Multi-Sensor Preprocessing for Traffic Light Detection

BY

NISHAT ANJUM KHAN
B.S., Bangladesh University of Engineering and Technology, 2014

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2017

Chicago, Illinois

Defense Committee:

Dr. Rashid Ansari, Chair and Advisor
Dr. Ahmet Enis Cetin
Dr. Mojtaba Soltanalian

To my parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

abstract goes here.

CHAPTER 1

INTRODUCTION

We used hough circle (1) to detect traffic lights.

..

CHAPTER 2

SYSTEM

In this chapter, we discuss about the system overview for traffic light detection. This system detect the color of traffic light in a recorded videoframe. We captured video using a smartphone along with the sensor data.

2.1 System architecture

We use three features of traffic light, color,shape, and traffic bulb in a black box and the sensor feature of a smartphone as a system architecture for traffic light detection. While recording the video we logged in the sensor data. The first step of detection is the color filtering of the recorded videos. Each video frames is consist of different colors. In this step each frame is filtered with only red and green pixel. The traffic bulb shape is mostly circle. To detect this characteristic we use Hough Circle transformation(1). Based on the traffic light detection position in videoframes, we fix region of interest area, which is a subpart of videoframes. After that, on next videoframes the ROI change in respect to the sensor data.

2.1.1 Diagram

Section 2.1 discesses the overall architecture of our system.

Figure 1 shows the overview of the system.

At first we recorded videos along with the sensor data, gyroscope, accelerometer, pitch,roll, and azimuth. These data have syncronization with our recorded videoframes, since we logged

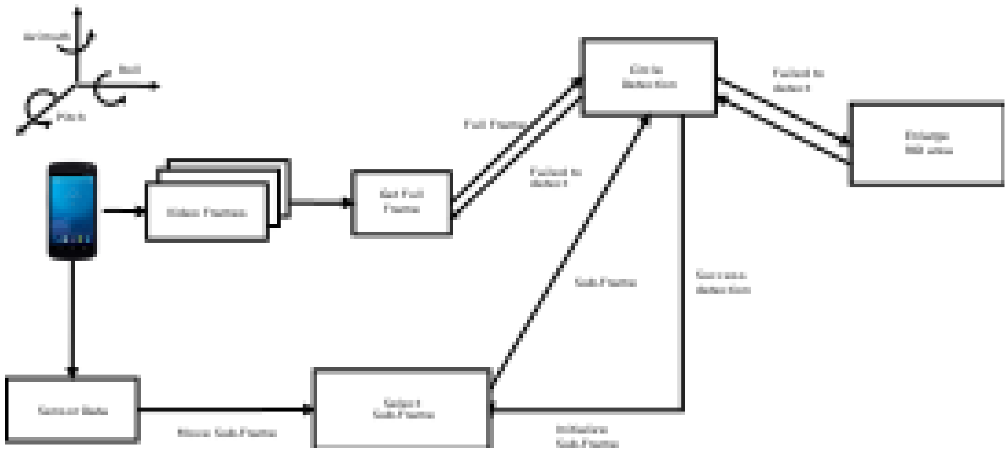

Figure 1: System Overview

in sensor data while recording. Now to detect the traffic light, we use three features of traffic light. Those are the color, shape of traffic light and traffic light location in a black box. For detection using the color and shape features,we need to filter out the frames with the red and green pixel value. For this purpose we convert the BGR color space to HSV space first. In BGR space, to detect specific color we need to depend on three different values (B,G,R) HSV space has the property to find out the color based on only single value,hue. The range of hue values of each color are well defined that help us to filter out our desired pixels. In OpenCV the hue range is from 0 to 180. We fix the range for red and green analysing the color range of traffic bulbs. Table I shows the hue range for red and green pixel.

Before use these filter images to detect circle, we use gaussian blur filter, in order to avoid false detection. After this, we use the circle Hough Transform (1) to detect the circles. But, noise from the original images can fool the hough transform to detect false and more circles.

TABLE I: Hue range for red and green pixel.

Hue range for red	Lower 0 to 10
	Upper 160 to 179
Hue range for green	65 to 95

As a solution to this poblem, before converting our original BGR space frame to HSV space, a median filter is used. Now to detect if traffic light is in the black box, the other feature of our detection, we scan a rectangle frame around the traffic light. Traffic lights locate in a black box, which can place in horizontally or vertically in street. To make our approch more global we scan all around the traffic light to check the intensity. The intensity of black color can detect to check the Value of HSV space. For black color, range of Value is very low, less than 40-45. Figure 2 shows there is one green false positive between two red traffic light. Figure 3 shows the output after using our heuristic black box checking filter.

For improvize our detection we also use the smartphone sensor data. So when we detect a traffic light in our video frame successfully using the color,shape of traffic light and the characteristics of traffic bilb being in a box, we have idea of the position of the traffic light. With this idea we subdevide our videoframe making a region of interest area. When we move to the next frame, we have a prior knowledge of traffic light position and we know the sensor data, pitch, roll and azimuth of this frame. Using these sensor data of curent frame and the previous frame we have the idea of movement of traffic light. With this change of movement the ROI also moves to the direction of change. Figure 4shows the movement of ROI with the

Figure 2: Output not using black box checking filter.

Figure 3: Output using black box checking filter.

change of pitch and azimuth of our recorded video. Apart from this, when our system can not detect any circle in our specific ROI area, it updates the ROI and try to detect the light.

(a) Initial ROI

(b) Movement with change of sensor data

(c) Movement

Figure 4: ROI movement with the change of sensor data.

With the successful detection, our region of interest again change with the light position and pitch and azimuth value. For unsuccessful detection, ROI enlarged until it process the full frame and then go to the next frame to detect traffic light. And this process is contuing to every frame.

Figure 5: Enlarged ROI to detect traffic light successfully.

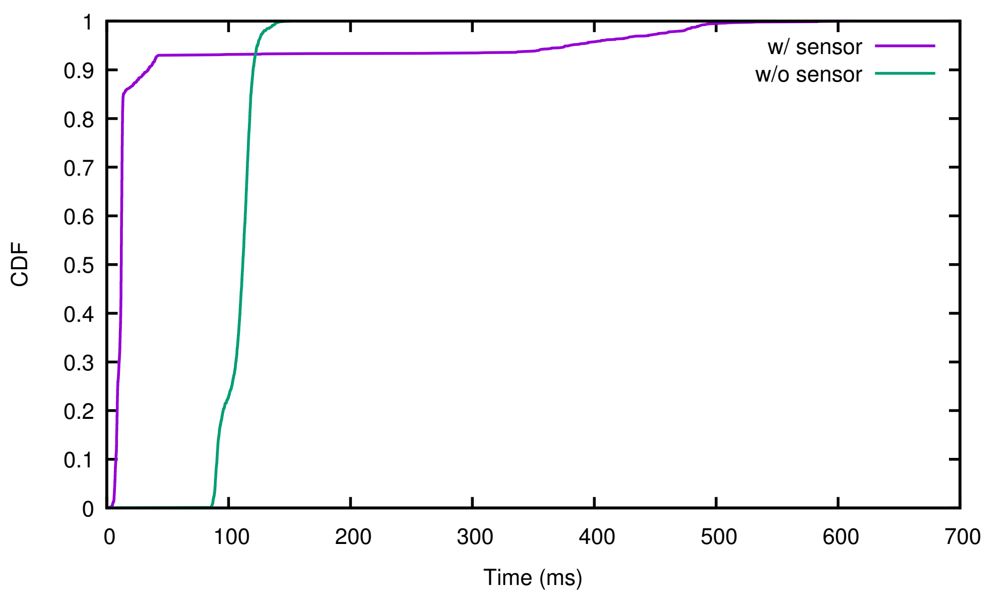

Figure 6: CDF of frame processing time.

CHAPTER 3

CONCLUSION

In conclusion, this is a great work!

CITED LITERATURE

1. Smereka, M. and Duleba, I.: Circular object detection using a modified hough transform. International Journal of Applied Mathematics and Computer Science, 18(1):85, 2008.

VITA

Name	ABM Musa
Education	Ph.D., Computer Science, University of Illinois at Chicago, 2017. B.S., Computer Science and Engineering, Bangladesh University of Engineering and Technology, 2008.
Publications	A.B.M. Musa, James Biagioni and Jakob Eriksson, Trading Off Accuracy, Timeliness, and Uplink Usage in Online GPS Tracking, in IEEE Transactions on Mobile Computing (TMC), vol. 15, no. 8, pp. 2124–2136, Aug. 1 2016. James Biagioni, A.B.M. Musa, and Jakob Eriksson, Thrifty Tracking: Online GPS Tracking with Low Data Uplink Usage, In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL13). ACM, New York, NY, USA, 496–499. A.B.M. Musa and Jakob Eriksson. Tracking unmodified smartphones using Wi-Fi monitors, In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys 12). ACM, New York, NY, USA, 281–294. A.B.M. Musa, Jakob Eriksson, Demo: WiFlow: Real-time Travel Time Estimation using Wi-Fi Monitors, In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys 11). ACM, New York, NY, USA, 429–430. A.B.M. Musa, Jakob Eriksson, On-Demand, Deploy-Anywhere Wi-Fi Diagnostics, Demo In Proceedings of the sixteenth annual international conference on Mobile computing and networking (MobiCom 10), ACM, New York, NY, USA.
Patents	Theodoros Salonidis, A.B.M. Musa, Seraphin B. Calo, Murtaza Jafar, Executing Data Stream Processing Applications in Dynamic Network Environments, International Business Machines Corporation, US20160156514 (2016). Lenin Ravindranath, Victor Bahl, A.B.M. Musa, and David Chu, Application-centric Socialization, Microsoft Corporation, US20140280443 (2014).

Awards	Third place in UIC research forum competition among Engineering and Physical Science (2011)
	Deans List for excellent academic performance in BUET (2004–2007)
	University Merit Scholarship at BUET (2003)
Service	Reviewer for IEEE Transactions on Intelligent Transportation Systems (2016–2017)
	Reviewer for International Journal for Geographical Information Science (2017)
	Treasurer at UIC Computer Science Graduate Student Association (2012–2013)