Traffic Light Recognition System for People with Color Blindness

Conference Paper · January 2015

DOI: 10.3850/978-981-09-4424-7_63

2 authors:

Timur Ibrayev
Nazarbayev University

Radkhan Sarmukhanov
Nazarbayev University

Traffic light recognition system for people with color blindness

Short paper (*ICP 2015*)

***Abstract*—This paper suggests the solution of traffic light color recognition problem for people with color vision deficiency. Solution will include the developed Windows OS compatible application that is further adapted to android platform. The application was initially developed by the C++ programming language with the Open Computer Vision (OpenCV) library. During the development of program the combination of Image Thresholding and Hough Circles object detection methods was used to detect the traffic light itself. After that the color of the traffic light was identified and written notification defining the color name was displayed on the screen. The algorithm is then used to write an application prototype for android based devices. It is expected that this program will assist people with color blindness in traffic light color detection.**

Keywords*—color blindness, traffic light recognition, object recognition, shape recognition, color recognition, OpenCV, C++, Android

I. Introduction

Approximately 500 million people are affected by color vision deficiency nowadays. Unique property of this disease is that it cannot be cured, thus people with color blindness become vulnerable at some situations. Experts imply that red-green color deficiency is the most widespread defect type. Hence, traffic light color identification can probably be considered the most encountered problem for color blind people. Spread of this disease among pedestrians and drivers is likely to increase the number of car accidents. Thus, the development of the application that will assist color blind people with traffic light color recognition cannot be underestimated.

Today image and video processing are mainly performed with the usage of OpenCV library. The application that we propose in this paper is based on two different functions: one of which is used for color thresholding and other is for shape detection. The main idea is by combining two functions implement algorithm for detecting traffic lights on camera or video stream. Consequently, program will recognize the color of the displayed signal by applying different color filters.

Finally, the program is planned to notify the user with the corresponding sound signal. This application was firstly designed for computers operating on Windows OS. Since web-camera is expected to be the main input source, laptop's location in the car is required. However the algorithm is further used with Android programming language to create prototype of mobile application. This option potentially will be more convenient for drivers and pedestrians.

II. Procedure

This section will explain the step-by-step procedure of the application realization. Firstly, the restrictions applied to the sample video data will be described.

Secondly the procedure of how object detection was realized in the program is described by explaining two different methods used to detect objects based on either color or shape. After explaining these and defining the drawbacks of using two methods separately, the core of the proposed algorithm will be presented by describing how these two methods complement each other in order to form traffic light recognition program.

After explanation of how the program will find traffic lights on video stream, the color identification technique used in the program will be explained.

A. Sample Data

The very first step in the design process was to obtain appropriate video samples. This version of the program was only considering the case when traffic lights were clearly visible and close to the recording device. This was done in order to see if the proposed logic works as required when there is a minimum amount of disturbing objects on the video. However, in turn, this means that program still should have the future improvements to deal with the cases when there is more objects introducing noise. The images of the frames with red, red and yellow, and green traffic light signals being on are displayed in Fig. 1.

B. Image Thresholding Method

This is a method that is basically used for object detection that is based on its color properties. The BGR colored image is converted into HSV color space image. Then HSV image is

Fig. 1.The sample of Traffic Light used in the program design with red (a), red and yellow (b), and green (c) light signals being displayed

thresholded by manually adjusting the HSV values using *inRange* function and track bars, so that the required color is represented as white and the rest of the image is represented as black color. Consequently, *findContours* function together with *moments* method applied to find and store the white colored contours from thresholded image. The example of such image conversions for thresholding red color is illustrated in Fig. 2.

As it can be seen the red colored object is easily extracted by thresholding HSV values. However, the drawback is that function *findContours* finds any white colored contour that was found in thresholded image. The morphological operations (such as erosion and dilution) are able to eliminate noise effects, but still there may be objects of not only the same color, but also of the bigger size than targeted traffic light. Consequently, the function may not only be unable to recognize the traffic light signal but also by making an error recognize another object instead of it. Moreover, it does not make a difference between the shapes of the contour found. Hence, the design of the proposed program adds another function to detect the shapes of the objects in the video stream.

C. Finding Hough Circles Method

HoughCircles function is introduced into the program to find the objects of the circled shape. In order to use this function, the original frame image from the video is initially converted to grayscale image and then blurred. The conversion is done because the function works on only 8-bit images. Blurring is then applied to reduce the noise and avoid the false circle detection. Example of applying this function to detect the circled shape of red colored signal is illustrated in Fig. 3.

As it can be seen the *Hough Circles* function has detected only one circle, which is the shape of traffic signal. However, the drawback of using this method alone for object detection is that if there are multiple objects with circled shape, the function will find all of them. For the specific problem that the following application is aimed at finding circle shapes without much emphasize on its color is not enough.

D. Combination of two methods to achieve better object detection

The main advantage of the proposed application is the implementation of two functions together.

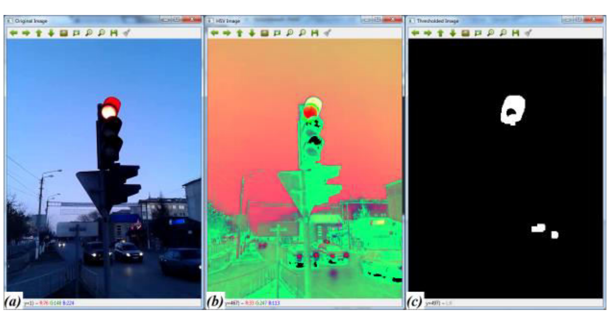

Fig. 2. The example of red traffic light contour detection using conversion of original image frame (a) into HSV color space image (b) and then applying threshold (c)

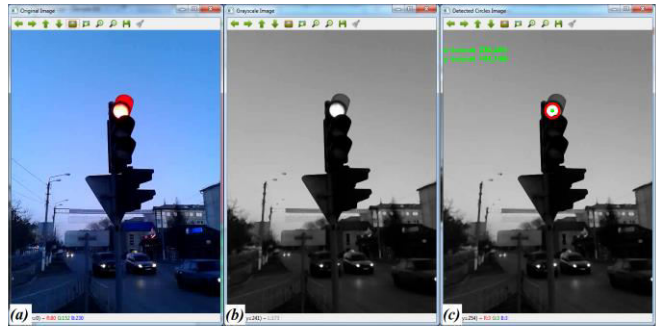

Fig. 3. The example of red traffic light circle shape detection using conversion of the original image (a) into grayscale image (b), and blurring it and displaying the circle shapes found (c) by the Hough Circles method

Firstly, the program searches for circle shapes using Hough Circles and, as it can be seen from Fig. 3, defines the boundary coordinates of them. Then, the coordinates are used to compare with every contour that has been found by thresholding the image to required color. It can be said that the main role of these coordinates is to filter out unnecessary contours, and to only display targeted traffic light signals. After potentially traffic light signal was detected, it is tracked and displayed by the program. Fig. 4 illustrates an example of how this object detection works for red signal.

However, this only detects the position of the required object on the image frame and serves as a training stage for determining HSV values for each of the required colors, but does not define it explicitly. For the program to automatically identify which color was thresholded another function was introduced.

E. Color Detection Technique

The color detection part of this program is working on the fact that the sets of HSV values for each of the three colors are determined in training stage. Each of these sets is then used to create color filters for three colors of traffic signal. These are then sequentially used in a loop form to create a function that automatically changes the threshold values. In case the object is found, the function stops changing the filter and displays the corresponding color value. When the object is lost because of traffic signal change, the application recalls the function and make color filter change to the next one. For example, one turn of the color recognition is illustrated in Fig. 5.

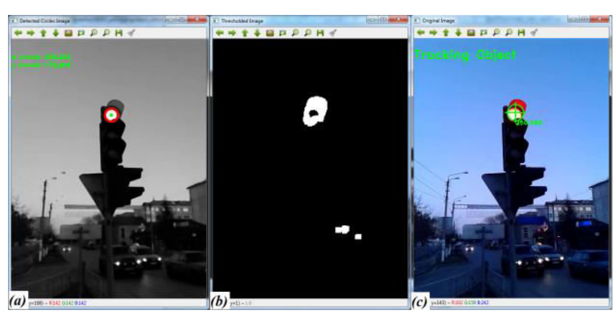

Fig. 4. Example of object tracking by comparing the boundaries found by HoughCircles function (a) with coordinates of contours found by thresholding (b) and displaying the tracked object on original image (c)

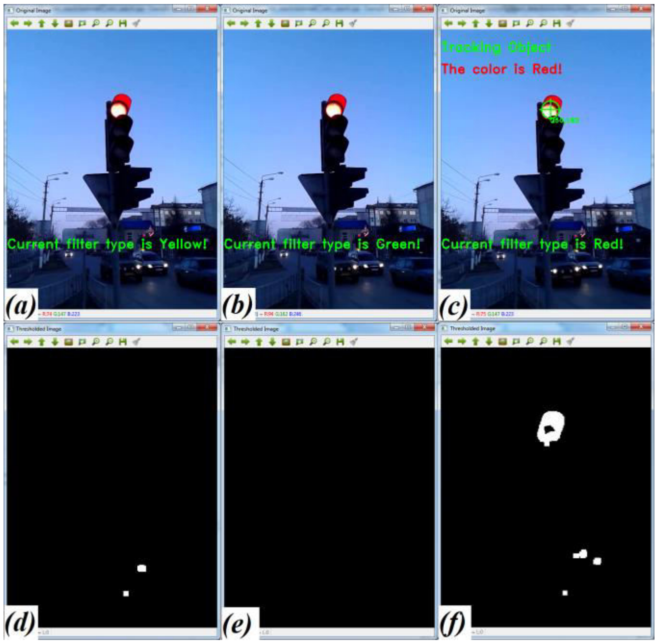

Fig. 5. The color detection implemented by the sequential application of yellow color filter (a), green color filter (b), and red color filter (c) to original image, each of filter being thresholded by HSV values for yellow color (d), green color (e), and red color (f)

The algorithm that was utilized to construct the application for Windows can be graphically represented by the flowchart shown in Fig. 6.

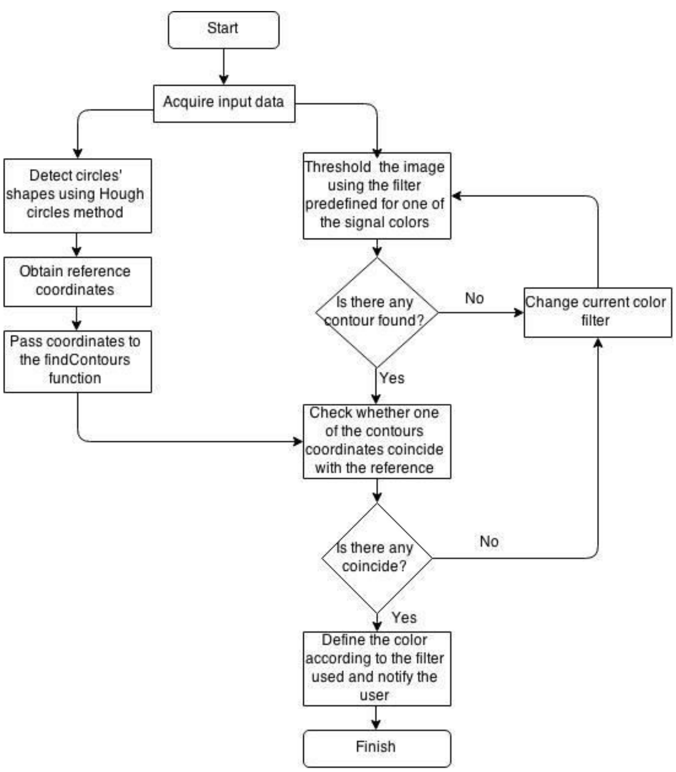

Fig. 6. Flowchart of the proposed algorithm for traffic light recognition application

III. Results of Windows program

The Fig. 7 illustrates the cases of successful identification of red, yellow, and green signals by the program on its own, without any user manipulations on HSV threshold values.

Despite the program works appropriately for the determined range of HSV values, there are still occasions when the program misinterpret displayed signal. The common case is when traffic light displays both red and yellow signals. This can be seen from Fig. 8.

Moreover, it should be remarked that such scenario of signal identification is very simplified for initial design purposes, but will not work if there are more objects on the video that can cause errors in identification processes.

IV. Android Implementation

After the program for Windows, despite of being not fully finished, worked well for traffic light used as a main sample, we tried to write an application having the same purpose and algorithm for Android platform devices. Precisely, the main interest was on constructing an application for mobile phones, because there is now increasing number of android mobile phone users as well as the rapid developments on increasing their operation capabilities.

The application at its final version is planned to be capable of identifying the traffic lights from the camera preview of the mobile phone, recognizing the color of signal being displayed, and finally providing sound notifications to the user.

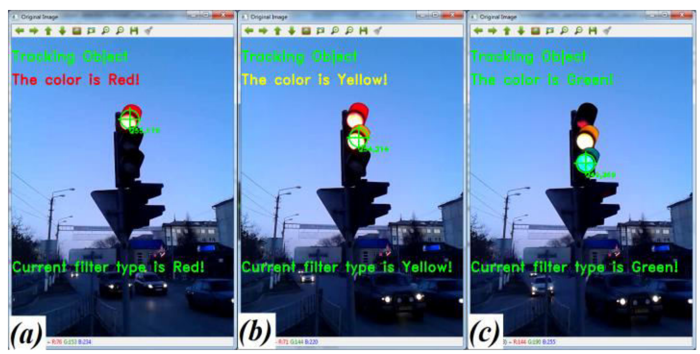

Fig.7. Examples of successful traffic light recognition with red color (a), yellow color (b), and green color (c) signals being displayed and identified

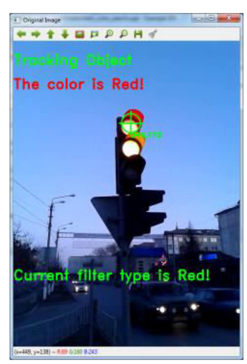

Fig.8. Example of misinterpreting traffic light signal when both red and yellow color signals displayed

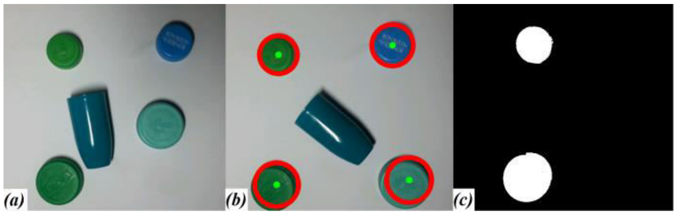

Fig. 9. Android application realization process of testing example samples in RGB color scale (a) for Hough Circles detection (b) and for thresholding green colored objects (c) using Android OS mobile device.

Due to time limitations for accomplishment of this project and low level of required skills in programming on Java programming language, for the first stage we put the main goal to create a working application that will separately implement functions to find circle shaped objects and to threshold video frames. The samples illustrated in Fig. 9 indicate successful accomplishment of this step.

The second part of the program development was to make program compare between these two functions, and to represent an object that satisfies both the circled shape and color of the traffic signal. In order to do this, the prototype version of application was tested on Galaxy S3 mobile phone utilizing Android version 4.3.1. The results of this stage accomplishment are represented in Fig. 10.

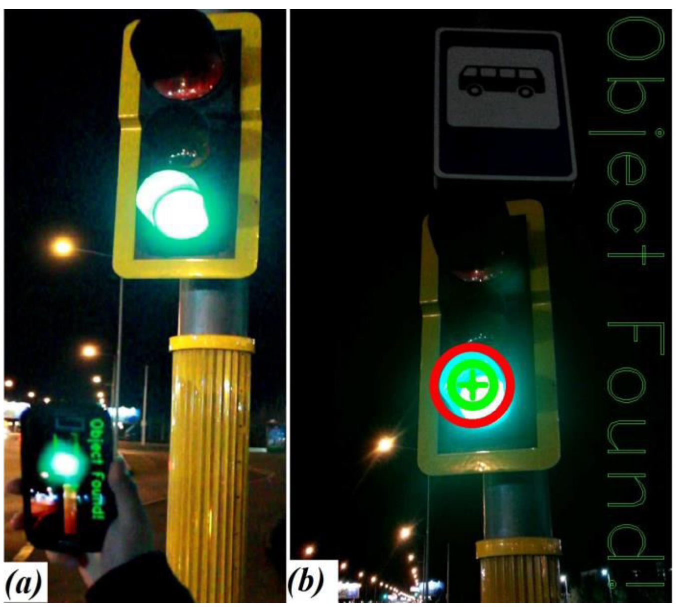

Fig. 10. Photo of Galaxy S3 mobile phone running preliminary version of the application (a) and the screenshot of the device (b) showing the coincidence of circle detection result (red circle) and of green color recognition result (green crosshair) and corresponding message

The further parts of android application realization will include the implementation of automatic color filter changing, and sound notifications for users.

V. Conclusion

This paper has proposed an application design for traffic light recognition for Windows based systems and its prototype for Android based devices. The significance of the following design is that the program was not only utilizing the threshold based object detection principles but also incorporated the shape detection functions. This gave the possibility to eliminate any objects that may present on video frame and have the same color as these of traffic lights. Moreover, this ensures that the coordinates of the detected traffic light become also known.

However, the proposed design still has significant number of imperfections. So, the program was not tested for the case when there is the number of disturbing moving objects. In addition, the color recognition function based on filtering function is not appropriate for various light intensities. Hence, the program may fail to recognize color appropriately during different time of the day.

Hence, future improvements may include encountering time of the day, which may, in turn, be used to determine appropriate HSV values according to light and traffic intensity conditions.

VI. Acknowledgement

Authors would like to express their gratitude to anonymous reviewers for spending their time to give useful comments which contributed to improving the quality of the paper and for raising motivation to accomplish the initial version of the program for android platform.

Special thanks to these people who kindly helped by providing their android devices/cameras for testing and data collection.

VII. References

[1] Abdallah, Mahmoud, and Eiland, Daniel. Traffic signal detection and tracking. Spring 2011.

[2] Baggio, Daniel Lélis. *Mastering OpenCV with practical computer vision projects*. Packt Publishing Ltd, 2012.

[3] Bin Mohd Mokji, Mussa, and Zaim Bin Hamzah, Ahmad. Traffic light signal detection using Windows phone. March 27, 2015.

[4] Bradski, Gary, and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[5] Docs.opencv.org. Finding contours in your image. Last modified February 25, 2015.
http://docs.opencv.org/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html

[6] Docs.opencv.org. Hough cirle transform. Last modified February 25, 2015.
http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html

[7] Docs.opencv.org. Operations on arrays. Last modified February 25, 2015.
http://docs.opencv.org/modules/core/doc/operations_on_arrays.html

[8] Laganière, Robert. OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 recipes to master this library of programming functions for real-time computer vision. Packt Publishing Ltd, 2011.

[9] Lorsakul, Auranuch, and Jackrit Suthakorn. "Traffic sign recognition using neural network on opencv: Toward intelligent vehicle/driver assistance system." In *4th International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 22-24. 2007.

[10] Chung, Yun-Chung, Jung-Ming Wang, and Sei-Wang Chen. "A vision-based traffic light detection system at intersections." *Journal of Taiwan Normal University: Mathematics, Science and Technology* 47, no. 1 (2002):67-8