

PLAGIARISM SCAN REPORT

Words 994 Date February 24,2021

Characters 6291 Excluded URL

9%

Plagiarism

91%

Unique

4

Plagiarized
Sentences

40

Unique Sentences

Content Checked For Plagiarism

5. Implementation

5.1 Methodology

5.1.1 Concept

HSV color space - a cylindrical color model that remaps the RGB primary colors into 3D (hue, saturation, and value). Hue specifies the angle of the color on the RGB color circle. A 0° hue results in red, 120° results in green, and 240° results in blue. Saturation controls the amount of color used. A color with 100% saturation will be the purest color possible, while 0% saturation yields grayscale. Value controls the brightness of the color. 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 cursor position is changed accordingly by this function.

- To set cursor position, if the cursor movement is voluntary then

$$\text{Position} = \text{centriod} + (\text{learning rate}) * (\text{previous_pos} - \text{centriod})$$

- if the cursor movement is because of vibration of hand or involuntary then

Position = centroid + (learning rate)*(previous_pos - centroid)

S.No. Mouse Event Pixel Difference Description Learning Rate Sensitivity

1 Cursor Movement < 5 Voluntary

Hand Movement 0.7 Ideal

2 Cursor Movement < 5 Voluntary

Hand Movement 0.8 Ideal

3 Cursor Movement < 5 Voluntary

Hand Movement 0.9 High Sensitivity

Hand Movement 0.6 Low Sensitivity

6 Cursor Movement >5 Vibrations/Noise 0.1 Ideal

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.

6.2 List of Test Cases

S.NO Description Cursor Status Test Case Result

1 Test Case with only blue and red colors No movement of cursor or Action PASS

2 Test Case with only yellow and red colors Only Cursor Movement

No Actions performed PASS

3 Test Case with only yellow and blue colors Only Cursor Movement

No Actions performed PASS

7. CONCLUSION AND FUTURE SCOPE

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.

Sources	Similarity
Color models and color spaces - Programming Design Systems https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html#:~:text=Hue specifies the angle of,while 0% saturation yields grayscale.	25%
Hsv color A 0 hue results in red 120 results in green and 240 results in blue. The code may be useful to someone out there please let me know if it is HSB color Consists of ... http://new2.8-800.su/binary-trigger/hsv-color.html	25%
Color models and color spaces - Programming Design Systems Saturationcontrols the amount of color used. A color with 100% saturation will be the purest color possible, while 0% saturation yields grayscale. Valuecontrols the brightness of the color. A color with 0% brightness is pure black while a color with 100% brightness has no ... https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html	25%