Computer Vision Based Mouse using Image Processing
Rohan. N. S.¹, K. Prashanth², Pavan Kumar. M³, Mohan Kumar. S⁴, Mrs. Anuradha Budihal⁵
Assistant Professor⁵
Department of Electronics and Communication Engineering
Don Bosco Institute of Technology, Bengaluru, India

Abstract: We humans, have a robust sensory system, which helps us to spot people and objects, play sports, perform operations, drive vehicles, read, and so on. Computer-Vision, within the simplest terms, Is that the automation of such a visible system that computers or machines, in general, can obtain a high-level understanding of the environment. In this project, we have developed a computer vision-based mouse which will regulate the mouse cursor using the image processing and object tracking algorithms. We will perform the mouse functionaries by showing colors in front of the webcam. In this project we will learn about: Image processing algorithms, Image thresholding, Canny edge detection, and object tracking. This system is implemented in Python programming language using the Computer Vision-based library OpenCV.

Keywords: Computer Vision, Image Processing, Image Thresholding, Canny Edge Detection, Object Tracking, Open CV, Python Programming.

I. INTRODUCTION

Processing speeds have risen within recent times making computers help humans in various tasks. It took the globe by storm 40 years ago, but now, after four decades, the demise of the evolutionary mouse is incredibly near. The next three to 5 years will gradually experience the downfall of the mouse and therefore the powerful but old device is going to be confiscated by gestural computer mechanisms like touch screens and face recognition devices. The mouse works fine within the desktop environment but not for home entertainment when engaged in anote book. One of the first problems that arise with cursors is jumping. Jumping is traced to mechanical problems within the mouse or a software glitch. And wheeled collects dust inside the wheel trap, causing improper functioning of the mouse. Additionally, even a wireless mouse requires the PC user to be relatively near the PC to register mouse movements. There is not much to move the pointer around the screen, since gestures and efficient key combinations eliminate much of the requirement for clicking on software window controls. In our proposed system, we’ve controlled mouse cursor movement and click on events employing a simple webcam using color detection technique. The user wears colored tapes to supply information to the system. A webcam is used for capturing live video and then processes it. No additional hardware is required by the system apart from the quality webcam which is provided in every notebook computer.

II. ALGORITHMS

A. Image Thresholding

* Colored image to binary image conversion:

Image Thresholding could be a technique to convert a colored image into a binary image, supporting some criteria. It is primarily employed in image segmentation - to segment a picture into different areas involving specific criteria and has only those areas visible within the image for further analysis.

● Types of Thresholding
B. Image Smoothing

- Remove Noise from an Image
- Reduce pixelation
- Replace pixel with an 'average' from its neighborhood
- Kernel – Smaller Matrix used to apply the smoothing algorithm on the image

Types of Smoothing:

- Averaging → Replace central pixel with average
- Gaussian Blur → Replace central pixel with weighted average with Gaussian weights
- Median Blur → Replace central pixel with Median
- Bilateral Filtering → 2D Gaussian Filter (Space & Intensity)

C. Canny Edge Detection
1. Noise Reduction
The first step consists of removal of noise in the image using a 5x5 Gaussian filter because edge detection is easily affected by noise.

2. Intensity Gradient
Finding Intensity Gradient of the Image. The smoothened image is then filtered with another technique, to obtain the first derivative of the pixel intensity within the horizontal direction and also the vertical direction. Gradient direction is often perpendicular to edges.

3. Find Local Maxima
A full scan of the image is done to remove unwanted edges once gradient magnitude and direction are obtained. Therefore each pixel is checked for the local maximum in their neighborhood.

4. Hysteresis Thresholding
The hysteresis thresholding stage is used to detect edges. Here two threshold values are used, min-Val and maxVal. Any edges with intensity gradient grater than max-Val are edges and those lesser than min-Val are non-edges and hence discarded. Edges between these two thresholds are either edges or non-edges. Otherwise, they are also discarded. Finally, we get powerful edges in the image.

III. PROPOSED METHOD

A. Block Diagram

1. Image acquisition
Any computer vision application starts with Image acquisition. Image acquisition is the digital representation of the visual characteristics of the physical world. Image sensors are accustomed to detect and capture the information required to create an image. Digital Cameras, Medical Imaging equipment, thermal imaging devices, RADAR, LIDAR, etc are some examples of image sensors. The images acquired are then processed in the next stage.

2. Image Processing
In this step, the signals of the acquired images are filtered to eliminate the noise or any irrelevant frequencies. If needed the pictures are padded and transformed into a distinct space, the processed images are then analyzed to extract useful information.

3. Image Analysis
Pattern identification, color recognition, object recognition, and feature extraction are the steps involved in analyzing the image.

4. Decision Making
This step houses meaningful numerical information, which ends up in making decisions.

B. Flow Chart

1. Image Acquisition:
The first step is to track an object, from a live video being captured by your webcam
   - Capture Video: So, we are going to first capture video from the webcam. As we had learned that a video can be a collection of frames.
   - Read Frame from Video: We will then read the frames from the video, to apply the image processing techniques, on the individual frames. Now that we’ve obtained the image, we’ll prepare the image appropriately, for the further process.

2. Image Preparation
   - Gaussian Smoothing: We will remove any noise from the image, using Gaussian Smoothing. The default images in Open CV, are within the BGR color space. But it is easier and better to spot colored objects within the HSV space.

3. Color Recognition
   - Color Space BGR TO HSV: But it is easier and better to identify colored objects in the HSV space. Finally, we will separate the objects of a specific color, or a color range, from the image.
   - Thresholding Based on Marker Color: We will do a thresholding operation to spot the objects of the precise color range. Once all the objects of the particular color are separated using color recognition, we are going to identify the article of our interest.

4. Object Identification
   - Contours for Colored Objects: We will first find the contours for all the colored objects.
   - Contour with Max Area: And, then identify our object because the object with the maximum contour area.
   - Features of Contour with Max Area: Finally, we are going to use contour features to trace the item within the live video.

IV. RESULT

A. The pipeline for CV based mouse
Image Acquisition: image acquisition, which involves acquiring the photos, that may have the markers on your palm, which requires to be tracked.
Image Preparation: The following step is to organize the raw images.
Color Recognition: Next step is to recognize the colored objects within the image.
Marker Identification: Now we are going to identify the markers, from all the objects recognized within the image.
**Marker Motion Tracking:** Once you identify the markers, in each frame, we will track their motion in different frames.

**Cursor Control:** Finally, we are going to translate this marker motion, to the control of the mouse cursor on your computer.

Note that, the primary 4 stages of the CV based Mouse, are equivalent to the Coloured Object Tracking.

Once the marker is identified in Marker Identification Stage, we will track the motion of the marker in numerous frames, by locating the centroid of the marker, within the Region of Interest. Finally, we should move the cursor, based on the marker motion. We will also click the mouse, depending on the marker color, as mentioned earlier.

**Figure.2. Grid Generation on the ROI**

**Figure.3. Find the location of the marker relative to grid**

B. Functions in Object Tracking

1. **IMAGE ACQUISITION**

i. **CAPTURE VIDEO**
To capture a video, we would like to make a Video Capture object. The argument is the device index or the name of a video file. The device index is the number that specifies which camera is connected to be used for video capture. Normally one camera will be connected, hence 0 is passed. The second camera can be selected by passing 1, and so on. Later afterward, we can capture the video frame-by-frame.

**ii. READ FRAME FROMVIDEO**
We will then use the read function, of the video capture object, to read the image from the video being captured. The function gives two outputs. It returns a Boolean, ret, which is true when the frame is read correctly. It also gives out the frame from the video captured, as an output. The captured image is stored within the variable, frame. We can now perform the usual image functions on the variable, called the frame.

2. **IMAGE PREPARATION**

i. **GAUSSIAN SMOOTHING**
We will use the function CV dot Gaussian blur, to use the Gaussian filter on the image. The primary argument is that the image. The second argument is that the kernel size of the Gaussian filter. The third argument is that the Gaussian blur standard deviation value, which will be set to 0 by default, to possess a central Gaussian filter. We can see the effect of the Gaussian blur on a picture. Gaussian blur, giving a more robust smoothened image.

**3. COLOUR RECOGNITION**
i. COLOR SPACE BGR TO HSV
For color conversion, you may use the cv.cvtColor function. The primary argument for the cvt Color function is the image file that has to be converted. The second argument is that the code for the conversion type. So, we are going to use the code, cv.color_BGR2HSV.

4. THRESHOLDING BASED ON MARKER COLOR
The in-range function takes three inputs: the original array, the lower bound, and, the upper bound. The function gives an array with the identical shape just as the original array the elements, corresponding to the elements of the first array, lying within the bounds, are given the value of 1. And, the elements, corresponding to the elements of the original array, beyond the bounds, are assigned the value of 0. So, as we have seen, the in-range function, applied for the green color, picks only the objects in green color and removes the remainder of the objects.

5. OBJECT IDENTIFICATION

i. CONTOURS FOR COLORED OBJECT
Now that first, we converted the original image into a binary image, we are going to find the contours of the colored objects, from the binary image. To find the contours, we will have to use the cv. find contours function. There are three arguments in cv. find Contours function. The primary one is that the contour retrieval mode, which you had learned about earlier. And the third is that the contour approximation method. The find contours function, outputs a modified image, the contours, and therefore the hierarchy. Contours contain a Python list of all the contours within the image. Finding contours, that is the array of x, y coordinates of points lying in the circumference of the object using NumPy will result in a modified image. Hierarchy is the representation of the relationships between different contours, found from an image.

The contours identified, is displayed on an image, using the draw Contours function, as shown.

ii. CONTOUR WITH MAXAREA
Finally, we will find the contour with the maximum area, as the contour of interest, and identify it as the object.

iii. IGNORE THE SMALLER CONTOURS AS ERRORS
Smaller contours must be ignored because the smaller contours are because of error in some settings or they might be some pixels that do not belong to the object.

6. FEATURES OF CONTOUR WITH MAXAREA

i. CALCULATE AND USE FEATURES OF CONTOUR WITH MAXAREA
a. CENTROID
b. BOUNDINGRECTANGLE
C. Computer Vision-Based Mouse

1. Cursor control with no physical device
A computer vision-based mouse is a system to manage the cursor of your computer, with no physical device, even without a mouse.

i. Track colored markers on the palm
We will essentially have colored markers on your palm. We will capture the video of the motion of your palm, using the webcam of your computer.

ii. Translate marker motion to cursor movements
We will track the colored markers, and use their motion, to manage the cursor of your computer. We use three different colors for the three typical actions of the mouse.

2. Color 1 – Cursor Movement
Color 1 for cursor movement

A computer vision-based mouse is a system to manage the cursor of your computer, with no physical device, even without a mouse.

i. Track colored markers on the palm
We will essentially have colored markers on your palm. We will capture the video of the motion of your palm, using the webcam of your computer.
ii. Translate marker motion to cursor movements
We will track the colored markers, and use their motion, to manage the cursor of your computer. We use three different colors for the three typical actions of the mouse.

2. Color 1 – Cursor Movement
Color 1 for cursor movement

We will implement the cursor movement, left-click, and right-click during this project.

a. Use PyAutoGUI to maneuver the cursor based on marker location in the grid. You will use the pi auto G UI library of python, to control the mouse cursor.

b. Click, if in case Color 2 or 3 are detected within the screen and to manage the clicking also, based on the color of the marker.

PyAutoGUI is a Python module for programmatically controlling the mouse and keyboard of your computer.

I. The mouse functions of PyAutoGUI use x- and y-coordinates of the pixels on the screen, just like any image. So, the resolution of a screen, which determines what percentage of pixels wide and tall your screen is, which may be a significant parameter.

II. You will be able to control the mouse using the yellow marker, and you will just flash the green marker, whenever you would like to click. Ensure that there is no interference of any background object, with the marker colors.

3. Color 2 – Left Click
Color 2 for left click

Table 1.

<table>
<thead>
<tr>
<th>Finger tips color detected</th>
<th>Operations performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow</td>
<td>Move Cursor</td>
</tr>
<tr>
<td>Blue</td>
<td>Right Click</td>
</tr>
<tr>
<td>Green</td>
<td>Left Click</td>
</tr>
</tbody>
</table>

Table 2.

V. CONCLUSION
In this paper, we develop a computer vision-based mouse to control the cursor using the object tracking algorithm. In this project, we can perform all functions of the mouse by showing colors to the webcam. By building this project we learn about: image processing algorithms, image thresholding, canny edge detection, and object tracking.

VI. REFERENCES


[5]. S Song, D Yan, & Y Xie, (IEEE ICNSC 2018). Design of control system based on hand gesture recognition- based on USB singlecamera.