IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. II (Mar-Apr. 2016), PP 77-80
www.iosrjournals.org

Virtual Mouse: Computer Vision Aided Pointing Device

Anupama.H.S[1], Vishwas S[2,] N.K.Cauvery[3], Lingaraju.G.M[4]
[1,2]*Department of Computer Sc. and Engineering, R. V. College of Engg., Bangalore, India*
[3]*Department of Information Sc. and Engineering, R. V. College of Engg., Bangalore, India*
[4] *Department of Information Sc., M. S. Ramaiah Institute of Tech., Bangalore, India*

Abstract: *This paper puts forth the idea for an easy to use device to replace the conventional mouse. One such approach is a virtual pointing device. A web-camera based solution takes advantage of computer vision technologies to design a software based solution to this problem. Virtual Mouse uses image processing techniques to perform color tracking which is used as the basis for the pointing action. This method reduces cost of hardware and also is very useful for controlling various applications which require a freer motion of the pointer compared to a mouse or a touchpad. This is done through tracking of user-defined, specific colored objects.*

Keywords: *Virtual Mouse, Computer Vision, web camera, color tracking, OpenCV*

I. Introduction

A mouse is the most commonly used and most familiar input device to a PC user. However, a mouse provides very little freedom especially for free-hand motion. Today, most devices in the market use either a touch-screen interface or a touchpad built into the keyboard in case of laptops, notebooks and the likes. Though touch-screens are a very convenient input device, this technology is not very affordable at the scale of a desktop computer. Also, touchpads, while a convenient addition to laptops, are not the easiest pointing device to use, especially for beginners. A viable alternative to the touchscreen is a virtual Human Computer Interaction device. This system uses a web camera to track objects of a particular color. The color is detected from the image and the image is then scaled. The pixel position is then mapped into the mouse input after being suitably scaled.

A. Image Processing and Analysis

Digital Image processing can be considered as a form of processing and filtering a 2 dimensional set of discrete values, which makes it akin to Signal processing, which could be considered as a one-dimensional image. An image can be discretized into a number of pixels, each of which have RGB components. The intensity of each pixel is stored in a 2-Dimensional Matrix.

The analysis of an image is the process of making sense of an image: a processed image has to be analyzed to extract useful, meaningful information out of it. In this case, once we receive an image from the web camera, we perform a number of image analysis techniques on it. The first of these is Color Detection. We traverse through the image, and extract the RGB value from each pixel. This enables us to use different colored objects to indicate various levels of pointing action, and different colors to perform various other functions of a mouse. This approach reduces the overhead of recognizing gestures, and increases efficiency without losing much functionality.

II. Existing Systems

A. Pointing Devices

We are familiar with the number of pointing devices that we use with our desktop systems to act as input devices. The most common ones being the trackballs, and the different kinds of mouse.

1) Trackballs:

Also called Upside-Down Mouse, a trackball is a precision pointing device. It employs sensors to detect a rotation of the ball about two axes. The rotation of the ball makes a pointer move. It is advantageous in its high precision, but however it is not very user friendly nor very accommodative to free motion.

2) Mechanical Mouse:

A device that was commonly used as a pointer, the mechanical mouse uses a ball which rotates when the mouse is moved. Sensors detect the motion and translate it to motion of the pointer in the same direction. It is a familiar tool, but has space constraints and low precision.

3) Optical Mouse:

An optical mouse is a variant of a mechanical mouse. It uses a LED source and a number of photodiodes which detect the motion of the mouse relative to a surface. Modern mouse works on most opaque surfaces, though it does not function on glass or transparent surfaces. They come in wireless variants, and may also be custom made for specific purposes, such as a gaming mouse.

4) Touchpads:

Touchpads are the most common replacement to the mouse in a laptop/notebook. To a user, it works in a very similar fashion to the mouse. A touch pad employs a tactile sensor to detect the movement of the finger across the surface and hence translate the same into motion of a pointer on the screen. It is advantageous for devices that lack space, but it is not as user friendly as a mouse.

5) Touchscreen Systems

Touchscreens are now a common sight in most appliances. They can be capacitive or resistive and overlay the display device. The touchscreen is a space saving solution to the problem of a pointing device. Its ease of use and user-friendliness makes it a very suitable implement for input. However, at the scale of a larger system, such as a desktop computer, a touchscreen would be costly and not always feasible.

B. Other Virtual Implementations

This paper is inspired by the works of K S Chidanand Kumar [2] and Hojoon Park [3] who use web cameras and machine vision to control a mouse pointer using gestures. We propose to instead use only color tracking and not hand gestures, to reduce computational overhead. A similar endeavor has been undertaken by Abhik Banerjee et al [4], who used MATLAB to use a color based tracking system. In the aforementioned system, multiple colors were used to differentiate between various functionalities. In this paper, an open source implementation of a color based tracking system is discussed using OpenCV. This ensures that the computation and image processing and analysis algorithms do not take up a lot of the available processing resources and provides a smoother system. This system is entirely implemented in OpenCV with C++, and does not add the overhead of a java program running in the background. It also uses an open source platform, which encourages modifications and enhancement of the system.

III. Proposed System

Virtual Mouse is a software based implementation of a pointing device which works with the help of a web camera. The camera takes images continuously, and the movement of the user's had is mapped to the mouse input. This recognition of the hand movement is done by holding an object of a particular color in the user's hand, in such a way that the object is within the field of vision of the web camera. The color from the image is used to track a pointer on the screen. The image needs to be scaled, as the pixel position in the image need not correspond with the desired position on screen, as they are of different resolutions.

A. Web-Camera Control

A web camera is used to provide a continuous stream of images. This implementation takes advantage of the fact that nearly all laptop systems and most computer systems generally have a web camera as an accessory. In the occasional event that this is not the case, a web camera is not pricey, and definitely costs less than other alternatives such as touchscreens. Thus the virtual mouse is nearly a completely software based solution

1) *Advantages*

- Software based solution: no extra expenses incurred
- Greater freedom of motion compared to a mouse.
- Greater degree of control, and use in interactive applications, such as games, art, etc.
- No dangers of health issues such as wrist pain, carpal tunnel syndrome, etc.

2) *Drawbacks*

- Requires a web camera
- Adds a small overhead of computation
- Not as accurate or precise as a touchscreen
- All actions must be within field of vision of the camera

IV. Implementation

A. Platform

The platform chosen for this implementation is OpenCV. The OpenCV project was an Intel-led research initiative, launched in 1999, for the purpose of advancing CPU intensive applications. It aimed to make optimized, reliable vision infrastructure available as open source so that one need not work from scratch to do

basic vision-related tasks. OpenCV is primarily C++ based and has an extensive C interface. It also has python, java, and MATLAB/OCTAVE interfaces and is supported on Windows, Linux, Android and iOS, and is an ideal platform for this endeavor.

B. Flow Diagram

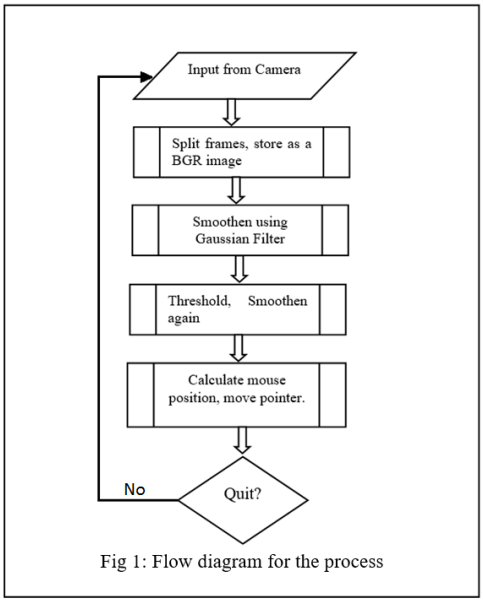

Fig 1: Flow diagram for the process

The image is captured using a web camera, and the only colors that are required are blue and red. The captured image is stored as frames. From these frames, the pixels are extracted by traversing through the image and the RGB values are extracted from them. The frames are then identified by the OpenCV library, which stores them as a matrix containing the intensity of the pixels as the values. The color defined for motion triggers the mouse movement when identified, and the color defined for clicking the mouse on identification, causes a mouse click function to be called.

C. Color Detection

Color detection is arguably an integral component of the system. The images are first captured using the web camera, and stored as frames, as mentioned before. Another blank image of the same size is then created and filled with black.

Each frame stored is iterated through, and the original image is smoothed using a Gaussian kernel. This acts as a low pass filter, removing high frequency components. This reduces noise in the image and makes it suitable for scaling, as well as more susceptible for edge detection.

Fig.2: Image captured and smoothened from the web-camera

The smoothened image is then converted to HSV (Hue, Saturation and Value) and threshold-ed to form a binary image. To do so, the image-array (matrix) is checked element by element and only the elements which lie between two predefined scalar values (which define range of values which form a particular color in HSV format) are retained. Hence, the threshold-ed image contains only the colors of interest to us. The binary image is once again smoothened using a Gaussian filter, and the required position is obtained. For better accuracy, a minimum size can also be defined so that false positives are minimized.

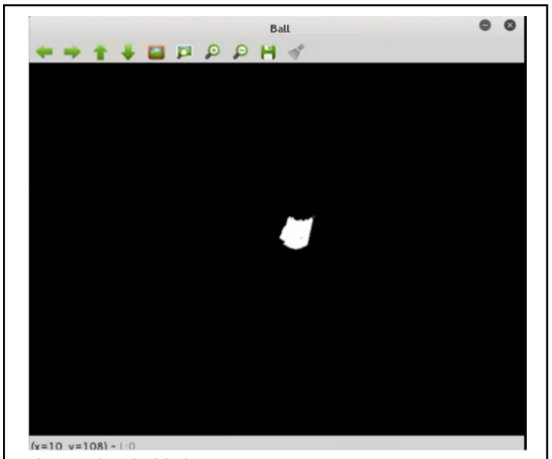

Fig. 3: Thresholded Image

The threshold-ed image is then tracked and mapped to the correct pointer movement. There is also a need for lateral inversion of the image. Also, a minimum area requirement is maintained to ensure there are no false positives due to noise.

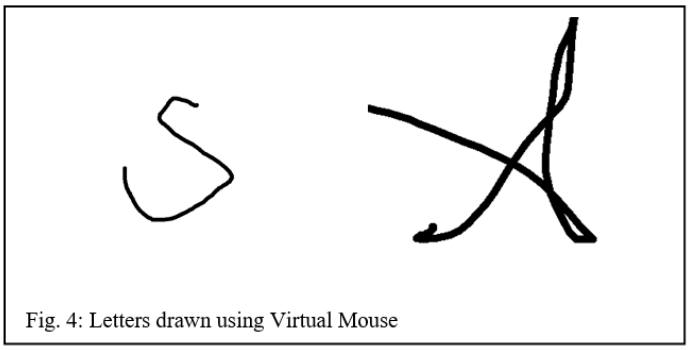
Fig. 4: Letters drawn using Virtual Mouse

Fig. 2 shows the final image captured by the camera, fig. 3 shows the threshold-ed image used to find the position of the pointer. Fig. 4 shows two letters drawn using the software.
As such, we see that we are able to use the virtual pointer for normal use.

V. Conclusion

A viable system has been realized on an open source platform to facilitate a computer vision aided, adaptable, multi-functional pointing tool which overcomes physical barriers. This system is realistic and provides an easy interface to interact with a computer, with no prior training required. However, it is not the end of the road just yet. The system can and will be further adapted for more functionality, such as double clicks, right-clicks, etc. A possible solution for a multi-functional system is the use of different colored stickers on different fingers. As with any other system, it can be further improved upon and optimized. However, we can conclude that a computer vision based virtual pointer is very much capable of acting as a "virtual mouse".

References

[1]. Gary Bradski, Adrian Kaehler, *Learning OpenCV*, O'Reilly Media, 2008.
[2]. K S Chidanand Kumar, *A Vision based Application for Virtual Mouse Interface Using Finger-Tip*, NIT-K, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3), 2012, 4460 - 4464
[3]. Hojoon Park, *A Method for Controlling Mouse Movement using a Real-Time Camera*, Brown University, Providence, RI, USA, Department of computer science, 2008
[4]. Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj, Hemanta Saikia, *Mouse Control using a Web Camera based on Colour Detection*, International Journal of Computer Trends and Technology (IJCTT) –volume 9 number 1 – Mar 2014
[5]. A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, *Computer vision based mouse*, A. E. Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASS). IEEE International Conference
[6]. Vision based Men-Machine Interaction: http://www.ceng.metu.edu.tr/~vbi/