Intangible Canvas Drawer Using Image Processing

CH. Srigiri[1] | K. Jaya Surya[2] | K. G. V. S. Avinash[3] | B. Nageswara Rao[4] | A. Jayakar[5]

[1]Assistant Professor, Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology (A), Rajahmundry, Andhra Pradesh, India
[2,3,4,5] UG Scholar, Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology (A), Rajahmundry, Andhra Pradesh, India

Abstract: Intangible Canvas Drawer is a modern technology that allows you to draw your imagination by just waving your finger in the air. In this, we have presented an Intangible Canvas Drawer using Image Processing which can draw anything on it by just capturing the motion of a colored marker with a camera. Here a colored object at the fingertip is used as the marker. We will be using the module called OpenCV to build this project. The language using here is python due to its exhaustive libraries. Color Detection and frame tracking are used to get the result. If the color marker is detected and a mask is produced. It performs the steps of morphological operations on the masked image generated called Erosion and Dilation. Erosion reduces the impurities present in the mask and dilation further restores the eroded main mask. This Project is more appropriate for seminars, webinars, and online classes.

KEYWORDS: OpenCV, NumPy, Python, Masking, Morphological operations (erosion and dilation).

Check for updates

DOI of the Article: https://doi.org/10.46501/GIETEC22

Available online at: https://ijmtst.com/icetee2021.html

As per UGC guidelines an electronic bar code is provided to seure your paper

To Cite this Article: CH. Srigiri; K. Jaya Surya; K. G. V. S. Avinash; B. Nageswara Rao and A. Jayakar. Intangible Canvas Drawer Using Image Processing. *International Journal for Modern Trends in Science and Technology* 2021, *7*, pp. 123-126. https://doi.org/10.46501/GIETEC22

Article Info. Received: 18 May 2021; Accepted: 25 June 2021; Published: 30 June 2021

INTRODUCTION

Ever tried to write anything by just moving your hand in the air. In this, we are going to design an Intangible Canvas Drawer which can draw anything you want by just capturing the movement of a colored object with the camera. Here a colored object at the fingertip is used as the marker.

Here Color Detection and frame tracking are used to get the wanted result. The color object is detected and a masked image is generated. It performs the steps of morphological operations on the masked image generated called Erosion and Dilation. Erosion reduces the impurities present in the mask and dilation further restores the eroded main mask.

This project is developed to avoid and reduce the hardships faced by this previous system. Moreover, this system is designed to carry out the operations smoothly and effectively. This project is reduced as much as possible to avoid errors with the previous system. There is no need of any knowledge for users to use this software. Thus by this, it says it is user-friendly.

Everyone loves the touch screen. When using the touch screen gadget, we feel very comfortable with it. When the touch screen mobiles were invented and introduced, everyone felt the same. But slowly, the excitement of using a touch screen started fading. While using the phone with the fingertip the screen started getting lots of scratches and damages on the screen. When we use a screen guard; still scratches and marks over such a beautiful screen is a defect. The same happens with every mobile touch. Most of the time we have to clean the screen to get rid of damages and scratches on the screen.

OBJECTIVE

The objective of this project is to create a simple prototype for a drawing tool that uses hand gesture recognition to paint on a screen. Color Detection and frame tracking are used to get the wanted result. The colored object is detected and a masked image is generated. It includes the further steps of morphological operations which are Erosion and Dilation. Erosion reduces the impurities present in the mask and dilation further restores the eroded main mask. The points which will be captured will be stored in an array for remembering the locations where the pointer had arrived. This will help to draw the pattern of the movement of the pointer. Each pixel at the location of the pointer is colored with a specific color to show the drawing on the screen. This object detection is based on the OpenCV module. The object is to develop less touch, clear explanation, and less time. Frame results are generated for observing in the Jupyter Notebook. Each Block of the Design is written in Python code for each module. Run the code in Jupyter Notebook to see the output. We can able to see each pixel at the location of the pointer is colored with a specific color drawing on the screen.

LITERATURE SURVEY

Initially, we have used boards and markers for classes. Later, We have used Powerpoint presentations for a clear explanation. After that, we have used some software like Microsoft Whiteboard. There is a screen touch presentation software. That software can damage the screen and there will be no further proper teaching with this software. In our proposal work, we can get rid of all physical components required for drawing and makes use of a pointer tracking mechanism to convert pointer traces to drawing.

MOTIVATION

The initial motivation was a need for a dustless classroom for the students to study in. We know that there are many ways like touch screens and more but everyone cannot afford to buy the large screens like televisions and software for teaching. So, We thought why not can a finger be tracked, but that too at an initial level without deep learning. So we have used a module called OpenCV for these kinds of projects. Intangible Canvas Drawer is a digital drawing canvas that utilizes a Camera, and OpenCV to write on a screen. The direction of the pointer is controlled using the open-source OpenCV library and to map the pointer onto the screen.

BLOCK DIAGRAM

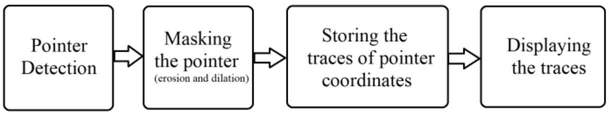

Pointer Detection
Masking the pointer (erosion and dilation)
Storing the traces of pointer coordinates
Displaying the traces

Figure -1: Block Diagram

The incoming image from the webcam is to be converted to the HSV color space for detecting the colored object at the fingertip**.** The Python code converts the incoming image to the HSV color space. So it is a very suitable and perfect color space for tracking the color**.** We will make the track bars to set the HSV values to the certain wanted range of color of the colored object that is placed at our fingertip. The various HUE values and other different color ranges can also be observed. When the track bars are created, we will get the value from the track bars and create the required range. These ranged values are sent into the function which is a numpy structure. The function which has received the ranged values generates a mask of the colored object.

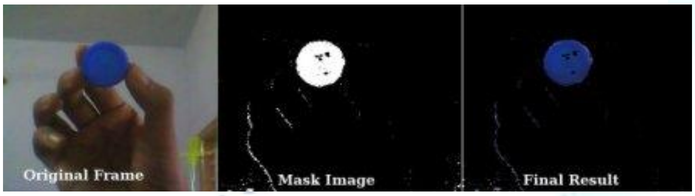


Figure -2: Masking an image

The generated mask shows white pixels of the detected color object in the same shape. After generating the masked image, it's time to find the center of the colored object. In the Code, We are performing some morphological operations on the Mask, to make it free of impurities and to detect contour easily.

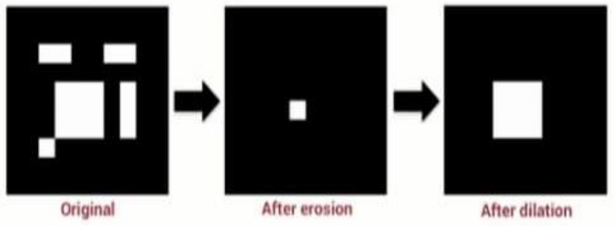


Figure -3: Morphological operations

The real process happening behind the process is it all stores the movement of the finger in a deque (A data structure) using Python. The deque stores every coordinate that the colored object is moving and print all stored coordinates. It is all done using the module called OpenCV.

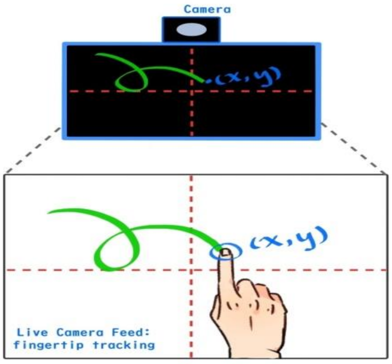
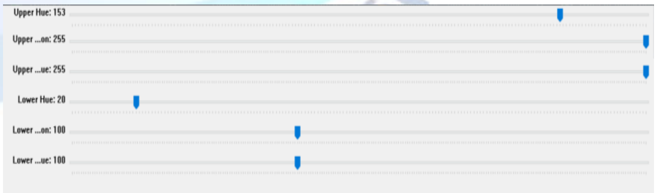


Figure -4: Fingertip Tracking

Firstly Make Four deques, for four distinct colors of the project. We will use the position of the contour to decide if we want to click on a button or we want to draw on the sheet. We have arranged some different colored options on the top of the screen, if the detected colored object comes into their area of those options, we will call their method. We have four different colored options on the canvas to draw using OpenCV. We write an else condition for avoiding unnecessary drawing when the contour is not present. Lastly, We draw all the stored points in deque at the particular positions with the respective selected color.

RESULTS AND DISCUSSION

The code which is written in Python is executed in Jupyter Notebook. The outputs are observed as shown below.



Upper Hue: 153
Upper ...on: 255
Upper ...ue: 255
Lower Hue: 20
Lower ...on: 100
Lower ...ue: 100

Figure -5: Color Detectors

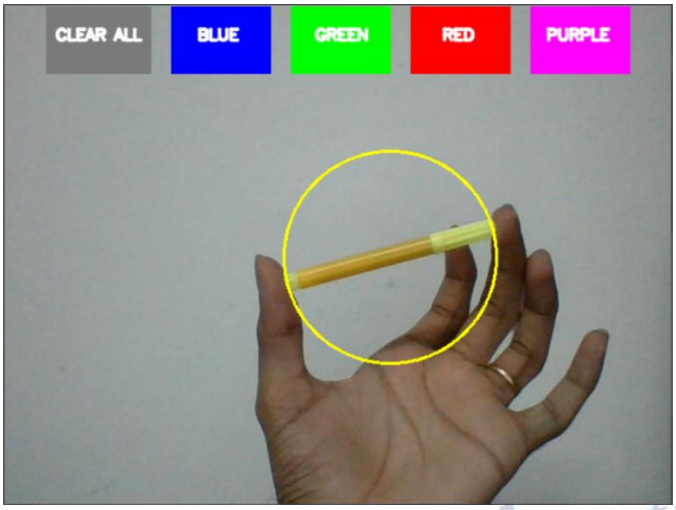


Figure -6: Frame tracking

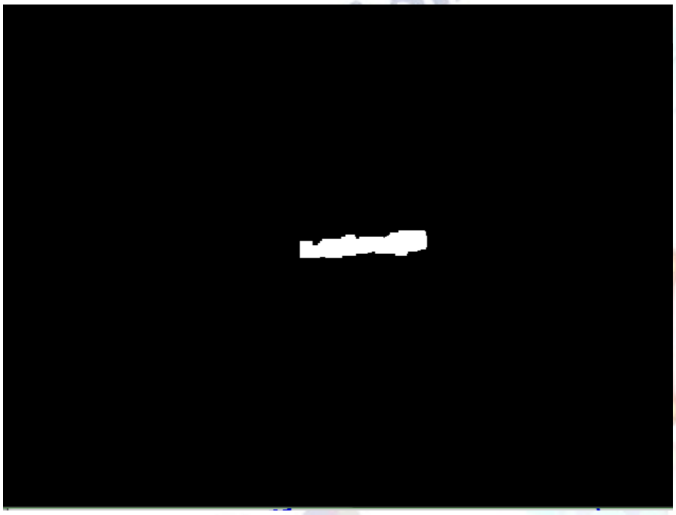


Figure -7: Masked image

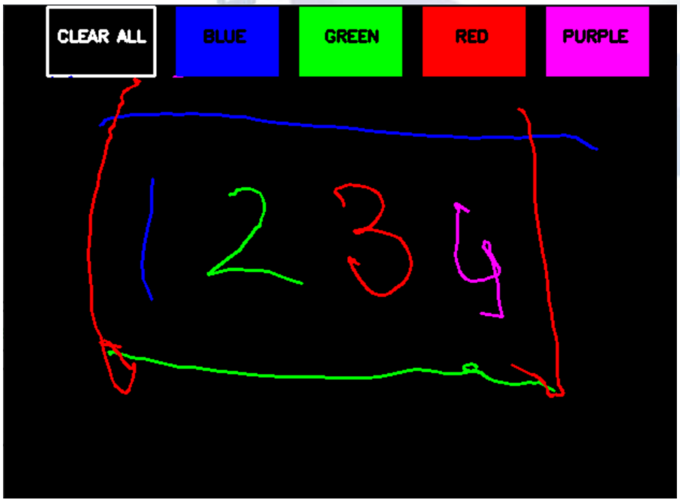


Figure -8: Paint Window

We get four tabs namely Color detectors, Masking, Frame tracking, and Paint Window. The color that has to be detected can be set in the Color detectors tab. The Webcam detects the colored object. The detected object is first eroded and then dilated. Masked image is obtained after performing those morphological operations. The tracking of the object can be seen in the frame's window. Following the movement tracking, what we write in the air can be reflected and visualized in the Paint Window.

References

[1] Jaya S. Kulchandani and Kruti J. Dangarwala, "Moving Object Detection," published in IEEE, 2015.

[2] Yizhong Yang, Qiang Zhang, Pengfei Wand, Xionglou Hu, and Nengju Wu, "Moving Object detection for Dynamic Background Scenes Based on Spatiotemporal Model," published in 2017.

[3] K. Sreedhar and B. Panlal, "Enhancement of Images Using Morphological Transformations," published in IJCSIT, 2012.

[4] Basilios Gatos, Stavros J. Perantonis, Nikos Papamarkos, and Ioannis Andreadis, "Fast Implementation Of Morphological Operations Using Binary Image Block Decomposition," published in International Journal of Image and graphics, 2014.

[5] Akshansh Sharma, Firoj Khan, Deepak Sharma, and Dr. Sunil Gupta, "Python: The Programming Language Of Future," published in IJIRT, 2020.

[6] G. Lindstrom, "Programming With Python," published in IEEE, 2005.

[7] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek, "A Bried Introduction to OpenCV," published in IEEE, 2012.

[8] Ahmad Puad Ismail, Farah Athirah Abd Aziz, Nazirah Mohamat Kasim, and Kamarulazhar Daud, "Hand Gesture Recognition On Python and OpenCV," published in ICEEPE, 2020.

[9] Arshi Prabhakar, Neeti and Rakhi Devi, "Different Color Detection in an RGB Image," published in International Journal of Development Research, 2017.

[10] H. Altun, R. Sinekli, U. Tekbas, Fuat Karakaya, and Murat Peker, "An Efficient Color Detection In RGB Space using hierarchical neural network structure," published in IEEE, 2011.

[11] T. S. Huang, W. F. Schreiber and O. J. Tretiak, "Image Processing," published in IEEE, 1971.

[12] Naveen Kumar Mahamkali and Vadivel Ayyasamy, "OpenCV for Computer Vision Applications," published in Researchgate, 2015.

[13] S. Chris Colbert and Gael Varoquaux, "The Numpy Array: A Structure for Efficient Numerical Computation," published in IEEE, 2011.