

Virtual Mouse-Implementing Mouse in Software

Nitish A.*

Abstract

As technology is advancing, the number of devices being used for a specific purpose is reducing. Implementing any hardware device in software will be of great advantage in this context, which also accounts for economy. One of them can be the mouse implemented in software.

In this paper, we propose a method to control cursor and perform mouse operations in real time using a camera. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques serves as an alternative way for the touch screen technology. The approach followed here is mere software programming which reduces the use of hardware required. It provides the familiarity of interaction without actually the need of a real hardware mouse.

Here, we propose an alternative for hardware mouse using the principles of image processing, mainly the color detection. The objects are detected based on their colors and are tracked to perform corresponding mouse operations assigned.

Keywords: Color Detection, Digital Image Processing, RGB, HSV, Morphological Operations, Contour, Reference Moment

1. Introduction

Our matter of emphasis here is the reduction of hardware use, meanwhile providing the same user experience and efficiency as done by the hardware, to perform any task on a computer.

Many approaches have been made to replace the hardware mouse previously. Few of them use an alternative hardware in addition to the webcam and few are implemented in MATLAB. The approach incorporated here is based on software using object oriented programming and few digital image processing principles, i.e., color detection and object recognition. The hardware used here is the webcam, which captures image frames for further processing.

In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Digital image processing is the use of computer algorithms to perform image processing on digital images. A digital image is a numeric representation (normally binary) of a two-dimensional image. Depending on whether the image resolution is fixed, it may be of vector or raster type. By itself, the term “digital image” usually refers to raster images or bitmapped images.

A digital image allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems. But our emphasis here is processing two dimensional images.

2. Organization

The paper is organized as follows: In section 3 we explain the System Architecture.

* Student, Department of Computer Science, Visvesvaraya Technological University, Belgaum, Karnataka, India.
E-mail: nitish.achar@gmail.com

In section 4 it is dealt with Frame Capturing and Image Conversion, followed by Image Filtering in section 5. In section 6 we deal with Object Recognition and Tracking. In section 7 it is dealt with Mapping and Mouse event Generation. Finally the paper is concluded in section 8.

3. System Architecture

The design of the system under discussion is as shown in figure 1. The process is divided into seven prime phases, where the given object in the input image is used to perform a desired mouse operation.

Suitable memory management operations are implemented in the process, wherever necessary, keeping in mind both the time and space complexity required for the process.

4. Frame Capturing and Image Conversion

To begin with, the webcam connected to the computer must be activated. If the computer is connected with more than one webcams, any of them can be selected for use. As the camera starts capturing the image frame (RGB) it is stored continuously in an image container and is processed.

The stored RGB frames cannot be efficiently processed to binary image. Hence, an intermediate representation called HSV (Hue, Saturation and Value) is used, which provides an easy way for image conversion to binary format. HSV model corresponds to RGB model with associated light intensity (Saturation). Hence, same shade of RGB can have different HSV values. Each HSV image frame is then temporarily stored for processing into binary model. This conversion is shown as follows:

The converted HSV image contains noise, which should be filtered out to produce an optimal threshold binary

image. This is done by adjusting the HSV values within a given range. The image processed here is of 24 bits (8 bits for each of RGB components), resulting in the range of 0 to 256 for each of HSV components. The filtered HSV image frames result in binary image frames, which are also stored temporarily.

5. Image Filtering

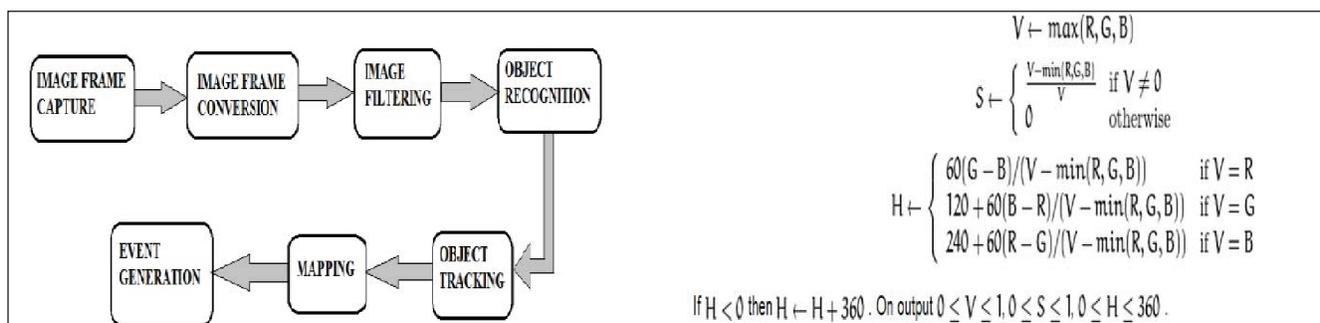
Image filtering technique is used to eliminate the noise involved in the image frame conversion from HSV to binary. Image Filtering involves performing morphological operations on the binary image to eliminate the noise and emphasize the filtered objects. Binary images may contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image. Prime morphological operations include Erosion and Dilation.

Erosion results in sharpening of the required part of the image, thus expecting it to be clearer for further processing. It is followed by the Dilation process, which enhances the sharpened image by filling the minute gaps. The combined effect of erosion and dilation operations result in filtered binary (threshold) image optimal for recognition.

6. Object Recognition and Tracking

The morphological operations performed on the binary image result in an optimal threshold image. Our job now, is to select a part of the image by finding suitable contours (borders). It is possible to detect multiple objects of different colors in the image simultaneously with the help of object oriented approach. Suitable

Figure 1: System Architecture



constraints are assigned on the area of each object for efficient processing. The range of the area of the required object is predetermined for extracting the contours. The area value, out of the threshold range will be neglected and hence, only the color of the required object is detected even if the background constitutes of same color as that of the object. Each object will behave as an independent entity; but they can be related as required for a particular task.

Once the required number of objects are recognized, they must be tracked for further manipulation. Tracking refers to finding the position (X and Y coordinates) of an object on the screen. This is done through calculating moments for every recognized object. A Reference Moment is calculated, which implies to an average intensity of all the pixels of the recognized object in the image frame.

The X coordinate of the object is obtained by calculating the ratio of row moment (average intensity of pixels in a particular row) of the object to its reference moment. In the same way, the Y coordinate is obtained by calculating the ratio of column moment of the object (average intensity of pixels in a particular column) to its reference moment.

This process outputs the position of the objects with respect to the dimension of the image frame.

7. Mapping and Event Generation

Since the area covered by the cursor is confined to the whole computer screen, the image frame coordinates should be mapped to screen coordinates. This is achieved by obtaining the ratio between the dimensions of image and screen coordinates and multiplying the constant obtained, to the image coordinates.

A desired mouse event is generated at a particular mapped position of the object. Basic mouse events include: Cursor movement, Left click, Double Click, Right click, which are achieved by detecting various contours of different colors. In addition to the basic mouse operations few advanced operations like play/pause, stop can also be implemented.

The combination of different contours with different colors can be used to generate various mouse events.

7.1. Mouse Operation

7.1.1 Cursor Movement

This mouse operation is performed by moving the forefinger throughout the screen area, where a specific color associated with the finger (contour) is detected and its position with respect to the screen coordinates is tracked continuously.

7.1.2 Left Click

The Left Click operation at a particular position is implemented by displaying the thumb along with the forefinger. In this case, the contour associated with both the colors are tracked simultaneously. As soon as the number of contours change to two with different colors, this event is generated.

7.1.3 Double Click

This operation is associated with the combination of different colors of forefinger, thumb and middle finger. Here, the number of contours changes to three.

7.1.4 Right Click

The Right Click mouse event at a particular position is generated when the combination of forefinger and middle finger with different colors are displayed.

8. Conclusion

This paper emphasizes the implementation of mouse in software as an alternative to the hardware mouse through the principles of Digital Image Processing. RGB image frames are captured continuously through the webcam and are converted to HSV and binary image frames sequentially, for efficient processing. The binary image is filtered through repetitive erosion and dilation processes, resulting in an optimal threshold image with minimal noise. Different objects as desired, are recognized from the binary image frame through suitable contours. The recognized objects are tracked by calculating moments associated with them, resulting in an objects position with respect to the dimension of the image frame. This process is followed by mapping of object's position from image coordinates to screen coordinates. Desired mouse event

generation for the tracked object at a particular position, marks the end of the whole process.

8.1 Future Enhancements

The basic mouse events generated here include: Cursor movement, Left click, Double Click and Right click. There is a room for future enhancements for various advanced events like: Media play/pause and stop, Volume increase/decrease, Track navigation and Screen refresh.

References

1. <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm>
2. <http://docs.opencv.org>
3. http://en.wikipedia.org/wiki/HSL_and_HSV
4. http://en.wikipedia.org/wiki/Image_processing
5. Jayaram, S., Esakkiranjana, S. & Veerakumar, T. (2011). *Digital Image Processing*. Tata McGraw-Hill Education.
6. Hounslow, K. Journal of Electrical Engineer.