Available online at www.ijapie.org

INTERNATIONAL JOURNAL OF ADVANCED PRODUCTION AND INDUSTRIAL ENGINEERING

IJAPIE-2018-01-130, Vol 3 (1), 48-51

IJAPIE
Connecting
Science & Technology
with Management.

A Journal for all
Products & Processes.

Object Detection and Tracking in Real Time Video Stream

N. Jayanti
(Department of Electronics & Communication Engineering, Delhi Technological University, Delhi, India)
Email: njayanthi@dce.ac.in

Abstract : To achieve fully automatic surveillance of some specific color objects, an intelligent real-time detection method based on video processing is proposed. The main aim of this paper is to identify the colors and use them to achieve their applications. Proposed algorithm is used to detect a specific color and also to track it in the live video feed which could be eventually used for many different applications like surveillance camera, fire detection in cases of forest fires, etc. For the color recognition part , several stages such as image subtraction, noise filtering, binary image and blob extraction is used to recognize a specific color in the video feed. Then the corresponding pixels on the GUI are drawn to track where all the color has been. This might find application in various areas; one such area in which this has been used often is in detection of forest fires.

Keywords: Object detection, tracking, video processing, GUI (graphic user interface).

I. INTRODUCTION

An image is digital if that includes a specific intensity for each individual pixel and that too in three or four columns is known as a color image. An image when defined in terms of color will have three intensities each for a different color and can even measure the light's chrominance.

For such results which can be deciphered easily and are acceptable visually and also sufficient to give us with at least three samples or three intensities for each different color space which can be thought of to be coordinates of the color space. . In color image processing, the RGB color model has less computational complexity than other color models and hence is adopted to describe the object pixels [1]. Other color spaces however are also used such as the CMYK and the HSV and they can be used in other contexts such as computers, newspapers etc. Whenever a model of color adds to the previous intensity it is known as an additive model of color. In RGB all colors are added together to form a large variety of colors and provide us with the amazing visual that could be often seen.

RGB comes from the first letters of all the three colors it contains i.e. Red, Green and Blue. It has always been used in normal photography and thus is always used to sense, represent and display images in many different electronic representations such as the display of a computer or a television.

Color detection is also used in vision tracking system is presented in [2]. RGB model will always depend on the device it is running and it may be interpreted or reproduced in different manners because of the different color elements behind their usage. This detection scheme is also being used in robotic arms for sun tracking to control the position of solar panel [3].

RGB model has been used in edge detection algorithm for surveillance in image as can be seen in [4] and also for forest fire surveillance [1]. The proposed method is try to recognize the different colors in the live video feed. But only the color red will be useful to us. An algorithm is developed to recognize the red colored pixels captured in the image (A snapshot of the live video feed). Then filtering techniques are used to recognize the exact position of the colored pixels and consequently blob analysis technique is used to identify different features of the colored pixels if required. Nowadays machine learning is also playing its role in color detection to increase the efficiency of Traffic light system [5].

So in this paper it is required to create a GUI in our software. The next step in the proposed method is to virtually draw on the blank GUI using the red color recognized in the corresponding video stream. For this to run successfully there should not be any outlier red color in the video. Since the salient red figure in the video feed are not recoganizable, outliers might disturb the drawing and random pixels could be marked in that case. For drawing , the corresponding pixels in the GUI is marked. Different colors are made available to draw which can be selected from the drop down menu with the help of the mouse and the corresponding drawing can be saved in various available formats such as .bmp, .jpg, and .png.

II. ALGORITHM IMPLEMENTED

The process to detect a colored object and to track it using matlab in addition to any good resolution camera is explained with the help of a Flowchart drawn in Fig. 1. In this paper the camera which is being used is the already integrated webcam which comes in a laptop. Any external USB camera can also be used instead of an integrated camera to capture the live video stream and can be interfaced with matlab by including the DCAM Hardware support package from the various hardware support packages it provides [6]. While adding the package for the external camera the resolution mode with which it is required to work on needs to be set accordingly. The functioning of each process steps are elaborated in the subsequent subsections.

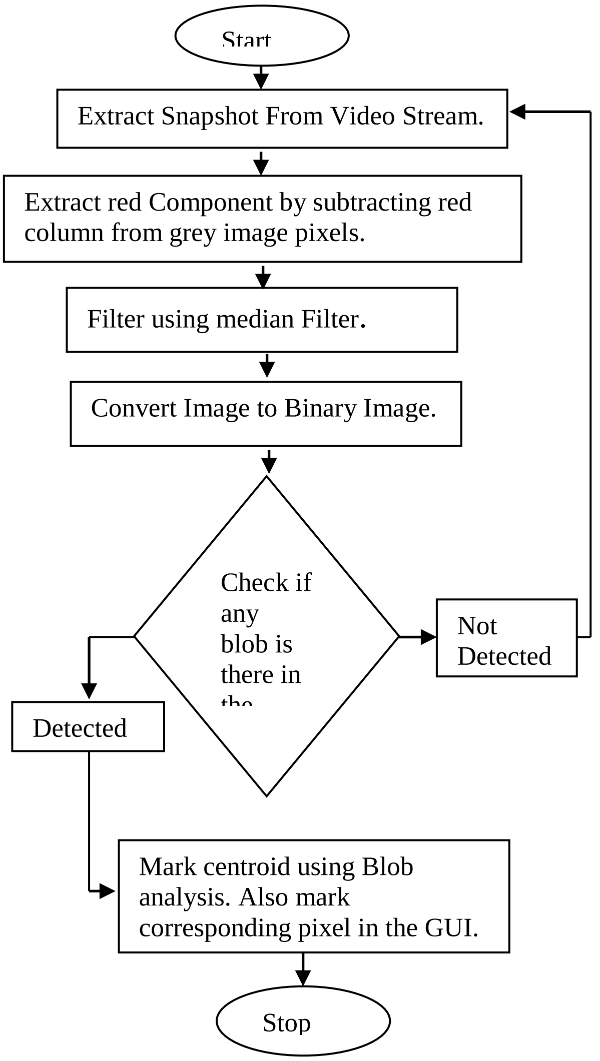
Start
Extract Snapshot From Video Stream.
Extract red Component by subtracting red column from grey image pixels.
Filter using median Filter.
Convert Image to Binary Image.
Check if any blob is there in the
Not Detected
Detected
Mark centroid using Blob analysis. Also mark corresponding pixel in the GUI.
Stop

Fig. 1 Flowchart of the process flow

A. Capturing Image from Live Video Feed

[1] First initiate the video stream in MATLAB and give parameters to it to return to us an RGB image.
[2] Ttake continuous snapshots (images) from the video stream and process them each in the same manner.
[3] This is taken care of with the command ‘step’.
[4] Each matrix returned by using the above command is of the size of the resolution of the video stream with each R, G and B component. Size of the matrix returned is: [480 640 3].

B. Extraction of Red Component

- Get only the red component of the image by subtracting the grey scale values from the red column of the matrix.
- Grayscale image is a type of an image in which the value of each pixel will be a single value unlike the RGB model, such sort of images can be said to be a mixture of black and white and their different combinations thus representing many shades of grey. This shade of grey will be black at the lowest intensity and white at the highest.
- This type of grayscale images are very different form normal black and white images if the view point of a computer and its imaging is considered. Black and white images are known as bi-level since they have only two values whereas a grey image has lots of shades.

C. Filtering Noise

[5] Noise: Any random variation in any property of the image from the general pattern or the surrounding region is known as Noise. This can be developed as a result of any electronic disturbance. Any sensor or wrong circuit can trigger this in any device.
[6] Filters such as the Gaussian filter or the mean filters are not able to enhance the edges and remove the noise uniformly. For this purpose median filter is used which can produces edges of higher quality as compared to the other filters.
[7] Median Filter: It is one of the most used filters because it is better at removing noise than the other filters available. Also after removal the edge quality is better in this filter.
[8] Now explaining the working and the idea behind a median filter. What it does is it puts all the values surrounding the pixel into an array, if it is a boundary pixel it appends that many zeroes to the array, now it arranges the array in increasing order and then selects the median of the array. The original pixels whose neighbors were taken out is now replaced with a new value which is the value of the median, the array containing all the neighbors is known as the "window". Process is iterated again for each one.

Working of a median Filter has been shown in Fig. 2:

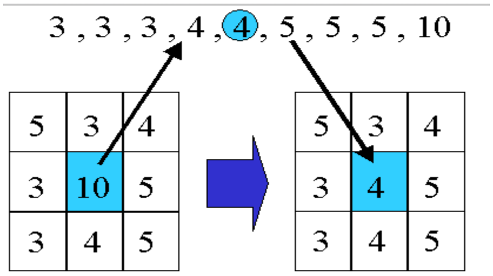

Fig. 2 Working of a Median filter

The same process is repeated for each pixel of the image. For pixels on the boundary the empty pixels are given a value of zero.

D. Converting Normal Image to Binary Image

[9] The image is then converted to a binary image by using a particular threshold of the red level.

[10] This conversion to binary is done to make the further steps less complex to solve.

[11] The threshold for the red color is decided by running it at multiple thresholds and then deciding the optimum one based on the lighting of the room.

E. Blob Analysis

[12] The method of detecting blobs works on a very simple principle. What it does is that it detects regions in the image which are very different in properties such as brightness or any other thing in comparison to the region that surrounds it.

[13] Basic Scenario of blob analysis consists of three major steps:

i. Extracting the region.

ii. Refining the blob identified.

Analyzing the final blob.

F. Recognition

[14] Once the blob is recognized the function returns the bounding box and the centroid of the blob.

[15] Mark both the centroid and the blob on the video stream.

Define a threshold for the maximum and the minimum area of the blob

III. EXPERIMENTAL RESULT

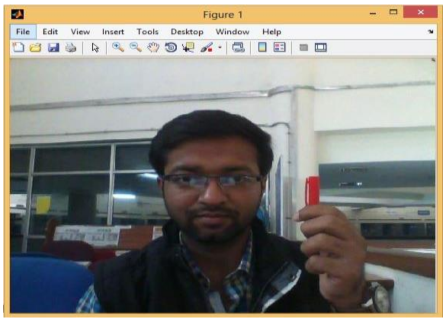

Fig. 3 A snapshot from video stream.

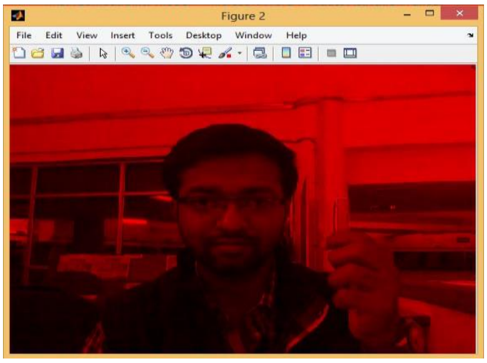

Fig. 4 Snapshot with Red Column only

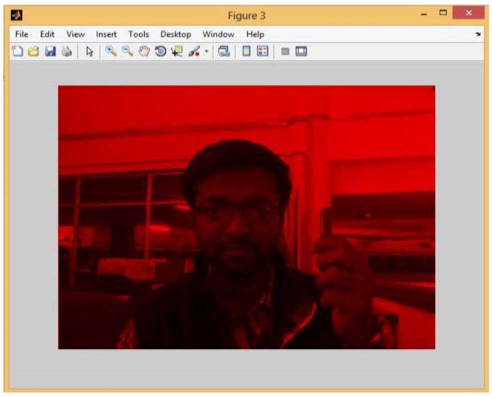

Fig. 5 Gray scale Snapshot

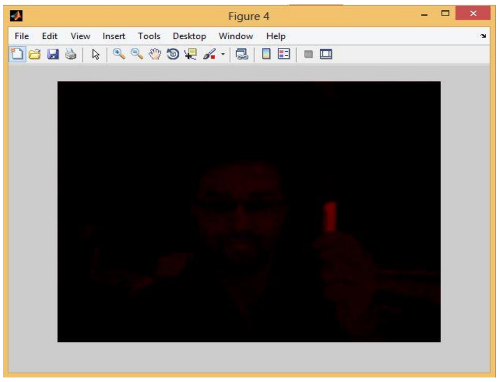

Fig. 6 Snapshot after Subtraction (Only Red Component)

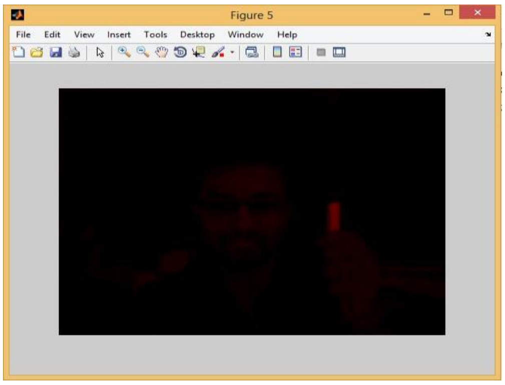

Fig. 7 Snapshot after applying Median Filter

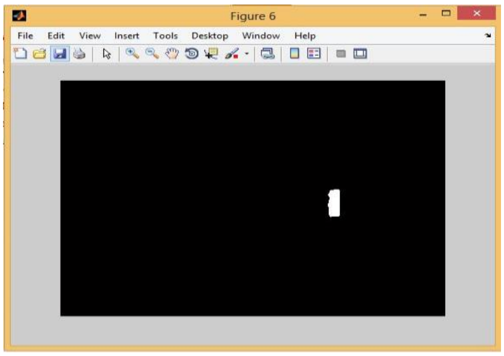

Fig. 8 Binary converted Snapshot

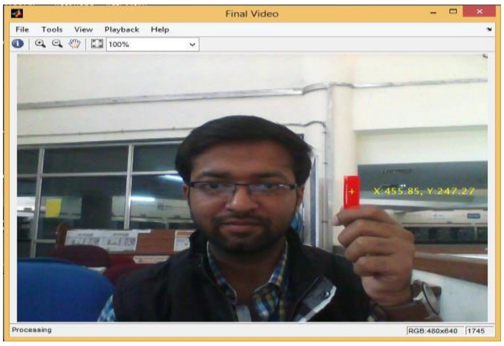

Fig. 9 Finally Recognized Object

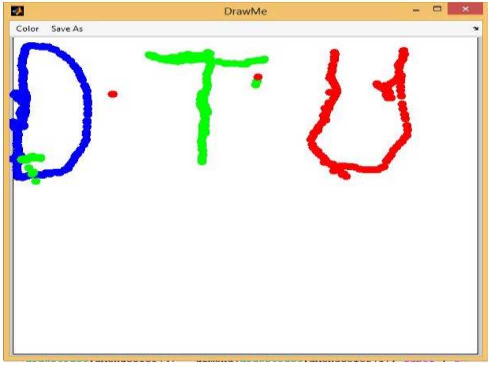

Fig. 10 Trace of the obect detected

[16] Snapshot from live video: First we have to start up the video stream using the command imaq.VideoDevice [6]. Once the video stream is live we will continuously take snapshots using ‘step’ command.

[17] Extracting Red component: Now to extract the red component part of the image we will convert the image into its grey form shown in Fig. 5, by using the command rgb2grey which finds out the grey component of a particular pixel by using 29.89% of the red component, 58.70% of the green component and 11.40% of the blue component [6]. From this obtained image we subtract the red column of the image shown in Fig. 4 to get the red component image as a separate one which is shown in Fig. 6. For extracting any other component other than red, that particular column can be subtracted.

[18] Fig. 7 shows the image after applying the median filter to the red extracted image.

[19] The final binary image is used to detect the red objects after defining a proper threshold based on the image surrounding and then we have only allowed values according to the threshold level to show in the binary image shown in Fig. 8.

[20] Finally it is necessary to know which area is the one in which our object resides and then blob analysis method is used to find the centroid of the recognized area. This is then marked in the live video stream to show that the correct object is being recognized.

.

[21] Then go on marking the complete object with red color in the video feed by continuously applying the same algorithm. A GUI is created to track and mark the corresponding pixels on which the object has been to keep a track of its path as shown in Fig.10.

V. CONCLUSION

The lighting of the room is one big factor and we have to train our machine in such an efficient manner that it is able to overcome all of that. The proposed methodology could be used for multiple applications in places that are in attention nowadays especially those parts which are growing. As previously stated, the color detection is being used for forest fires, it might even find applications in controlling things while wearing different colored caps on fingers and make pre-defined gestures to control some application.

REFERENCES

[1] Chen, Thou-Ho, Cheng-Liang Kao, and Sju-Mo Chang. "An intelligent real-time fire-detection method based on video processing." Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on. IEEE, 2003, pp. 104-111.2003.

[2] Jin, Ruibing, and Jianliang Wang. "A vision tracking system via color detection." Control and Automation (ICCA), 2016 12th IEEE International Conference on. IEEE, 2016.

[3] Szabo, Roland, and Aurel Gontean. "SCORBOT-ER III robotic arm detection and control with color recognition with the possibility for using in sun tracker applications." Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), 2017 IEEE International Conference on. IEEE, 2017.

[4] Park, Chang-Hee, et al. "A design of real time detection IP with color detection for surveillance." SoC Design Conference (ISOCC), 2016 International. IEEE, 2016.

[5] Binangkit, Jeffrey L., and Dwi H. Widyantoro. "Increasing accuracy of traffic light color detection and recognition using machine learning." Telecommunication Systems Services and Applications (TSSA), 2016 10th International Conference on. IEEE, 2016.

[6] Mathworks. (2011). Global Optimization Toolbox: User's Guide (r2011b). Retrieved November 10, 2011 from www.mathworks.com/help/pdf_doc/gads/gads_tb.pdf.