# Virtual Mouse using Hand Gesture and Color Detection

Amardip Ghodichor
Engineer
Hella India Automotive Pvt Ltd
Baner, Pune 411045

Binitha Chirakattu
Department of Electronics and Telecommunication
Government College of Engineering
Amravati

## ABSTRACT

In today’s technological era, many technologies are evolving day by day. One such promising concept is Human- Machine Interface. For example, in a wired mouse there is no provision to extend limit. In wireless mouse, one should have Bluetooth hardware installed in the computer and Bluetooth dongle attached. The proposed technology will have no such limitations and will instead depend on gesture recognition. In this project, three technologies are mainly used: object detection, image processing and color recognition using ‘Sixth sense technology’. Sixth sense technology is a set of wearable devices that acts as a gestural interface between the physical and digital world. The aim is to move the mouse cursor on the screen without using hardware such as a mouse and only by moving the cursor through finger movements i.e. the process of gesture recognition. In this paper, we present a novel approach for Human Computer Interaction (HCI) where cursor movement is controlled using a real-time camera.

## Keywords

HCI, Ubiquitous Computing, Background Subtraction, Skin Detection, HSV Color Model, gesture controlled mouse.

## 1. INTRODUCTION

In this project, the hand movement of a user is mapped into mouse inputs. A web camera is set to take the live video continuously and then from this video various images are being captured by using MATLAB. The user must have a particular color marker or pointer in his hand so that when the web camera takes an image it must be visible in it. This color is detected from the image pixel in MATLAB and object detection is used to map pixel position into mouse input. Taking into account the size of image and resolution, scaling techniques need to be used like image processing or segmentation. Sixth sense technology can be further developed to include multimedia services like videos or saving images using hand gestures. This work aims to provide usual mouse operations using a camera based on colour detection technique. The user wears coloured gloves, the information from which is processed by the system. Individual frames of the video are separately processed. The processing techniques involve an image subtraction algorithm in MATLAB to detect colours. Once the colours are detected the system performs various operations to track the cursor and performs control actions which are explained in further sections. No additional hardware is required by the system other than the standard webcam which is provided in every laptop or computer.

## 2. COMPONENTS

The components used in this project can’t be specific, since this project is a prototype for all computers. As such, certain prerequisites are as follows:

### 2.1 Webcam

Webcam is a necessary component for detecting the image. Sensitivity of mouse is directly proportional to resolution of camera. If the resolution of camera is good enough, an enhanced user experience is guaranteed. The webcam serves the purpose of taking real time images whenever the computer starts. On the basis of gestures and motion of fingers, system will decide the respective action.

### 2.2 Hand Gloves

Hand gloves as shown in figure 1 with a color is a necessary component here. It is essentially the first step of the project. The gloves will have different colors painted on different fingers. This clear distinction in colors will make the system faster and easier for the algorithm.

## 3. TOOLCHAIN FOR FIRMWARE

### 3.1 MATLAB

MATLAB version greater than 2012a is used for making this project. Image Acquisition toolbox and Image processing toolbox is necessary for developing the firmware. Along with this, an updated version of Java is required.

### 3.2 SDK tool and .NET Framework

SDK and .NET is required in order to create standalone applications for windows based system.

## 4. SYSTEM DEVELOPMENT

In the object tracking application one of the main problems is object detection. Instead of finger tips, a color pointer has been used to make the object detection easy and fast. A circle blue sticker is used as a color pointer in this study. To simulate the click events of the mouse, three fingers serving as three color pointers has been used. The basic algorithm (see figure 1) is as follows:

- Set a pointer in the image
- Detect the pointer using the defined color information
- Define the region and the center of the pointer and draw a bounding box around it
- Track the motion of the pointer
- Move the cursor according to the position of the center of the pointer
- Simulate the single and the double left click and the right click of the mouse.

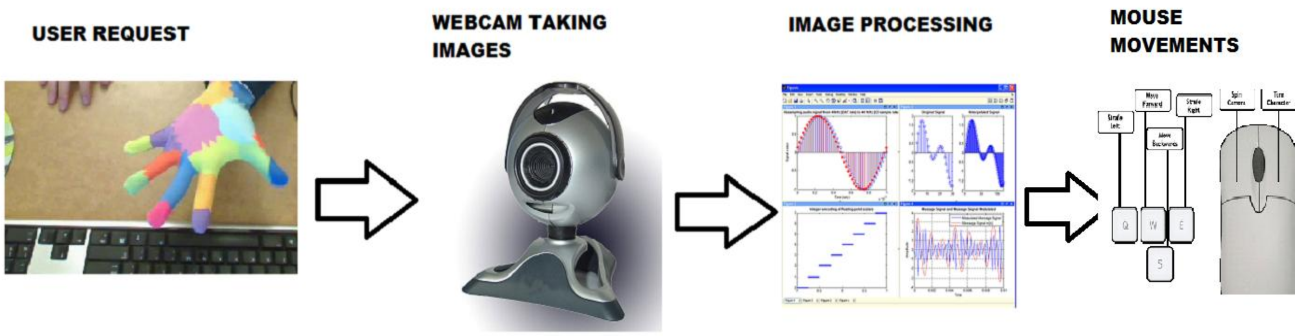

**Fig 1: Overview of system**

## 4.1 Color detection

In this project, there are uses of blue and red planes (see figure 2, figure 3 and figure 4). In order to identify the blue color of the hand, Matlab built in function ‘imsubstract’ can be used.

Z = imsubtract(X,Y)

where this function subtracts elements from one array to another which is specified as X and Y and gives the difference in the corresponding element to the output array Z.

X and Y are real, non-sparse numeric arrays of the same size.

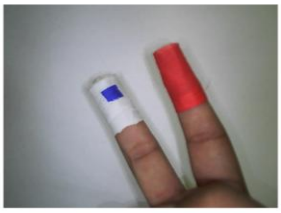

**Fig 2: Input Image**

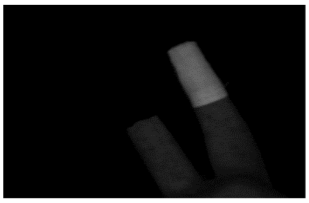

**Fig 3: Red plane detected**

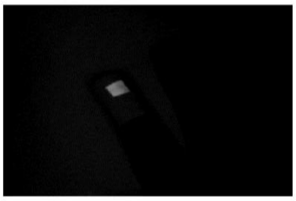

**Fig 4: Blue plane detected**

## 4.2 Filtering the Images

As soon as the blue color in the real time image is detected, the next step is to go for filtering of this single frame. Care has to be taken about the processing speed of every frame since every camera has different frames per second.

Median filtering gives optimum results for such operations. Result of the filtering should look like as mentioned in Fig 5. Median filtering is basically to remove “salt and pepper” noise. Although convolution can be used for this purpose, a median filter is more effective when the goal is to reduce noise and preserve edges. Illumination is major part while taking the real time images. There is not much noise when the illumination is high, as seen from the images.

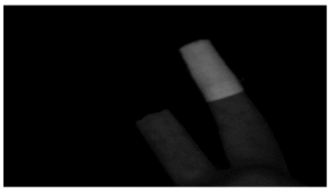

**Fig 5: Red color filter**

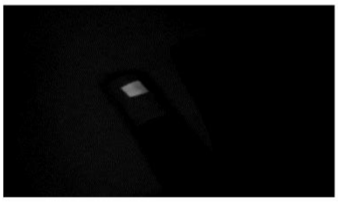

**Fig 6: Blue color filter**

## 4.3 Conversion of Images

As soon as filtering is done over a frame, next step is to convert an image. For conversion of image one may also use in built function ‘im2bw’.

The function can be used as

BW = im2bw(I, level)

where I is the image. The BW image replaces all pixels in the input image with some threshold greater than the Value 1 with the value 1 (white) and replaces all other pixels with the value

0 (black). Level should be in range from 0 to 1. This level is very much understood since the image itself is binary levels not greater than 1. Therefore, a level value of 0.5 is midway between black and white color, regardless of class. The threshold 0.15 gave the best result for the large range of illumination.

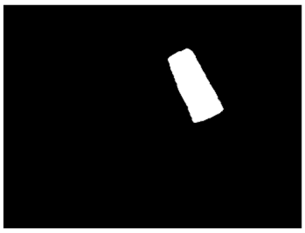

**Fig 7: Red converted to BW**

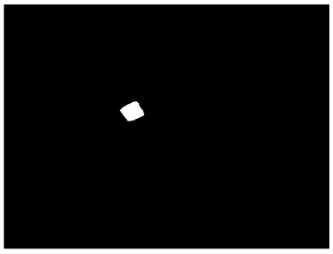

**Fig 8: Blue converted to BW**

## 4.4 Removing small areas

After the filtering and conversion the next step is to remove small areas from images. To do this one can use MATLABs 'bwareaopen'function.

Function can be used as

BW2 = bwareaopen(BW, P)

Where BW is image and it will remove all the pixels which are less than P pixels, producing another binary image, BW2. Selecting best Pixel value depends upon the illumination again.

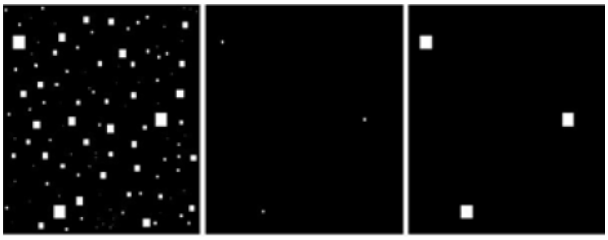

**Fig 9: Removal of small areas**

## 4.5 Find Centre

In order to make more precise pointer of mouse finding centroid is necessary. Here 'bwlabel' 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., MATLABs 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.

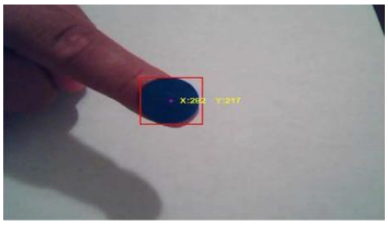

**Fig 10: Detected Center**

## 4.6 Move the cursor

Movement of cursor is the last step where actual decision has to be taken. By seeing the centroid from above image, movement has to take place. To move the cursor to desired (X,Y) coordinates, MATLAB has a set(0,'PointerLocation',[x,y]) function. Matlab doesn't provide any function for the clicking events. To move the mouse and to simulate the mouse click event Java class java.awt.Robot which has all these abilities can be used. Resolution of camera is directly proportional to the resolution of mouse pointer. Therefore using better quality camera will be more beneficial. In figure 11, the resolution of the input image was 640x480 and the resolution of the computer monitor was 1280x800. In case if the resolution of camera is less than monitor screen then scaling should be used.

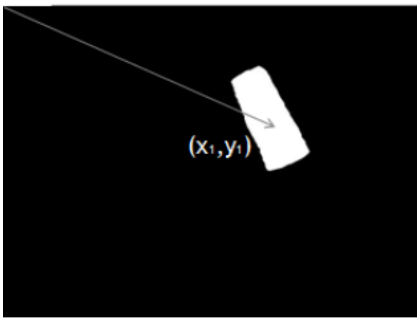

**Fig 11: Move as per co-ordinates**

## 4.7 Mouse click event

Clicking could be the challenging task since Matlab doesn't provide any function for this. In usual mouse operation the left button of the mouse performs a different task for single click and double click. There are several ways to do this .One of the methods is to use three pointers and if those pointers are detected decide the clicking events depending on the time that the pointer is being detected. For movement use all pointers and for clicking use two pointers. In case user wants to go for left click one pointer should be less which is from right side and same is the case for right click.

## 4.8 Erosion and dilation

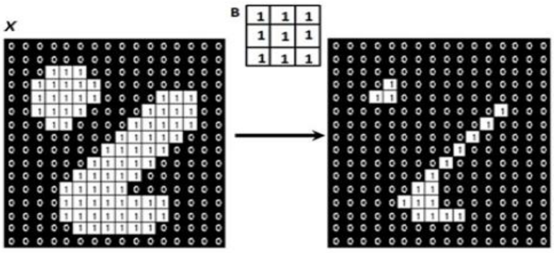

**Fig 12: Erosion**

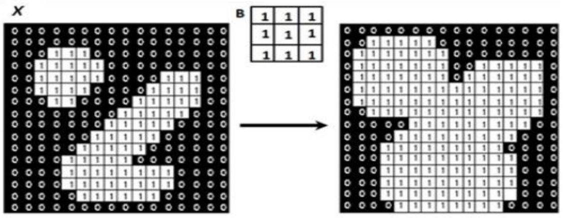

**Fig 13: Dilation**

Morphology can be said to be a set of image processing operations that processes images based on their shapes. A structuring element is applied to an input image that gives a similar sized output image. The respective input image is compared with its neighbors, thereby giving a value for the output image. The factors of the neighborhood such as shape and size can be decided by the programmer, thereby constructing programmer-defined morphological operations for the input image. The most basic morphological operations are dilation and erosion (see figure 12, figure 13). Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. According to the structuring element used, the number of pixels added or removed will differ. In the morphological dilation and erosion operations, the features of any pixels in the output image is determined by applying a specific rule to the corresponding pixel and its neighbors in the input image. Whether the operation is dilation or erosion can be determined from the rule that has been applied to the image pixel.

## 5. PERFORMANCE ANALYSIS

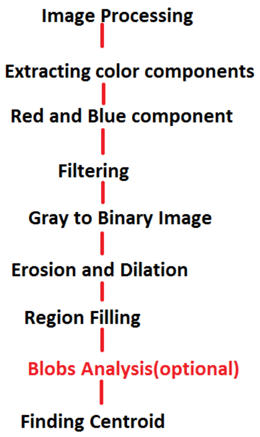

**Fig 14: Image processing part sequence**

Figure 14 is only the representation of Image processing part in the firmware. Above diagram explains how single shot of the image processes throughout. Every image taken from real time camera has to go from the above mentioned processes. Every real time camera has its own frames per second. In a certain video every single frame has to go from this process.Figure 15 is the overall system flow. System architecture of this project is already mentioned in Figure 1. This is basically a software architecture which helps decide how to process the image and how to take the required decisions. The procedure from capturing image to performing demanded action can be seen from the design,

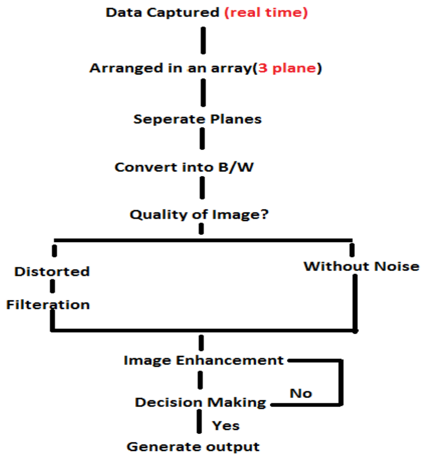

**Fig 15: Software Architecture design**

## 6. CONCLUSIONS

The system architecture that has been proposed will completely change the way people would use the Computer system. Presently, the webcam, microphone and mouse are an integral part of the Computer system. This project will completely eliminate the necessity of mouse. Also this would lead to a new era of Human Computer Interaction (HCI) where no physical contact with the device is required. The use of object detection and image processing in MATLAB for the implementation of our proposed work proved to be practically successful and the movement of mouse cursor is achieved with a good precision accuracy. 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. The motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system. Various application programs can be written exclusively for this technology to create a wide range of applications with the minimum requirement of resources.

## 7. REFERENCES

[1] Chang S. L., Chen L. S., Chung Y. C. and Chen S.W, 2004, "Image Recognition", IEEE Transactions on Intelligent Transaction Systems, Vol. 5, No.1, pp. 42-57.

[2] Kate R., 2012, "Color Recognition Using Segmentation", International Journal of Engineering Research and Technology (IJERT), Vol. 1, No. 9.

[3] Du S., Ibrahim M., Shehata M. and Badawy W., 2013, "Image Recognition: A State of the Art Review", IEEE Transactions on Circuits And Systems For Video Technology, Vol. 23, No. 2, pp. 311-325.

[4] Chhoriya P., Paliwal G., Badhan P., 2013, "Image Processing Based Color Detection", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 4, pp. 410-415.

[5] Khalil M. I., 2010, "Red Color Recognition Using the Template Matching Method", International Journal of Computer Theory and Engineering, Vol. 2, No. 5, pp 683-687.

[6] Koval V., Turchenko V., Kochan V., Sachenko A., Markowsky G., 2003,“Gesture Recognition System Based On Image Processing Using Neural Network”, IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 123-127.

[7] Dramola S. A., Adetiba E., Adoghe A. U., Bdejo J. A., Samuel I. A. and Fagorust T., 2011, “Automatic Gesture Identification System”, International Journal of Engineering Science and Technology (IJEST), Vol. 3, No. 2, pp. 1712-1719.

[8] Oliveira Neto F. M., Han L. D, G., Jeong M. K., 2013, “An Online Self Learning Algorithm For Gesture Recognition”, IEEE Transactions on Intelligent Transportation Systems, Volume 5, Issue 4.

[9] A Book of Digital Image Processing and Analysis by B. Chandra and D. D. Majumdar.

[10] A Book of Digital Image Processing by F. C. Gonzalez and R. E. Woods.

[11] A Book of Digital Image Processing (Implementation using MATLAB) by W. K.Pratt.