5.3%

PlagScan by Ouriginal Results of plagiarism analysis from 2021-02-24 16:18 UTC

Unnamed

Date: 2021-02-24 16:16 UTC

✱ All sources **2** | Internet sources **2**

✓	[0]	programmingdesignsystems.com/color/color-models-and-color-spaces/index.html 5.3% 7 matches
✓	[1]	www.ijirset.com/upload/2020/september/31_Indian_NC1.PDF 3.1% 5 matches

4 pages, 1611 words

PlagLevel: 5.3% selected / 5.3% overall

7 matches from 2 sources, of which 2 are online sources.

Settings

Data policy: *Compare with web sources, Check against my documents*

Sensitivity: *Medium*

Bibliography: *Consider text*

Citation detection: *Reduce PlagLevel*

Whitelist: --

1. Introduction

1.1 Motivation

The mouse is one of the important inventions in computer world which improved human machine interactivity. In present scenario, wireless and Bluetooth embedded mouse technology is developed but it is not that ideal to use as this technology is not completely hands free. In Bluetooth technology the mouse developed requires power such as battery power or connecting devices like dongle.
By adding these extra components to the mouse the cost increases and it is also not user friendly to use. So, in the proposed model these limitations are removed and interaction is made user friendly as well as cost is also reduced. This model is actually a virtual mouse system using hand gestures with the help of computer vision.

1.2 Definition

The primary aim of this project is to improve human and computer interaction by developing an effective and alternative way of controlling the cursor and its various functions such as left click, right click, scroll up, scroll down and selection.
With the help of this model a user can easily interact with the system/computer without the use of an external device. It also allows the user to interact with system from a considerable distance. And because of this improvement the essential hardware required for the computer/laptop is also reduced.
This model makes use of the webcam to capture the video feed, which is used for image processing, and now a days webcam is an essential device/hardware for a laptop/computer and therefore it is not considered as an extra hardware.

1.3 Objective

The objective of this project is to improve the usability and experience of the users by providing a better machine and human interaction. The Image/Video feed from the webcam is used to track and detect gestures and mapping those unique gestures to different mouse events.
Retrieve the image data from the webcam and convert it to a usable format and use it as input. The image must be filtered to identify different colors. Make contours and find the mean positions of those contours. Track those mean positions to recognize the gestures. Map those gestures to different mouse actions and use them.

2. LITERATURE SURVEY

2.2 Proposed System

The model/system that we proposed is a cursor control model using image processing, which takes hand gestures that are being captured from a webcam as an input in HSV format and by detecting the centers of the color contours thereby detecting the gestures, it performs mouse events and mouse movements. We use three standard colors which are represented on three fingers of a hand to detect the gestures.
Python, PyAutoGui and OpenCV (cv2) library is used to implement this model for Hand gesture tracking. This system is easy to use and it is also easy to remember the gestures that are provided to perform different operations. And once the Model is deployed on the system, we can use this version of the mouse all the time.

3. SYSTEM ANALYSIS

3.1 Functional Requirements Specifications

After analysing the project objective, we have come up with following modules:

• Color Calibration:
With the help of Computer vision (Opencv2), we recognised three standard colors and converted them into HSV format using cv2.convert() function and calibrated the colors. Users can dynamically adjust hue, saturation, value of the colors.

• Centroid Detection:
From the output of the color calibration module, we calculate a mean point which represents contour of each color. With these centroid points we detect the type of gesture position made by user.

• Multiple Gesture Recognition:
By the position and movement made by the user various gestures are recognised, if any gesture is not recognised by the system model then it considers it as free movement of the mouse cursor with help of only one color.

• Mouse Events and Movement:
Based on the gestures recognised by the model equivalent mouse events are performed with the help of ‘Pyautogui', a python library.

3.2 Software Requirements

• 64-bit Operating System: Windows 7 or Higher
• Jupyter Notebook/ Google Colab
• Python version 3.0
• PyAutoGUI
• OpenCV

3.3 Hardware Requirements

• A Good Webcam
• RAM preferably 8GB for faster processing
• GPU
• Memory space 500GB

3.4 Environmental Specifications

• A will lit room with mostly light and pale colors.
• No other objects should be present in front of the webcam (particularly red, yellow blue colored objects) except colors on the user's hand.

5. Implementation

5.1 Methodology

5.1.[0]1 Concept

HSV color space - a cylindrical color model that remaps the RGB primary colors into 3D (hue, saturation, and value).[0] Hue specifies the angle of the color on the RGB color circle.[0] A 0° hue results in red, 120° results in green, and 240° results in blue.[0] Saturation controls the amount of color used.[0] A color with 100% saturation will be the purest color possible, while 0% saturation yields grayscale.[0] Value controls the brightness of the color.[0] A color with 0% brightness is pure black while a color with 100% brightness has no black mixed into the color.

5.1.2 Implementation of Model

In this model we recorded the image from webcam and converted it from RGB to HSV color format. Then user required to calibrate the three colors i.e, red, yellow, blue in the image which consists of user hand with three colors on fingers individually. In this model we used calibrateColor() function to calibrate the three colors individually by the function. The user can manually calibrate the colors by adjusting the Hue, saturation, value ranges or they can choose default values.

From these calibrated individual images the three fingertips where the color band lies are extracted from user hand, in the video captured by webcam, using the cv2.inRange() function i.e. contours are detected. Noise is removed by applying morphism which contains two steps erosion and dilation. The output images are free from noise and are termed as ‘mask', and from these images the centroids are calculated.

As a webcam is not ideal, so the noise of the background is also captured in the image and also the vibrations of the user may also get captured. Due to these vibrations of the user hand the centroids keep varying about a mean position which reduces the efficiency of the model. In order to increase the efficiency and differentiate the actual movements from vibrations, we used an enhanced formula to handle the noise caused by vibrations.

The function chooseAction is used to calculate the three centroids, which is used to decide the functions to be performed by the mouse based on the positions of the three centroids. Based on the action returned by chooseAction() function, we use performAction() function to perform the equivalent mouse events and movements using PyAutoGUI library which can be implemented in python environment. The actions that need to be performed by the system are:
• free cursor movement
• scroll up
• scroll down
• left click
• right click
• drag/select

5.1.3 Noise rate formula
To deal with the noise caused by unwanted vibrations we use following method- first we compare the new position of the centroid with the previous position. If the difference between them is less than 5 pixels, we considered that it occurred due to noise.
If the difference between them is more then it is done by user voluntarily, so the cusor position is changed accordingly by this function.

• To set cursor position, if the cursor movement is voluntary then
  Position = centriod + (learning rate)*(previous_pos - centriod)
• if the cursor movement is because of vibration of hand or involuntary then
Position = centriod + (learning rate)*(previous_pos - centriod)

6. TESTING
6.1 Types of Testing
We have implemented Integration testing and Regression testing methodologies for testing the working of this project. In the Integration testing we tested the code

for each module and then integrated them based on the hierarchy to top and tested the final code to verify it passes all of the test cases. As we were doing Integration testing, we identified few bugs in final code and then we modified the code and retested it again and again until it satisfied all test cases.

7.1 Conclusion

Hand gesture tracking mouse control is developed using python language and OpenCv library. Using this model the system can perform all the mouse events and mouse movements without the need of the actual mouse. The system can control the Cursor by recognizing the hand of the user. And for every cursor function or mouse event a unique hand gesture is provided.

This developed model has the potential to replace the actual mouse but because of few constraints or drawbacks it cannot completely replace the mouse. If those drawbacks are tackled then it can completely replace the mouse. There are different tools for gesture recognition. We have used multiple functions from computer vision library and developed a model by tackling few limitations that are present in previous object tracking models.

7.2 Future Work

The problem with this model is that it works well only in a room with good lighting. If the lighting in the room is not good then its performance also decreases. Only because of this reason it cannot completely replace an actual mouse, as people generally use laptops in the outdoor environment as well where the lighting is not good. Therefore further improvements can be made to use it in the bad lighting environments as well. And improvements can also be made such that the three color bands are not required.

It can be further developed with AR technology for more human-machine interaction like in video games etc. More Gestures can be developed where each unique gesture can open certain applications.