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,Co-Advisor
Dr. Mojtaba Soltanalian

To my parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

abstract goes here.

CHAPTER 1

INTRODUCTION

This dissertation addresses the problem of efficient traffic light detection using the sensor hint of a smartphones.

Smartphone usage is growing in past years. This growth of smartphones and GPS using provides a great oppertunity on navigation. There is a lots of work on navigation for the automatic car. For the pedestrian navigation, there is very limited work.

For the automatic cars and pedestraian navigation, though we hava GPS data of smartphones, we need to know the environment. Traffic light detection is a very crutial part for this navigation. Traffic light detection only a part of the navigation, so it's important to have lower comutatational time for this. Sensor hint of smartphone can improve the compuutational time and detect the traffic light efficiently.

Our system focus on the detection of traffic light as a part of the navigation for the visually impaired people. The contribution of this work are:

- Our system uses the sensor hint to improve the computation time. We can improve 10x computational time.
- We use heuristic filter to check the traffic bulb in a black box that improve the accuracy.
- We experiment and implement our system in the real time.

The reminder of this dissertation is structed as fllows. Background of the inertial sensor of smartphones and the image processing for traffic light detection is provided in chapter 2. Some previous and ongoing work related to our system is described in chapter . We describe the overall operation of our system in chapter 4 followed by the evaluation result in chapter .

CHAPTER 2

BACKGROUND

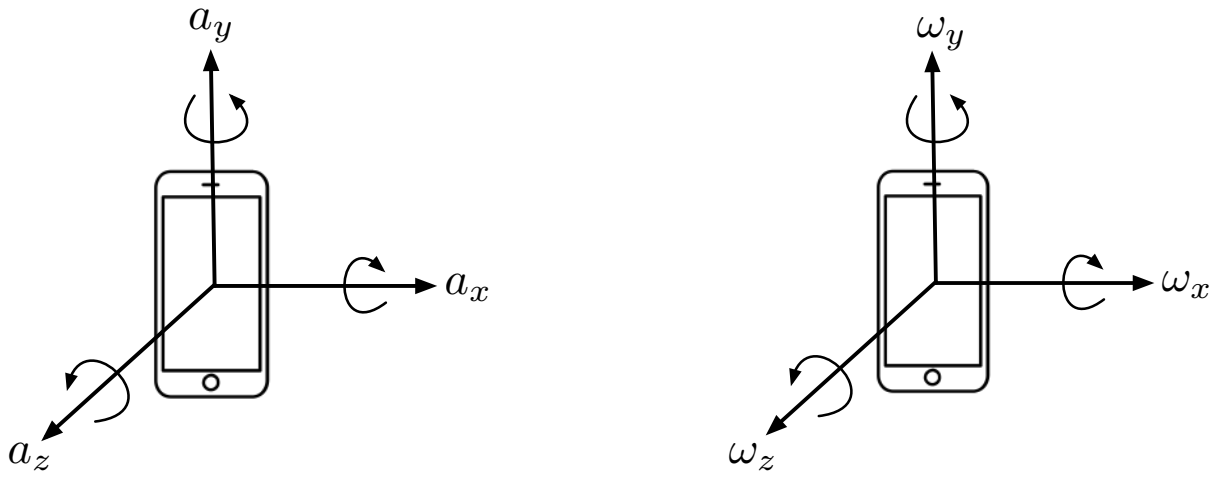
In this chapter we will discuss the background of our system. The main approch of our system is to improve the traffic light detection using smartphone sensor. In first part of this chapter we will discuss the inertial sensor of Android and sensor fusion. Later we will discuss the traffic bulb detection process.

2.1 Inertial sensor of smartphone

2.1.1 Accelerometer

Accelerometer is one of the motion sensor in smartphone. Smartphone now a days have three axis accelerometer,which can measure acceleration along this three axis and return a vector in the reference frame. Accelerometer can not measure free fall acceleration rather it measure the forces that are applied to the sensor itself. For example, when phone is on the table, it will measure acceleration of g, the acceleration due to gravity, because this is proportional to the accelerometer experienced weight.

Figure 1a shows the reference coordination system of accelerometer in Android phone. The device measures acceleration a_x, a_y and a_z along the three axis of android reference frame. The acceleration a_z points towards the outside of screen face, that is perpendicular to the phone plane. The acceleration a_x and a_y are along to phone plane.



(a) Accelerator coordination (b) Gyroscope coordination

Figure 1: Smartphone reference coordination system.

2.1.2 Gyroscope

Gyroscope measure the rate of rotation or the angular velocity of rotation along the three axis of smartphone's reference frame. This is also a motion sensor of smartphone like accelerometer. These gyroscope use MEMS (micro-electro-mechanical sensing) technology, contain vibrating elements to measure Coriolis effect. When smartphone rotates, there is a change of direction of vibrating elements because of the Coriolis force. MEMS gyroscope measure these variation along the three axis to estimate the rate of rotation. Gyroscope output is quite smooth, and very responsive to small rotations.

Figure 1b shows the reference coordination system of gyroscope in Android phone. The device measures rate of rotaion ω_x, ω_y and ω_z along the three axis of android reference frame. The angular velocity of rotation ω_z points towards the outside of screen face, that is perpendicular

to the phone plane, which is the angular velocity of phone's rotaion in X-Y plane. The rate of rotation ω_x and ω_y is along with the phone plane, which is the angular velocity of phone's rotaion in Y-Z plane and Z-X plane respectively.

2.1.3 Geomagnetic field sensor

Geomagnetic field sensor is a position sensor of smartphone. It helps to determine the device's physical position in the world's reference frame. Geomagnetic field sensor mesure the change of magnetioc field and estimates the magnetic field at earth's point to find the declination from the true North. It provides the geomagnetic field strength along the three coordination axis of refernce frame.

2.1.4 Orientation

Each sensor defined in previous section has it's own strength and weakness. Gyroscope is fast, accurate and reliable. It is very responsive to small rotation so it can track the change of rotation at every timestamp. But gyroscope data can not measure gravity. So with accelerometer it can provide a better rotation or motion of the sensor in application's frame reference. But to have the position of the device in world's reference frame, we need to fused this data with geomagnetic field sensor. Another motion sensor in android, rotation vector, use this three sensor data to report the orientation of the device in vector form along the axis of reference frame.

This rotation vector provides the axis of the system and the angle of rotation from these axis. From this angle and axis data, we can get the orientation of the system in terms of pitch, roll and azimuth.

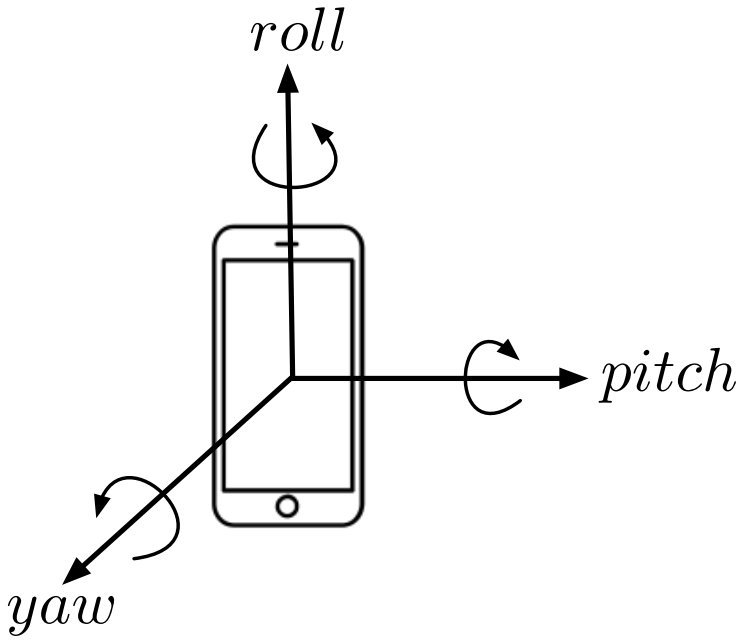


Figure 2: Orientation of the smartphone

Figure 2 shows the orientation of a smartphone about the device's coordination system. Pitch is the degree of rotation about the X axis. This is the angle between the plane parallel to device and parallel to the ground. If device top edge incle to the ground the pitch will be positive and vice versa. The range of pitch value is -180 degrees to 180 degrees. Roll is the degree of rotation abot Y axix. This is the angle between the plane perpendicular to device and perpendicular to ground. If device's left edge inclenes to the ground roll value is positive. The range of roll value is -90 degrees to 90 degrees. Azimuth is the degree of rotation about Z axis. If the device is faced to the earth's North the azimuth is 0 degree. Azimuth is 90 degree at earth's East, 180 degree to earth's South and 270 degree to earth's West.

2.2 Image Processing

2.2.1 Color Space

The main two color spaces for processing the color vision are RGB color space and HSV color space. RGB is a additive color space. RGB color space describes colors with the amount of red, green and blue color presents on that frame.

Figure 3 shows the model of RGB space. It shows that any color of this cube is made of red,blue, and green color.

HSV color space is similar to the way human sees the color. But RGB defines color in relation to the primary colors. HSV color space describes color in terms of Hue, Saturation and Value. It separates the color information from the intensity or the brightness. Hue is the continuous representation of color type. In HSV, hue is an angle from 0 to 360 degree. But since OpenCv can process data from 0 to 255, so in OpenCv the range is halved. Every color can be discrimated with the different range of Hue. Saturation represents the vibracy of the color. The ranges for saturation is 0 to 255. The lower saturation meaning gray is present in the color, higher saturation represents primary color. Value represents brightness of the color. The range of value is same as saturation, 0 to 255. Lower value means less bright color, 0 brightness represents black color.

Figure 4 shows the cylindrical model of HSV color space. The Hue is the circular part. The radious of this circular part represent Saturation and the height is the Value.

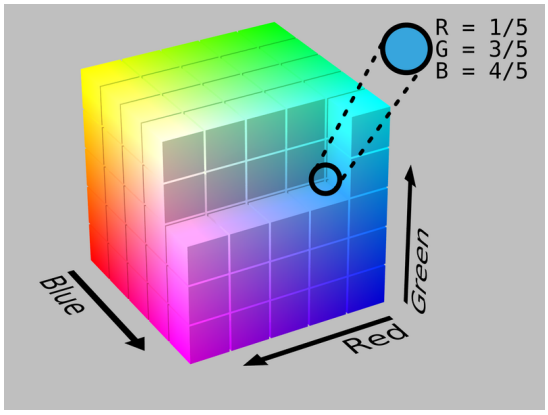


Figure 3: RGB color space

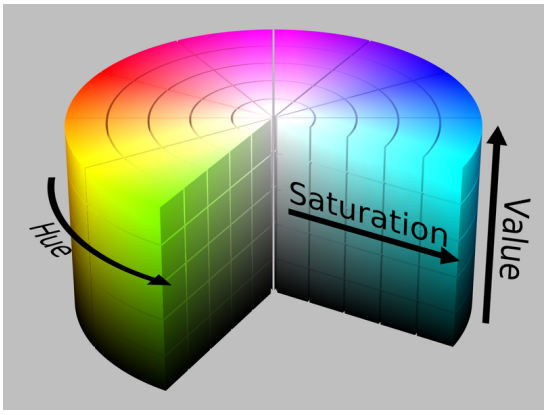


Figure 4: HSV color space

2.2.2 Hough Circle Transform

Our system detects the traffic bulb using the hough circle algorithm (1). To detect the circle first process is to find the edges in the input image. The hough circle algotithm uses canny operation for the edge detection. At every edge pixel, it generates a circle. An accumulator matrix is created to find the intersection of these circles. If circle passes through the grid of the accumulator matrix, it increase the coting number of the grid. The psition of the local maxima of this accumulator matrix provides the corresponding circles centers.

CHAPTER 3

RELATED WORK

??

CHAPTER 4

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.

4.1 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(2). 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.

4.1.1 Diagram

Section 4.1 discesses the overall architecture of our system.

Figure 5 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

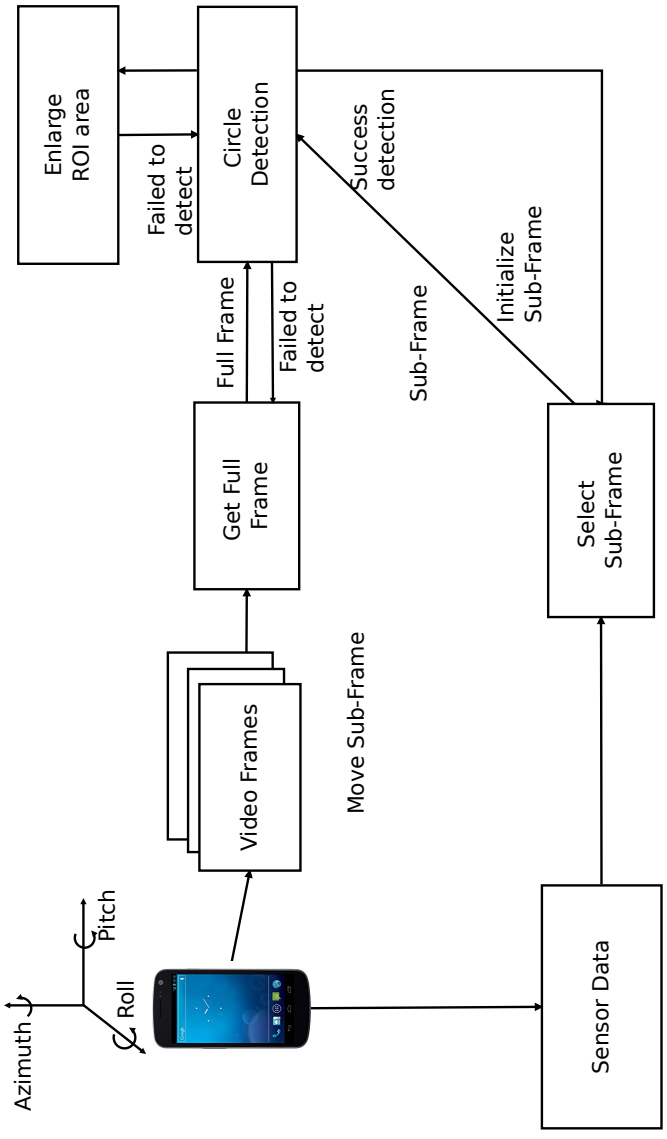


Figure 5: System Overview

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.

4.1.2 Color filtering

The first step of detection algorithm is color filtering of videoframes. The most distinctive feature of traffic bulb is it's bright color. Color filtering step filter out these candidate pixel from the videoframes. 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 analyzing the color range of traffic bulbs. Table I shows the hue range for red and green pixel.

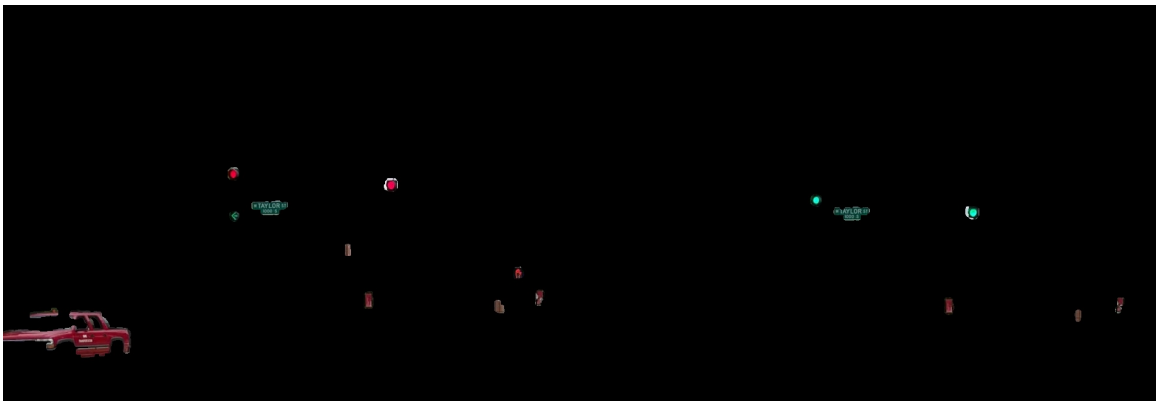
TABLE I: Hue range for red and green pixel.

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

Figure 6(a) shows the videoframe with red traffic signal light and (b) shows the frame with green traffic signal light. Figure 7 shows the image with only red and green pixel. (a) shows the filtering output with red traffic light and (b) shows the fltering output with green traffic

(a) Frame with red lights (b) Frame with green lights

Figure 6: Original videoframes.



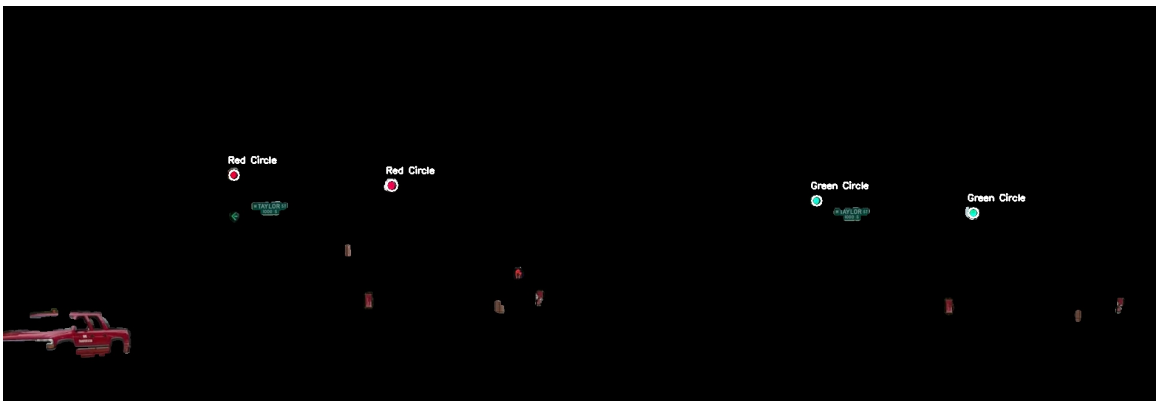
(a) Frame with red lights (b) Frame with green lights

Figure 7: Red green pixel filtering.

lights. The color filtering process is computationally lightweighted and it zeros out most of the pixels that reduce comutational time to next steps.

4.1.3 Circularity check

The next step of this detection algorithm is to detect the shape of traffic bulb (circle). These filter images has only the desired pixel values. To detect circle on this filtered images, we use gaussian blur filter, in order to avoid false detection. After this, we use the circle Hough Transform (2) to detect the circles. But, noise from the original images can fool the hough transform to detect false and more circles. As a solution to this poblem, before converting our original BGR space frame to HSV space, a median filter is used.



(a) Frame with red lights (b) Frame with green lights

Figure 8: Red green traffic bulb detection.

Figure 8 shows the circle detection output. (a) shows the red traffic light detection. In this image we have other red pixel values except the traffic signal light. But Hough circle transform

can detect the circle precisely. (b) shows the green traffic light detection output with Hough transform method.

4.1.4 Heuristic filter

At this point we have knoledge about the traffic bulb location and size (radius). Now to detect if traffic bulb is in the black box, the other feature of our detection, we tried two heuristic filters. Our first approach is to check pixels intensity of a circular area around the traffic bulb. We use midpoint circle algorithm to find out the pixels value on a circle perimeter which is larger than our traffic bulb size. our system give a matrix with the count of pixels which has the intensity of a black color of all those points. 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. If this count is greater than our threshold we select that circle as our traffic bulb. But we face some false detection problem with this approch. Since the intensity of traffic bulb color is high and it spreads out also. So we can not get successful result with this approch.

Figure 9 shows the red and green traffic bulb intensity. We can see that the color spreads out around the exact circle area. So the black color intensity measurement around the bulb circle area using midpoint circle algorithm is not giving actual result.

Our second approch is to scan a rectangle frame around the traffic light. Traffic bulbss 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. Our system give a matrix of the no of pixels that has the intensity same as black color. If this pixel no is more

(a) Red traffic bulb (b) Green traffic bulb

Figure 9: Red and green traffic bulb intensity.

than our threshold value we consider that the detected circle is in the traffic signal black box and we finally detect that circle as our traffic bulb.

Figure 10 shows there is one green false positive between two red traffic light. That false green circle actually a street nameplate and around that there is no black box. So when we use our black box checing filter we can remove this false positive detection. Figure 11 shows the output after using our heuristic black box checking filter.

4.1.5 Sensor hints

4.1.5.1 Syncronization sensor data

To improve our detection we also use the smartphone's sensor data hints. Our system logged the sensor data while recording the video. From the logged data we know the time for sensor data registered. We have the time when video is recorded. After that, we synchronized our data

Figure 10: Output not using black box checking filter.

with the video using the starting time of the video and the frame rate. So, using the frame rate and starting time we know the time for each frame, and from the logged sensor data, we know the sensor value correspond to the time. Using this time and our frame time we interpolated sensor hints if we miss any data correspond to that frame. Finally, we get the synchronized sensor data with our recorded video.

?? shows the interface of our android app. So when we start recording the pitch, roll and azimuth display in the screen and also start registering in a file.

4.1.5.2 Region of interest area selection

Now, 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

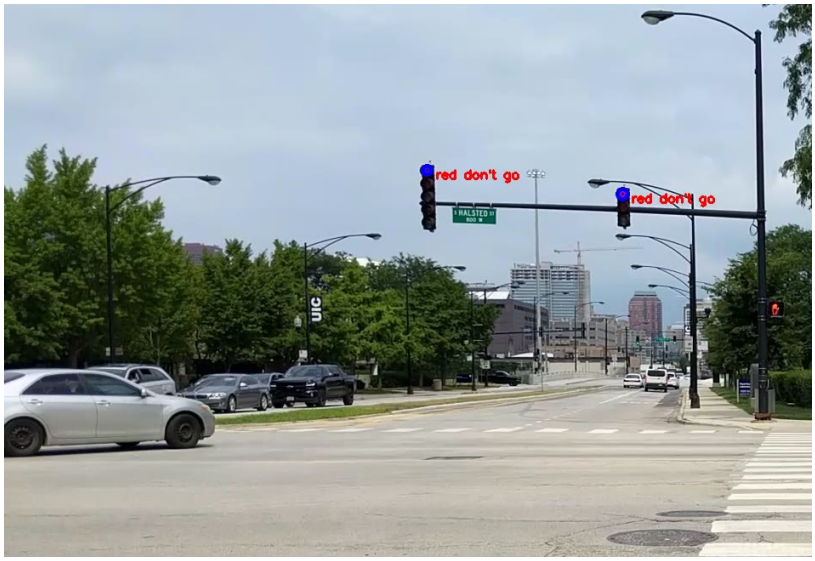


Figure 11: Output using black box checking filter.

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 12shows the movement of ROI with the 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.

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.

(a) Initial ROI

(b) Movement with change of sensor data

(c) Movement with change of sensor data

Figure 12: ROI movement with the change of sensor data.

Figure 13: Enlarged ROI to detect traffic light successfully.

CHAPTER 5

EVALUATION

This chapter discuss the n-to-n evaluation of our system.

5.1 Evaluation dataset

For collecting the ground truth data we walked along the path of 300 feet at an urban street to record the video with samrtphone. We record data at different time of the day and at different light condition. Figure 14 shows the scene variation of recored video. The right one is cloudy and the left one is sunny.

(a) Sunny (b) Cloudy

Figure 14: Scene variation of recorded video.

Table II shows the total no of frames, and time duration of our dataset.

TABLE II: Description of the dataset.

Name	Frame Number	Time Duration
Sunny day	5905	3 min 16 sec
Cloudy day	6205	3 min 26 sec

5.2 Computation time

We collected data at different time of the day as we discussed in 5.1. In this section we discuss the processing time of these dataset with the sensor hint and without it.

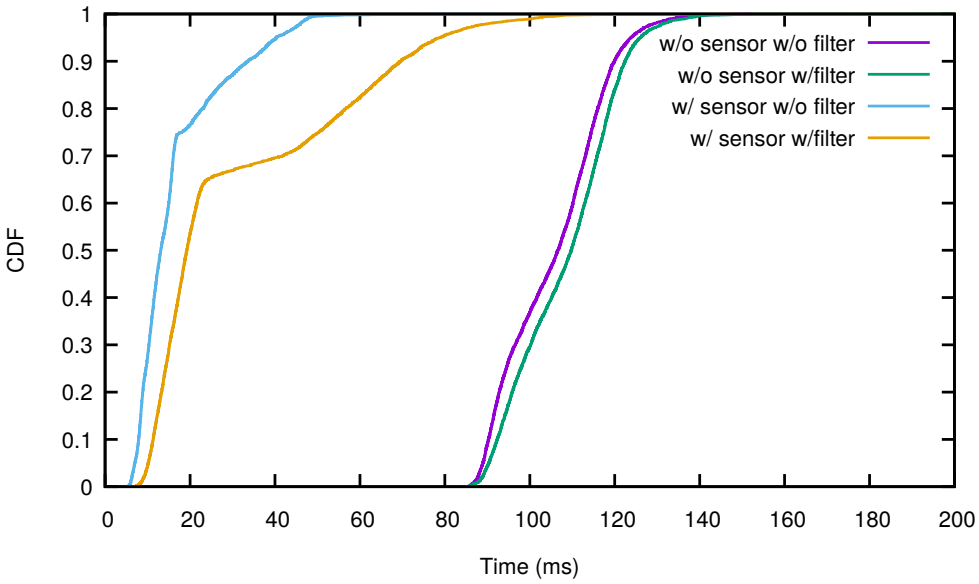


Figure 15: CDF of frame processing time.

Figure 15 shows the CDF of the processing time. It shows that the median of the processing time without sensor and any heuristic filter is 106ms. Wheather, the median of the processing time with sensor and without heuristic filter is 13ms. We can improve the processing time 8.15x. The median of the processing time witt sensor and heuristic filter is 19ms. Though we can improve the processing time 5.57x with the filter but it increases the accuracy, which we will describe at section 5.3

5.3 Accuracy

CHAPTER 6

CONCLUSION

In conclusion, this is a great work!

CITED LITERATURE

1. Duda, R. O. and Hart, P. E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM, 15(1):11–15, January 1972.

2. 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.