VIRTUAL MOUSE

TEAM

1.RAGINI LALWANI

2.RAJNEESH KATKAM

3.SHIVAM THAKUR

4.SIDDHI SHARMA

5.GAURAV KULKARNI

ABSTRACT

Since the computer technology continues to grow up , the importance of human computer interaction keeps on enormously increasing. Nowadays, most of the mobile devices are using a touch screen technology. However , this technology is still not cheap enough to be used in desktop systems. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques can be an alternative way for the touch screen. In this study finger tracking based on virtual mouse application has been designed and implemented using a regular webcam. The motive was to create an object tracking application to interact with the computer and develop a virtual human computer interaction device.

A virtual mouse is a software tha t allows users to give mouse inputs to a system without usimg an actual mouse. To the extreme it can also be called as hardware because it uses a ordinary webcam . Virtual mouse whuch uses webcamera works with the help of different image processing techniques.

INTRODUCTION

The basic principle involved in making the virtual mouse is image processing and object tracking.

Image Processing is any form of signal processing for which he input is an image such as photographs or frames of video. The output of image processing can be either an image or a set of characteristics or parameters related to

the image. Most image processing techniques involve treating the image as a two dimensional signal and applying standard signal processing techniques to it. Image Processing usually refers to digital image processing. Here image processing is in the sense splitting each pixel of the image into RGB components.

Another important idea used is object/video tracking. Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are: human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing. Video tracking can be a time consuming process due to the amount of data that is contained in video. Adding further to the complexity is the possible need to use object recognition techniques for tracking and other purposes.

TECHNICAL BACKGROUND:

Since this is a software oriented project, we didn’t have much mechanical work to do. We wrote our code in Python language and along with it, we used OpenCV.

OpenCV(Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Our main reason for using this library was that it has more than 2500 optimized algorithms, which can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements,etc. Its biggest advantage was that it leans mostly towards real-time vision applications, which is the main agenda of our project.

We have also used 2 4-pin RGB LEDs for showing red, blue and green colours, in order to perform the different mouse functions.

PROPOSED SOLUTIONS:

When we first decided on the project topic, we had planned to use an Arduino board and IR sensors. But after discussing with our mentors and chalking out the drawbacks of our idea, we switched to image processing, which was free from errors mentioned above. We were advised to do our coding in python language, using opencv library.

Our project uses image processing along with colour tracking. We have assigned the different mouse functions to 3 colours- red, blue and green. Blue is for cursor movement, as well as scrolling, while red and green are for right and left clicks respectively.

IMPLEMENTATION AND STEPS:

The first step is to capture a video with the help of a webcam and divide the video into a number of images called frames. Thus, we will be working with each frame, one at a time.

Next, we ran a while loop, where each frame was converted from RGB to HSV colourspace. This was done because most image processing techniques use HSV colourspace for performing functions. Moreover, it is easier to draw contours in an HSV image.(which we’ll see later)

Next, we assigned thresholding ranges to red, blue and green and performed thresholding using cv2.inRange() function.

Since the resulting image was full of disturbances and noise, we used masking, blurring, and erosion, in order to ensure that minimum noise crept in.

The next major step was to find and draw boundaries(contours) around the resulting image. Since only the largest image(blob) was to be detected and tracked, we found out the largest contour in the image and rejected the rest.

In order to track the blob, centroid of the largest blob(found out in previous step) was calculated with the help of cv2.moments() function.

After finding the centroid, our major concern was to connect the colours to the different cursor movements. For this, we used an opencv mouse library called win32api. With its help, we connected the colours with the mouse functionalities.

In order to flash red, blue and green colours for cursor movements, we used 2 4-pin RGB LEDs.

CONCLUSION:

The mechanism was working quite perfectly and the LED lights responded perfectly even in bright environment. We were able to implement various mouse functions like left click, right click and scrolling.

FUTURE IMPROVEMENTS:

At present , our device is not very user friendly and it needs to be manipulated in different system environments. The next phase of t his project would involve motion detection of the fingers directly rather than using color detection.

Precautions:

1. For proper functioning of mouse, use optimum lightning conditions.
2. Avoid working in environment having red , green ,blue colour.