { pixelsFilled[l][r] = 0; }
    }
}
}
Return pixelsFilled;
}
```

## III. IMPLEMENTATION

### A. Displaying Initialization Screens

Initialization is done by displaying three pictures on LCD. LCD will first filled with single color. This will be later used to distinguish area which should be turned into touch screen from the rest of the LCD. This step presumes that there are no of such color behind LCD. Then area is filled with lines. These lines will be later used to detect dots. Each line is used to define which dots belong to the same vertical coordinate. Then area is filled with dots. This picture is centre element in initialization process since the goal of initialization is to determine Web Cam & LCD coordinates of these dots. After all initialization screens were displayed initialization part of application minimizes and LCD then shows whatever was on it before starting initialization.



Fig 1: LCD is filled with single color

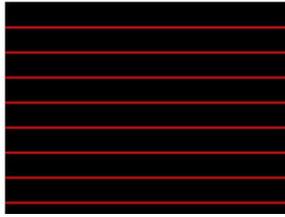


Fig 2: LCD is filled with horizontal lines

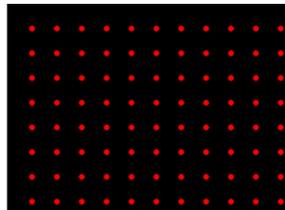


Fig 3 : LCD is filled with dots

### B. Capturing Initialization Screens

Each of these pictures is captured by Web Cam. Web Cam captures not only LCD but also parts of the environment which are outside LCD. After capturing initialization screens each of them is then transformed. Each image goes through different steps which can all be seen in an additional frame which is displayed at the end of initialization process as collection of tabs.

### C. Analyzing Screen Containing Background

Image taken when the whole LCD was in one colour, This image is then used to create new image, using 2D array, in the way that pixels in the new image are set to 1 only if equivalent pixel in original image has value of all RGB components greater than some predefined value. After that clustering is used to identify cluster with most pixels. This way background noise is eliminated. Last step is to invert threshold image. Now all pixels that are 1 present background.

### D. Analyzing Screen Containing Lines

Image taken by the Web Cam while the area was filled with lines, resulting image is then transformed by removing background. Then an algorithm for filling gaps is used. Then the lines are trimmed from above and below until their height is reduced to one pixel. Such image is then used to detect each line. This is done by going vertically through image from top to bottom at horizontal position in the middle of the image. Points at which lines were detected are shown as yellow dots at the middle of the lines. Now an algorithm is started which goes from that point to the left and right following line pixels until left and right line end are reached which are highlighted with yellow dots at the end of the lines. This shows how detected lines correspond to the original lines in the image taken by Web Cam.

### E. Analyzing Screen Containing Dots

Image taken by the Web Cam while LCD was filled with dots, Resulting image is then transformed by removing background. Then an algorithm for filling gaps is used. Dots are detected by starting at the left end of each line,

following line pixels to the right and detecting when dot is reached in the image with dots. Web Cam coordinates of each dot are saved and are used later for mapping to LCD coordinates. This shows how detected dots correspond to the original dots in the image taken by Web Cam. Additionally this shows effects of dilating dots to remove small gaps which might negatively influence dots detection. It demonstrates how the dots are actually being detected. Algorithm starts at left end of each line, follows the line to the right, detects when it has entered and exited the dot and uses these two values to approximate centre of the dot. This is how the white dots are calculated.

### F. Using LCD as touch screen

Now that initialization is done we can start using LCD as touch screen. By using finger we set cursor position. While finger is moving around nothing will happen. When finger becomes stationary left mouse click will be initiated. If finger is not moved after left click was done double left click will be initiated. Web Cam constantly takes images of LCD. Such image is then transformed into normalized RGB colour space. Normalized RGB coordinates contain only colour information, unlike RGB coordinates which also contain intensity information. Such image is threshold using skin colour as threshold parameter. After that isolated pixels are removed from the image, small gaps in the image are then filled. Algorithm now searches for different cluster and takes only those that have more pixels than some predefined value. In this particular case this will result in two big clusters. One belongs to the hand and the other to the brown wall on the left and behind the LCD. Each of these two clusters is now taken separately and background is erased. In the case of wall cluster result is empty image. Resulting image is now analyzed line by line from top to bottom going left to right at each line until first non zero pixel is reached. Coordinates of such pixel are taken as finger coordinates. Such coordinates are being memorized and if speed of the finger drops below predefined threshold clicking on left mouse button will be emulated.

## IV. ANALYSIS

Our problem is sequential because output of previous process is given as input to next process. And our problem can be divided into sub problem. Solution of sub problem gives ultimate solution to main problem in deterministic polynomial time, hence it NP-Complete type of problem.

### Time Complexity Calculation

1. for Image Threshold  
Complexity: -  $t = O(n^2)$   
Where n = Size of image
2. for Contour image detection  
Complexity: -  $O(n^2)$   
Where as n = size of the contour image
3. for Bounding Box  
Complexity: -  $O(n)$

4. for Average color

Complexity: -  $O(p*q)$

Where as  $p$  = no. of rows,  $q$  = no. of columns

**TOTAL Complexity:**  $(2n^2+n+p*q)$

## V. CONCLUSION

This problem can be converted into polynomial-time-reducible as there is a sequential way to solve the problem, so problem is NP-Complete.

## ACKNOWLEDGMENT

The authors would like to thank the guide of our project – Mr. Sachin Godse, for his encouraging discussions. We are also very grateful to the project in charge – Prof. Gandhali Kulkarni and to the H.O.D of computer department – Prof. B.B. Gite for their unending support and guidance.

## REFERENCES

- [1] 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.
- [2] Hojoon Park, “A Method for Controlling the Mouse Movement using a Real Time Camera”, Brown University, Providence, RI, USA, Department of computer science, 2008.
- [3] Chu-Feng Lien, “Portable Vision-Based HCI – A Real-time Hand Mouse System on Handheld Devices”, National Taiwan University, Computer Science and Information Engineering Department.
- [4] Kamran Niyazi, Vikram Kumar, Swapnil Mahe, Swapnil Vyawahare, “Mouse Simulation Using Two Colored Tapes”, Department of Computer Science, University of Pune, India, International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2, March 2012.
- [5] K N. Shah, K R. Rathod and S. J. Agravat, “A survey on Human Computer Interaction Mechanism Using Finger Tracking” International Journal of Computer Trends and Technology, 7(3), 2014, 174-177.