AN EFFICIENT TOOL FOR ONLINE TEACHING USING OPENCV

[1]Gangadhara Rao Kommu

[1]Assistant Professor
[1]Department Of Information technology,
[1] Chaitanya Bharati Institute of Technology, Hyderabad, India

ABSTRACT: the main goal of computer vision is to identify and recognize different objects of various size, shape and position. the major problems faced by the computer vision is the illumination and the viewpoint of the object, concerning this multiple studies on detecting and recognizing objects that showed a high level of accuracy and precision on these tasks. to facilitate object detection online the proposed work allows the user to track the movement of any cultured object of his/her choice. the user can even choose the colors of his choice to be displayed. by running the application, the camera is activated thus enabling the user to draw in the air just by waving the tracker object. the drawing is also simultaneously visible on the white window. instructor can choose any color of his choice displayed above to draw and also can clear the screen when needed. we will be using the computer vision techniques of opencv and python to build this application

Index Terms - OpenCV, Object Tracker, Computer Vision, Python

I.   INTRODUCTION

The idea for this tool is a result of interest in digital drawing and smart photo recognition software. The initial motivation came when there was a need for a dustless class room for the students to study in. We know that there are many ways like touch screens and more but what about the schools which can't afford it to buy such huge large screens and teach on them like a T.V. OpenCV in python to draw on the screen using a virtual pen i.e, any marker can be used to draw[1] using the technique of contour detection based on the mask of the desired colored target marker

II.   PROBLEM STATEMENT

To develop An AI based tool using techniques of OpenCV which can draw anything on any surface by just capturing the motion of a coloured marker with a camera. Here a coloured object at the tip of the finger is used as the marker.
.

III.   PROPOSED SYSTEM

This tool allows the user to track the movement of any coloured object[2] of his choice. The user can even choose the colors of his choice to be displayed. By running the application, the camera is activated thus enabling the user to draw in the air just by waving the tracker object. The drawing is also simultaneously visible on the white window. He/She can choose any color of his/her choice displayed to draw and also can clear the screen when needed. We will be using the computer vision techniques of OpenCV to build this application.

IV.   IMPLEMENTATION

Start reading the frames and convert the captured frames to HSV color space (Easy for color detection). Prepare the canvas frame and put the respective ink buttons on it. Adjust the track bar values for finding the mask of the colored[3] marker. Preprocess the mask with morphological operations (Eroding and dilation). Detect the contours, find the center coordinates of largest contour and keep storing them in the array for successive frames (Arrays for drawing points on canvas). Finally draw the points stored in an array on the frames and canvas. The following algorithm will better explain the process

A. *Algorithm step for building board Tool:*

1. Setting a trackbar function.
2. Creating the trackbars needed for adjusting the marker color.
3. These trackbars will be used for setting the upper and lower ranges of the HSV required for the specific color.
4. Initializing different arrays which will handle color points of diff colors.
5. These arrays will hold the points of the specific color.
6. They are then used to draw on the canvas.
7. Initializing indexes for all diff colours which will be used to mark position of pointers in the color arrays using a kernel for dilation purpose.
8. Creating an array "colors" which will be used for the drawing purpose.
9. Setting up a Canvas.
10. Loading the default webcam of PC.
11. Loop begin
12. Reading the frame from the camera.
13. Flipping the camera so that it is easy to see the same side of the user.
14. Getting the updated positions of the trackbar and setting the HSV values.
15. Adding the colour buttons to the love frame for colour access 1.e if the trackbar moves on to blue the tracker becomes blue. Similarly for green red yellow and to clear the screen.[5]
16. Identifying the pointer by making its mask.
17. Find contours for the pointer after identifying it.
    Loop end;
18. If (the pointers are formed) begin
19. Sort the contours and find the biggest one.
20. Get the radius of the enclosing circle around the found contour.
21. Draw a circle around the contour.
22. Calculating the centre of the detected contour.
        end
23. If (Now we will check if the user wants to click on any button above the screen.) begin
24. Clear button
25. end
26. Append the next deques when nothing is detected to avoid any mess.
27. Draw lines of all colors on the canvas and frame.
28. Show all the windows.(Tracking, paint, mask)
29. If (q is pressed): then we stop the application.
30. Release the camera and all other resources.

V. RESUTLS

Overall, we achieved goal of developing a tool for conveniently drawing or writing any kind of surface. The screenshots of the results are listed below.

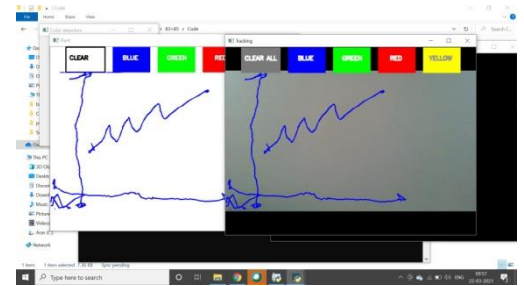Fig 5.1: Drawing text on surfaces with cam visibility

Fig 5.2:Free handwriting
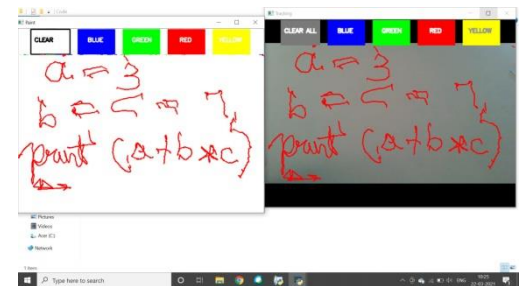

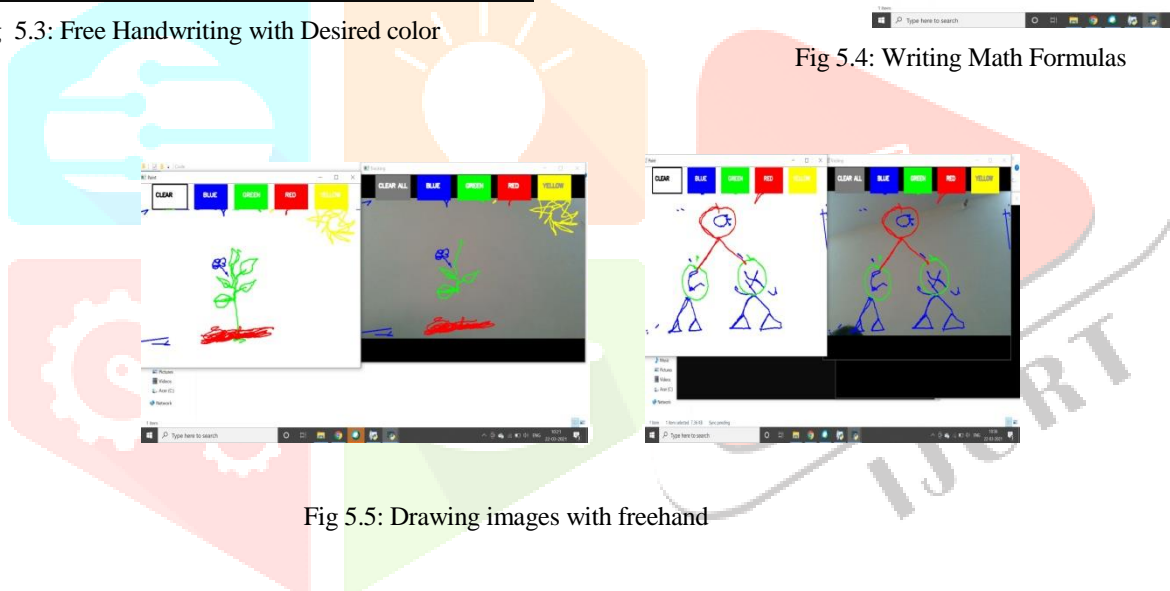Fig 5.3: Free Handwriting with Desired color

Fig 5.4: Writing Math Formulas

Fig 5.5: Drawing images with freehand

VI. CONCLUSIONS

we have developed a hands-free drawing program that uses OpenCV to detect the user's pointer finger. Colorful lines can be drawn wherever the user desires and the brush can even be modified. It is truly like drawing in the air. Our application allows the user to save their final work or watch their drawing process as a playback animation could also be unique features that resemble real creativity software

VII. FUTURE WORK

This work can further improved by including hand contour recognition, and multicore module. To enhance hand gesture tracking, we would have to delve more into OpenCV. Furthermore, we could experiment with different interpolation methods such as PyGame includes a line drawing method (pygame.draw.line ()) that could prove useful in producing smoother and cleaner lines. On the same vein, implementing a variety of brush shapes, textures, and even an eraser would make this tool more robust as a drawing application.

REFERENCES

[1] J.-F. Rivest, P. Soille, S. Beucher. Morphological gradients. Journal of Electronic Imaging, vol. 2 (4),pp. 326-336, 1993

[2] Comparison of OpenCV's Feature Detectors and Feature Matchers Frazer K. Noble Centre for Additive Manufacturing School of Engineering and Advanced Technology Massey University New Zealand

[3] Free Air Finger Painting on a Projected Canvas Jon Moeller,Nic Lupfer,Bill Hamilton,Haiqiao Lin.Interface Ecology Lab Texas A&M University Department of Computer Science and Engineering 3112 TAMU College Station, TX USA

[4] Tracking of Flexible Brush Tip on Real Canvas: Silhouette Based and Deep Ensemble Network Based Approaches JOOLEKHA BIBI JOOLEE , AHSAN RAZA , MUHAMMAD ABDULLAH, AND SEOKHEE JEON Department of Computer Science and Engineering, Kyung Hee University Global Campus, Yongin 17104, South Korea

[5] Drawing into the AIR CANVAS: Designing Embedded Visualizations for Augmented Reality Benjamin Bach* University of Edinburgh, UK Ronell Sicat Harvard University, MA

[6] C. Galambos, J. Kittler, J. Matas. Gradient-based Progressive Probabilistic Hough Transform. IEE Proc. of Vision, Image and Signal Processing, vol. 148 (3), pp. 158-165, 2001.

[7] http://opencv.willowgarage.com/wiki/

[8] R. Laganière. OpenCV 2 Computer Vision Application Programming Cookbook. Packt Publishing 2011.