IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. VI (May-Jun. 2014), PP 01-09
www.iosrjournals.org

Virtual Teaching and Painting Platform for the Colour Blind

[1]Ananya Roy,[2]Abhishek Rudra, [3]Souvik Bhattacharya, [4]Arnab Pal
[1]Department of Computer Science & Engineering, Institute of Engineering & Management, Kolkata, India
[2] Department of Computer Science & Engineering, Institute of Engineering & Management, Kolkata, India
[3]A.K Chaudhuri School of I.T, University of Calcutta, Kolkata, India
[4]Department of Computer Science, Chandannagar Government College, India

Abstract: *Education involves perception of colour and unfortunately some of us are not blessed with this gift of proper vision. Lacking the ability to distinguish certain colours is commonly known as Colour Blindness. In schools and colleges, colour blind students are not supported much in the classroom, some lose confidence and struggle to cope. Often colour blind teachers are unable to distinguish important colours of pens or markers and for an efficient and interactive learning environment it is seen that the results are tilted in favour of classrooms where multiple colours are used on the board. The numbers corresponding to people having this disease(or its variations) are compelling enough to make it highly imperative for us to seek a solution for it. Hence, we propose a software which can be used as a virtual teaching and virtual painting platform by these differently abled people. Built on the Graphical User Interface Development Environment(GUIDE) in the numerical computing environment MATLAB, this system helps the user to identify the colour of the object (marker) withwhich he can teach virtually by moving it in air.These movementsget traced on the screen which serves as a virtual white board.*

I. Proposed System

The software we propose can be used as a dual purpose tool.
1. It can be used as a virtual painting tool for the color blind children
2. It can be used as a virtual teaching tool for the color blind teachers.

Both purposes are served by a set of functionalities which use the basic web camera of a computer system only as a resource.

- The system using the web camera records the video of the user who holds an object (usually a marker) of a particular color(one of Red, Green, Blue, Cyan, Magenta and Yellow) as a tool used for drawing or writing on the V**irtual White Board** present on the interface.
- This video is displayed separately on the left side of the interface for the user's convenience in case he/she wants to track his/her hand movements. The user holds this object in front of the camera for some time for the system to detect the color of the object on a real time basis.
- As soon as the color is detected, the user is prompted about its information on the virtual white board mentioned above. After this, the paint tool gets activated which traces the movement of the object in the user's hands and paints the area (pixels) on the white board according to it.
- The tracing of movement is visible to the user and he/she is able to see a rectangular bounding box, surrounding the colored portion of the object, which moves along with the object simultaneously depending on the user's hand movement.
- The painting occurs with the color detected by the system in the initial phase. In case the user wants to add a space or a pause in the process of writing or drawing then he/she can click on a**Pause** button present at the bottom of the interface which halts the paint tool accordingly.
- The user can also clear the virtual white board using a **Clear** button.
- While writing or drawing if the user feels the need to change the color which he wants to use then he/she can do that using a **Change Color** button present on the interface. Once this button is clicked, the system re runs the detection tool to detect the color of the next object to be used by the user after which the user can continue his work.
- The user can also save his work using a **Save** button present on the interface which saves the image of the virtual white board in .jpeg format.
- The system also allows the user to paint or write on an image or a set of images which can be used as the same virtual board instead of the plain white one given before. This can be done by clicking on a button named **Open** present on the same interface. It also allows us to open a set of multiple images .

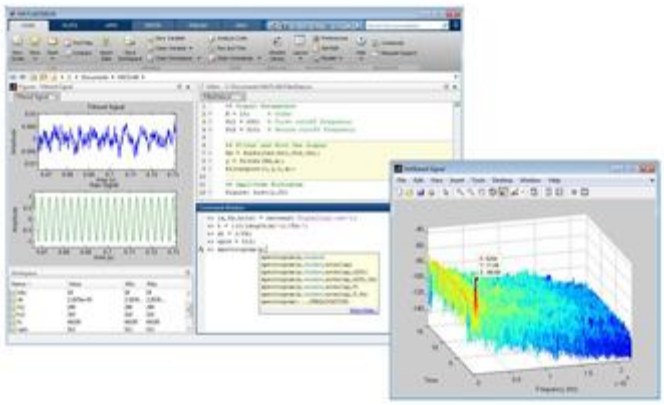
- **Open** functionality provides an experience equivalent to that of a power-point presentation on which the user can mark or write anything virtually. A set of buttons **Next** and **Previous** allow the user to switch between the set of images selected by the user before.
- After the user has finished working with the system he/she can close/terminate it using a button named **Close** provided for the same on the interface. The juxtaposition of all the aforementioned functionalities in our proposed would serve as a handy and uniqueteaching or drawing tool which can prove to be highly effective in addressing the needs of the color blind children as well as the teachers.

II. Why Matlab?

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java™.



Key Features:

- High-level language for numerical computation, visualization, and application development
- Interactive environment for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations
- Built-in graphics for visualizing data and tools for creating custom plots
- Development tools for improving code quality and maintainability and maximizing performance
- Tools for building applications with custom graphical interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel®

III. Advantages Of Matlab Over Open Cv

- **Ease of use:** Matlab is a relatively easy language to get to grips with. Matlab is a pretty high-level scripting language, meaning that you don't have to worry about libraries, declaring variables, memory management or other lower-level programming issues. As such, it can be very easy to throw together

```
I = imread('someImage.jpg');
imshow(I)
```

some code to prototype your image processing idea. Say for example I want to read in an image from file and display it. In Matlab, you could write this

- Easy, right? Now, if you wanted to do the same using OpenCV, it would look like:

```
#include "cv.h" //main OpenCV header
#include "highgui.h" //GUI header

int main()
{

// declare a new IplImage pointer
IplImage* myimage;

// load an image
myimage = cvLoadImage("someImage.jpg",1); //change the file name

//create a new window & display the image
cvNamedWindow("Smile", 1);
cvShowImage("Smile", myimage);

//wait for key to close the window
cvWaitKey(0);
cvDestroyWindow("Smile");
cvReleaseImage(&myimage);
return 0;

}
```

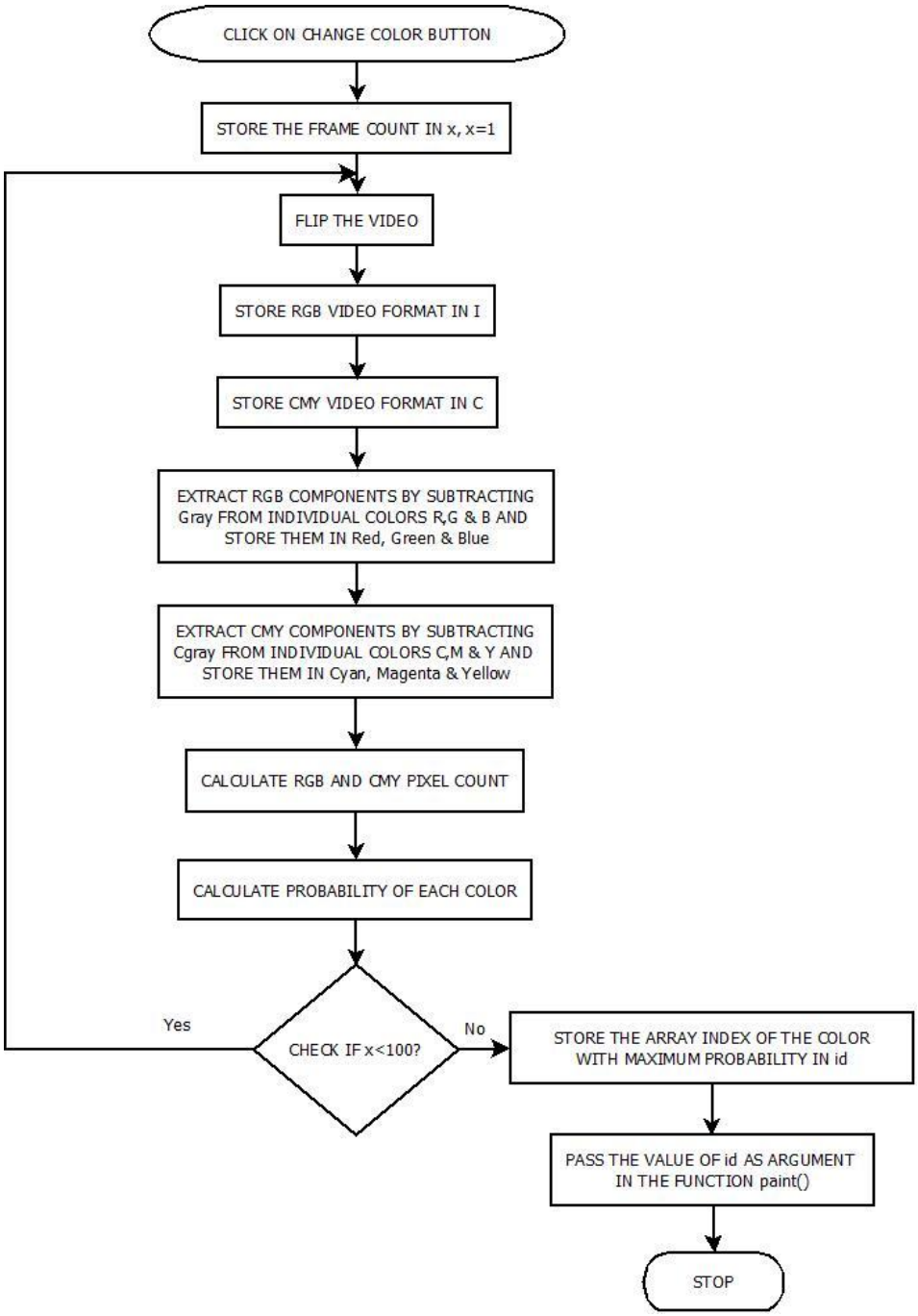
- **Memory Management:** OpenCV is based on C. As such, every time you allocate a chunk of memory you will have to release it again. If you have a loop in your code where you allocate a chunk of memory in that loop and forget release it afterwards, you will get what is called a "leak". This is where the program will use a growing amount of memory until it crashes from no remaining memory. Due to the high-level nature of Matlab, it is "smart" enough to automatically allocate and release memory in the background.

- **Development Environment:** Matlab comes with its own development environment. For OpenCV, there is no particular IDE that you have to use. Instead, you have a choice of any C programming IDE depending on whether you are using Windows, Linux, or OS X. For Windows, Microsoft Visual Studio or NetBeans is the typical IDE used for OpenCV. In Linux, its Eclipse or NetBeans, and in OSX, we use Apple's Xcode.

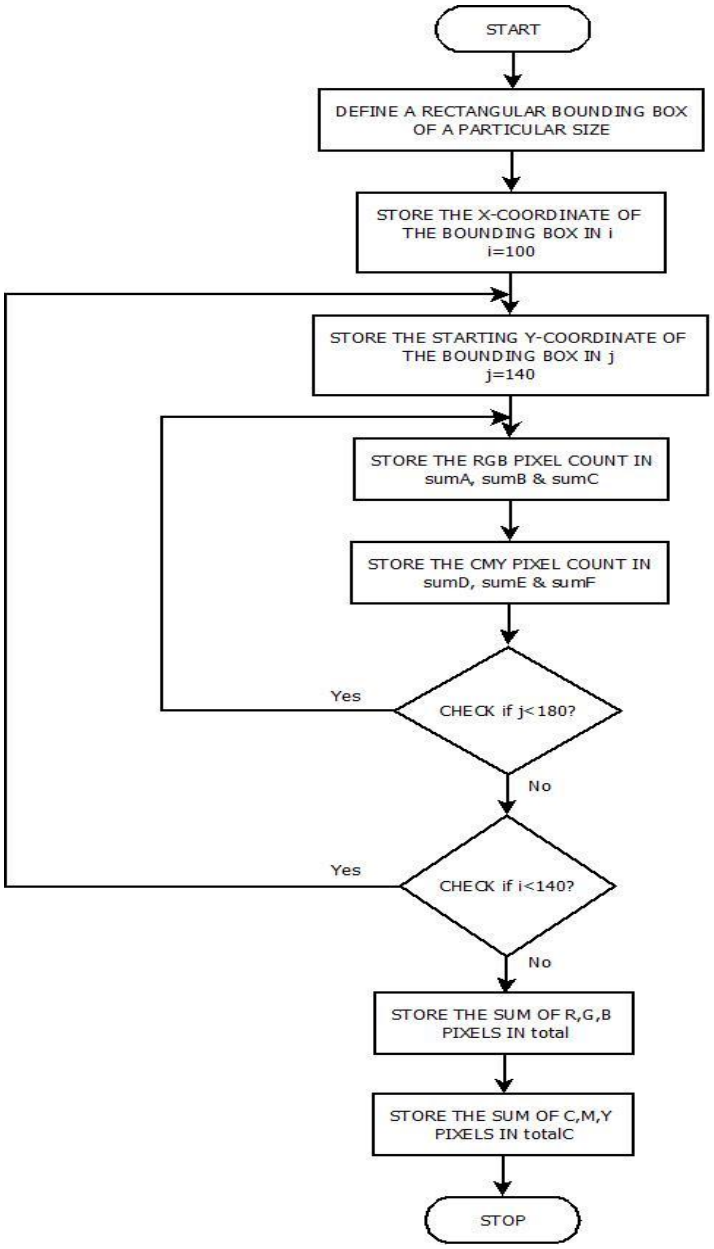
IV. Algorithms

Change Color

1. Click on the Change Color button.
2. Flip the video.
3. Convert the video into CMY format and finally store the grayscale forms of both formats in Gray and Cgray.
4. Find the individual color forms R,G,B,C,M,Y of the image by subtracting the grayscale forms of the image(s) from the individual color components of the same.
5. Assume a rectangular bounding box of a particular area.
6. Calculate the R,G,B,C,M,Y count for each pixel within the bounding box.
7. Calculate the sum of R,G and B counts and C,M and Y counts and store them separately.
8. Calculate the probability of each color by dividing individual pixel count of each color with the sum of RGB count(for red, green and blue only) or CMY count(for cyan, magenta and yellow only).
9. Repeat the steps 2 down to 8 for a designated no. of frames.
10. Find the color with the maximum probability and pass its id to the paint function.



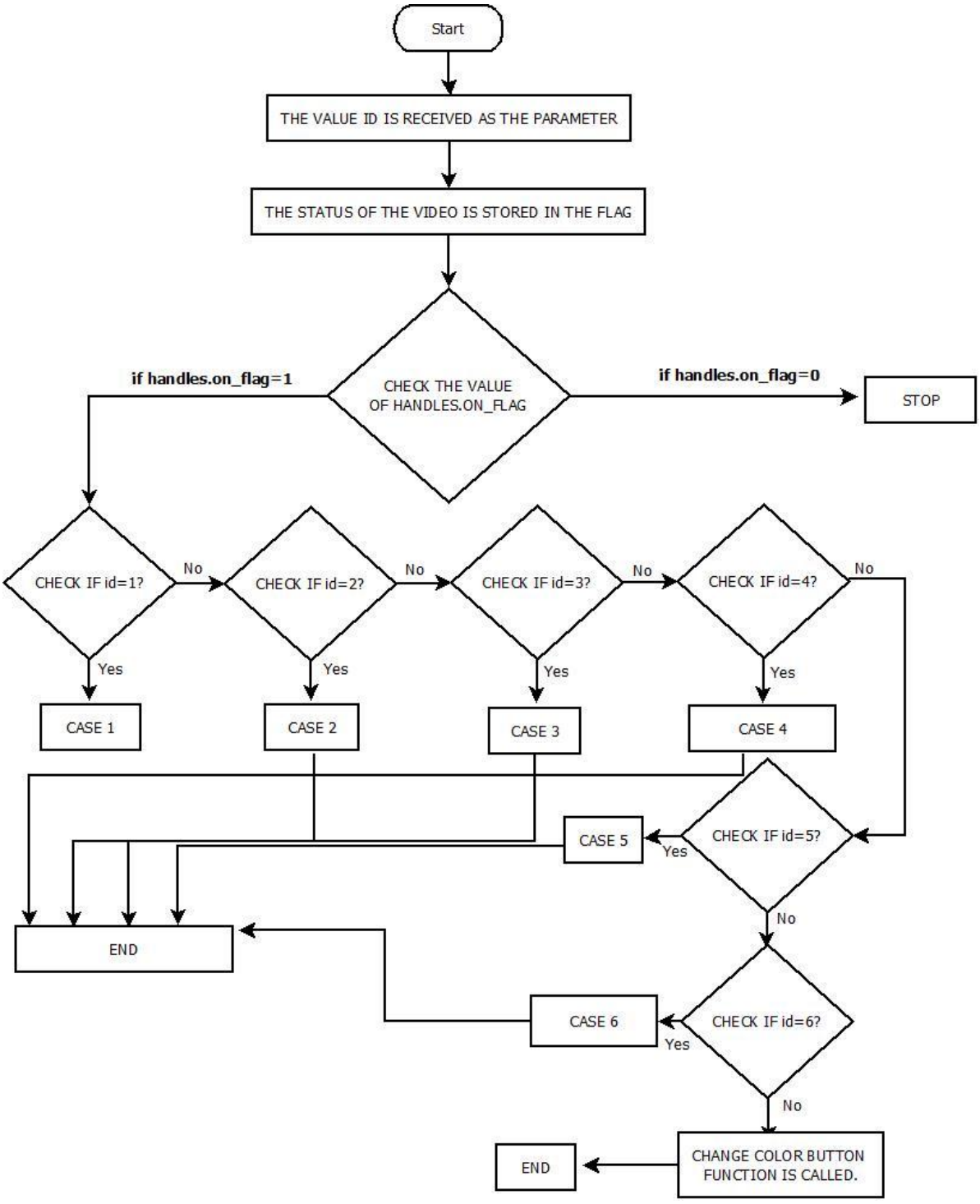
Colour Detection Algorithm



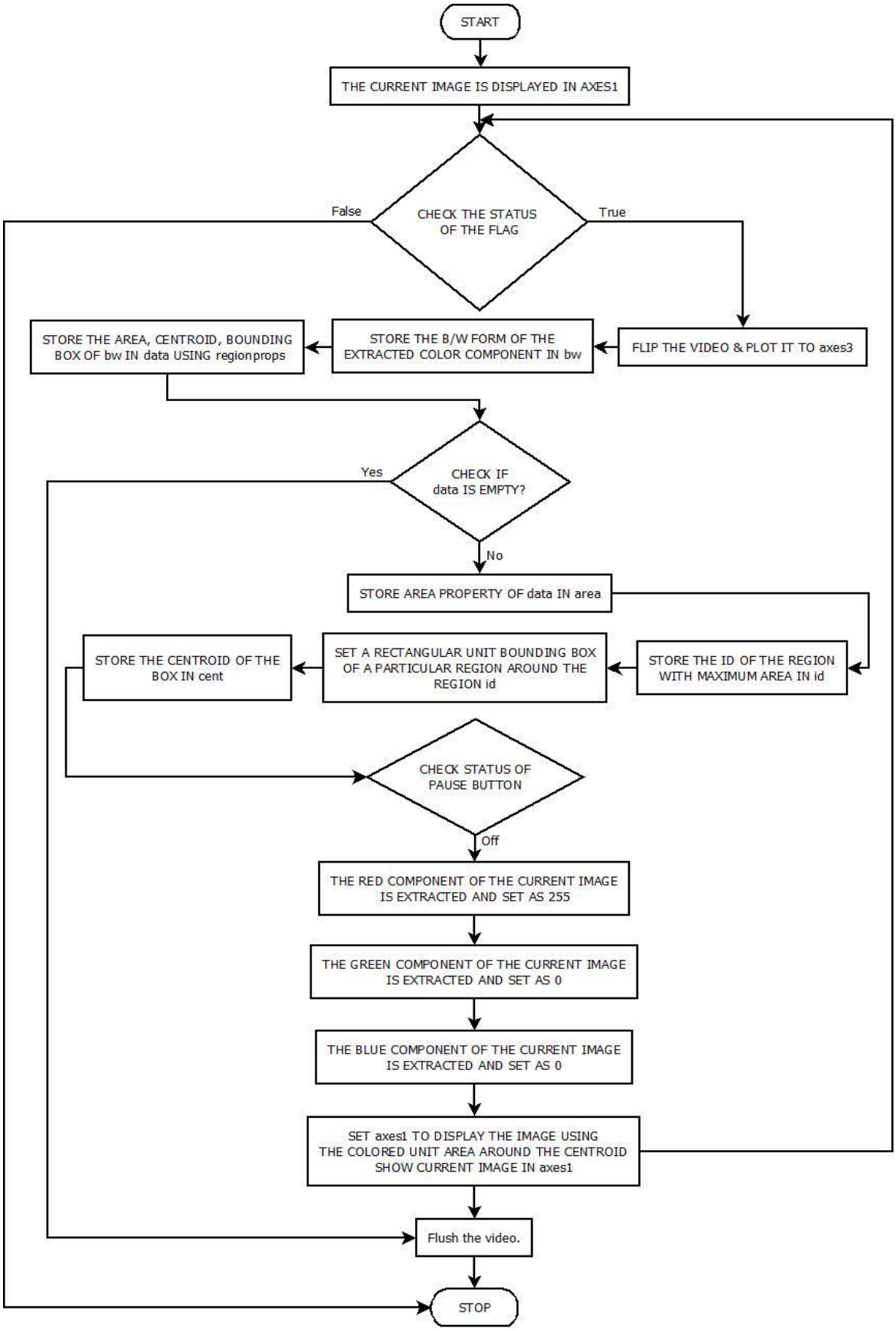
Algorithm For Colour Extraction

Paint

1. The id of the color with maximum probability is received.
2. The current color is set to the color with the value id.
3. Flip the video.
4. Display the video in the designated axes.
5. Find the area, centroid and bounding box properties of the black and white form of the extracted color component of the image.
6. Find the region of the image with maximum area and assume a rectangular bounding box around that region.
7. Find the centroid of the bounding box.
8. Set the value of the color components(R,G and B) of the adjoining pixels of the centroid as 255 or 0 according to the color with the value id.
9. Display the current image in the designated axes.
10. Repeat steps 7 and 8 until the status of pause button is true.
11. Repeat steps 5 down to 10 till the video is ON.
12. Flush the video and stop.

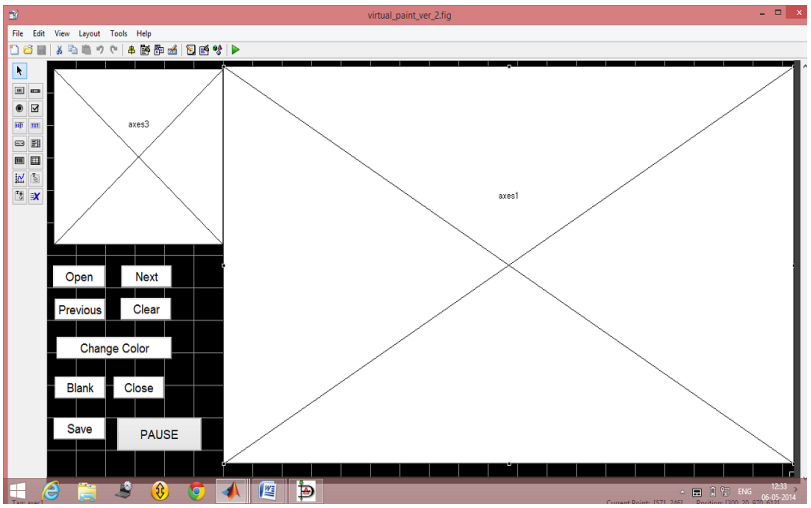


Paint Algorithm



Paint Algorithm For A Particular Colour

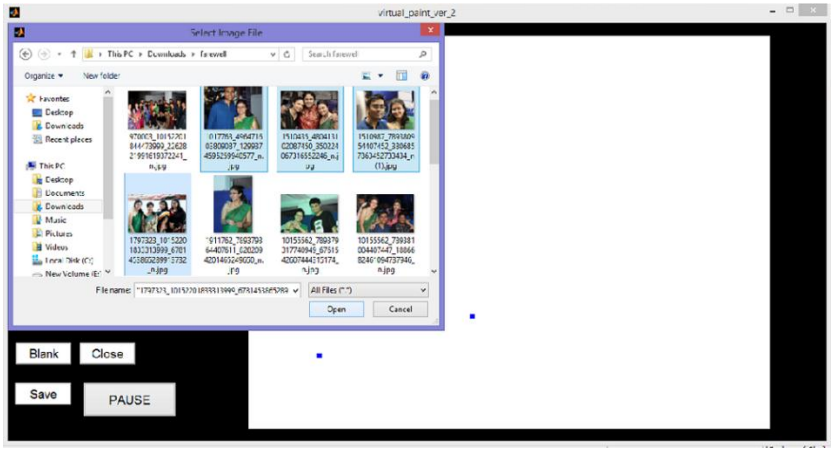
Screenshots



Virtual Teaching And Painting Interface

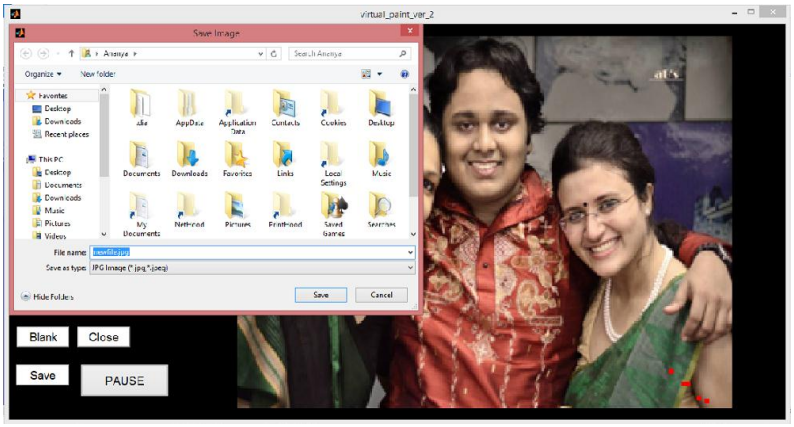
Detection Of Red Colour

Painting With Red Colour



Opening A Set Of Images

Image Used As A Base To Write On



Saving A Work

V. Conclusion

Detection of colour essentially forms the most vital aspect of proper vision. This project aims at automated identification of colours, which is one of the most commonly found visual ailments in this part of the world. Proper identification of colour by computers opens up a vast new horizon. Its application may range from teaching tools for visually impaired to automatic detection of traffic signals. The project provides numerous avenues for further advancements in machine intelligence & computer vision.