TRAFFIC DENSITY BASED SIGNAL DURATION MODULATION

Sushanth Chintalapati[1], Shashank Vishnu Conjeevaram[2], Arshad Shareef Shaik[3], Nazeer Unnisa[4]

[1]Student (ECE), Muffakham Jah College of Engineering and Technology, Hyderabad, India
[2]Student (ECE), Muffakham Jah College of Engineering and Technology, Hyderabad, India
[3]Student (ECE), Muffakham Jah College of Engineering and Technology, Hyderabad, India
[4] Sr. Assistant Professor (ECE), Muffakham Jah College of Engineering and Technology, Hyderabad, India

Abstract

Traffic is one of the major problems that have affected almost all the countries across the globe. As this problem of urban traffic congestion is constantly on the rise, there is an urgent need to introduce latest technologies and equipment to control the traffic. This sudden increase in a number of vehicles resulted in the need for a smart system that can efficiently handle traffic congestion based on the density of traffic. The fact is that the population of city and numbers of vehicles on the road are increasing day by day. The main reason behind today's traffic problem is the techniques that are used for traffic control. Today's traffic management system gives no emphasis on live traffic scenario, which leads to inefficient traffic management systems. This project has been implemented by using the Mat-lab 2014b software and it aims to prevent heavy traffic congestion. Moreover, for implementing this project Image processing technique is used. At first, an image of an empty road is captured and then this image is used as a reference image for calculating the traffic density based on image correlation. Then these images are efficiently processed based on edge detection techniques to know the traffic density. The correlation values are then exported on to an excel sheet for faster computations.

Index Terms: Traffic Density, Mat lab, Edge Detection, Image Processing, Correlation, Excel

I. INTRODUCTION

In this paper, our main focus is on reducing the traffic congestion at signal junctions. One of the reasons for such an enormous increase in traffic is the ever-increasing population and the development of the economy. The high volume of vehicles, the dodgy infrastructure and the non-uniform distribution of development are the major reasons constituting for a long traffic jam. To overcome this, the government should encourage its people to use public transport or vehicles of a smaller size such as bicycles. Most countries have imposed a law restricting the number of automobiles a particular family can have. Such methods can be effective but cannot be implemented for longer durations of time. The public transports are available and they are not properly maintained mostly in the establishing countries. Besides, the highway and roads are incapable of meeting the requirement of increasing number of vehicles.

There are several techniques which have been proposed over the years for tackling the issue of traffic congestion [1]. One method is to manually control the traffic with the help of traffic personnel. This can be an effective method but not an efficient one. Another way of tackling this issue is by using automated traffic signals. This method can be a very efficient one if the signal durations can be computed in a very short amount of time.

In this paper, we make use of the image processing techniques to find the correlation values, based on which we can calculate the

signal durations [2]. The correlation values are calculated by comparing an image of a road which has some amount of traffic to a reference image which is an empty road. Several databases have been created for the purpose of calculating the correlation values.

The correlation values have been calculated for various different scenarios at different time instants. All the computed correlation values are then copied to an excel sheet where these values are stored and its mean is calculated. By doing this, the computation time reduces drastically and this value can be used to calculate the signal duration. We have decided to use the Mat Lab Simulation Software 2014b for implementing and executing the image processing techniques along with calculation of correlation values and integrating the same into excel sheets.

II. Operation of the proposed model

Many techniques have been developed in Image Processing in the recent years. Most of the methods developed are for enhancing images and getting more information from them than previously used methods. Image Processing systems are very popular and easy to use nowadays due to the high-performance personal computers and memory devices capable of storing large data, graphics software and many more. A few of the operations that come under image processing are image enhancement operations such as sharpening, blurring, brightening, edge enhancement etc. Traffic density of lanes is calculated using image processing. Making use of the above-mentioned features of image processing we propose a technique that can be used for traffic control. The proposed model is as shown in **Figure 1**, the reference image and the target image are first converted into grayscale images and then image enhancement techniques are used to improve the quality of the image and then edge detection is used. Finally, we compare the results of the reference image and the captured image using correlation and based on this the signal duration is computed.

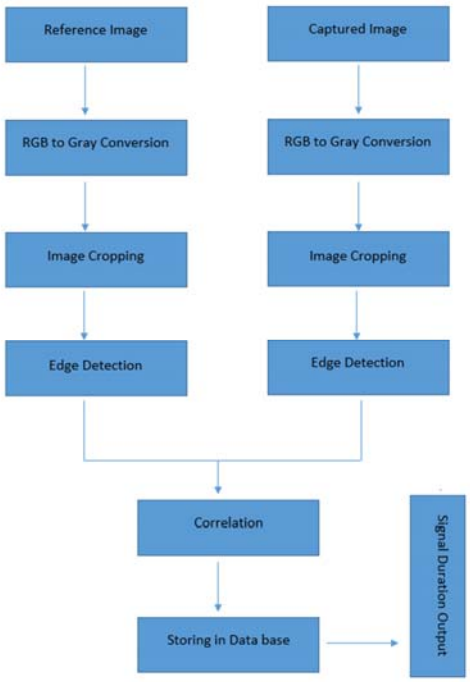


Figure 1: Block diagram representing the process of correlation and signal duration calculation

II A Image Acquisition

An image is a two-dimensional function f(x, y). The amplitude of image at any point say f is called intensity of the image. It is also called the gray level of the image at that point. Each digital image is a combination of finite elements and each finite element is called a pixel. Pixels are the basic elements of any image. Pixels give us an idea of what exactly must be done to get a better result than the previous results **[3].** The size of the image being used in our project is 480 x 720.

II B RGB to Gray Conversions

We, humans, differentiate colors with the help of wavelength-sensitive sensory cells called cones. There are three different varieties of cones; each shows different sensitivities to electromagnetic radiation (light) of a different wavelength. One cone is mainly sensitive to green light, one to the red light, and the other to blue light. This is the reason, why color images are often stored as three different image matrices placed one on another; one storing the amount of blue in each pixel, one the amount of green and one the amount of red. We call such color images as stored in an RGB format. In grayscale images, we don't differentiate the emission from the three cones of different colors; it emits the same amount in every channel. Here the whole image

information lies in a single plane. Where the lowest value, zero, corresponds to black and the highest value, corresponds to white. The intermediate values are called gray values. When converting an RGB image into a grayscale image, we should combine all the three-pixel values of an RGB image (Red, Green and, Blue) into just one-pixel value, I.e. we compress three planes into one plane. One of the approaches is to take the average of the contribution from each channel: (R+B+C)/3 and the other method is the weighted sum method. We used the weighted sum wherein red color wavelength was given the smallest value whereas green color wavelength the highest and blue color wavelength was in between green and red colors. Example: *0.3R + 0.59G + 0.11B.* The sole purpose of implementing RGB to Gray conversion is to increase the computational speed of the program and the process significantly as RGB image has 3 planes while the gray image has just one plane.

II C Image Cropping

Image cropping is done to specify the region on the road which is the area of concern for that traffic light. For example, in a two-way road, one in which vehicles are coming towards the traffic light and in other, vehicles going away from the traffic light. Here the vehicles coming towards the traffic light influence the traffic duration, so the image is cropped or selected to the region in which vehicles are coming towards the traffic light, eliminating vehicles going away from it. This is an internal operation and so is not visualized externally.

II D Edge Detection

One of the most important aspects of an image is its edges. Edges are said to contain the most important information regarding the image. This information would be of great help for us to create an algorithm to detect the vehicles on the roads. Edge detection is the name given to a set of mathematical methods which are used for identifying points in a digital image at which the image brightness changes sharply or, has discontinuities or noise. The points at which image brightness changes sharply are generally organized into a set of curved line segments termed edges.

There are several edge detection techniques which are inbuilt functions in Mat Lab. A few of the edge detection techniques used in Mat Lab are:

- Sobel Edge Detection
- Prewitt Edge Detection
- Roberts Edge Detection
- Canny Edge Detection

From the above-mentioned edge detection techniques, we have used **Canny Edge Detection** because it has a faster computation time and the effect of noise was minimal when compared to the other edge detection techniques [4].

II E Canny Edge Detection

The aim of this method is to develop an algorithm that is optimal with respect to the following
Criterions:

Detection: The probability of spotting real edge pixels/points should be maximized while the error-prone probability of detecting non-edge pixels/points should be minimized. This results in the maximizing the signal-to-noise ratio and improving the overall efficiency of edge detection.

Localization: The deviation of detected edges from the actual edges should be as less as possible. This implies that the detected edges should be very close to the actual edges.

The number of responses: Duplicability of real edges should not occur. This means the number of edges obtained should be close to the number of edges present.

The algorithm is of five steps and is as follows:

1. **Smoothing**: The image is initially blurred to remove noise.
2. **Finding gradients**: Where ever the gradients are high in magnitude the edges are marked there.
3. **Non-maximum suppression**: Off all the pixels/ points only the local maxima should be marked as an edge.
4. **Double thresholding**: Thresholding is used to determine possible edge locations.
5. **Edge tracking by hysteresis**: The edges that are not connected to a very strong edge and are scattered are removed by suppressing them.

1. Smoothing*:* It is not possible to capture an image without noise at all, and so contain some noise. To prevent the noise from disturbing the process and mistaken as an edge, it must be reduced. Therefore the image is first smoothed by applying a Gaussian filter, which eliminates a portion of the noise. The kernel of a Gaussian filter with a standard deviation of $\sigma = 1.4$ is shown in the equation given below

$$G = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2. Finding gradients: Gradients at each pixel in the smoothed image are determined by applying what is known as the Sobel-operator. Initially, we approximate the gradient in both x and y direction ($\mathbf{k_{GX}}$,$\mathbf{k_{GY}}$)**.**

$$\boldsymbol{k_{GX}} = \begin{bmatrix} \mathbf{-1} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{0} & \mathbf{2} \\ \mathbf{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{k_{GY}} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{-1} & \mathbf{-2} & \mathbf{-1} \end{bmatrix}$$

$$\boldsymbol{G} = \sqrt{\boldsymbol{G_x^2} + \boldsymbol{G_y^2}}$$

$G_x \; and \; G_y$ Are gradients in x and y direction.

$\theta = tan^{-1}\left(\frac{G_y}{G_x}\right)$ Θ stands for gradient direction.

3. Non Maximum Suppression*:* This step converts blurred edges obtained from the previous steps into sharp edges.

The algorithm is for and impacts each pixel in the gradient image:
1. Rounding off the gradient direction to nearest 45◦, corresponding to the use of an 8-connected neighborhood.
2. Compare the edge strength of pixel under test with that of the pixels in the positive and negative gradient direction. I.e. if the gradient direction is north (Theta = 90◦), compared with the pixels to the north and south.
3. If the edge strength of the current pixel is largest among the other pixels then preserve the value of the edge strength. If not, suppress (i.e. remove) the value.

4. Double Thresholding: The pixels (edges) remaining after the non-maximum suppression (removal) step is marked with their strength pixel-by-pixel. Even though the probability of most of these to be actual edges, some might occur because of noise or color variations for instance due to rough surfaces. The simplest way to differentiate between these would be to use the concept of thresholding so that only edges stronger than a certain value would be preserved and the others would be discarded. Canny edge detection algorithm uses two thresholds, one high threshold, and one low threshold. Edge pixels that are stronger than the high threshold are marked as strong, edge pixels weaker than the low threshold are suppressed, and edge pixels between the two thresholds (high and low) are marked as weak.

5. Edge tracking by hysteresis: Strong edges are interpreted as most probable edges, and can immediately be included in the final edge image. While weak edges are included only if they are connected or are close to strong edges, this increase the efficiency of the algorithm. The logic is that, with proper adjustment of the threshold levels, noise and other small variations are unlikely to result in a strong edge. Thus strong edges will be mainly due to true edges in the original image. The weak edges can either be present due to noise/color variations or due to true edges. Weak edges due to true edges are much more likely to be connected directly to strong edges, while the others are more likely to be distributed as they occur due to noise.

II F Image Correlation

Correlation is a process in which a reference image is compared with another image, whose result has to be computed. In simple terms, correlation of images is the similarity between two images. For the purpose of correlation, we used an inbuilt command in Mat Lab

X=corr2 (a, b);

The above command returns the correlation coefficient X between A and B, where a and b are matrices or vectors of the same size. r is a scalar double.
The corr2 computes correlation using:

$$X = \frac{\sum_m \sum_n (A_{mn} - \bar{A}) * (B_{mn} - \bar{B})}{\sqrt{((\sum_m \sum_n (A_{mn} - \bar{A})^2) * (\sum_m \sum_n (B_{mn} - \bar{B})^2))}}$$

Where $\bar{A}$ = mean of *A*, and $\bar{B}$ = mean of *B*.

These correlation values help us decide the signal duration based on the traffic densities. For calculating the correlation values we had created several databases at different instants of time. Each database on an average had about fifteen-twenty images of traffic at a junction and also had a reference image of an empty or nearly empty road for the purpose of calculating the correlations.

II.G Creating GUI (Graphic User Interface)

A graphical user interface (GUI) is a graphical display in one or more windows that make it simple and easy for the user to perform or understand a task or process. The user of the GUI need not create a script or type commands at the command line or run scripts to accomplish tasks. Unlike coding programs to accomplish tasks, the user of a GUI does not require understanding the details of how the tasks are being performed. GUIs that are created by using MATLAB tools can also perform any type of computation like read and write data files, communicate with other GUIs, and display data as tables or as graphical plots.

In Mat Lab, GUI's can be built in two ways

- Using GUI Development.
- Creating Code files for generating GUI's.

II H Creating Excel Sheets

The purpose of using an excel sheet is to dump in the correlation values generated from the traffic images and then use the same data from the excel sheets for calculating the mean of the so obtained correlation values which can be further used for calculating the signal duration at that instant of time. The sole purpose of creating and using Databases is to reduce the computational time exponentially. An average of 30-40 seconds is required for completing the flow till correlation values. But, to calculate mean it just takes 0.765 seconds. This reduces the time taken by **97.8142%.**

And this is the only point where this paper differs from other papers proposed for traffic signal modulation, significantly as we implemented the concepts of databases to store digits that add meaning to the traffic rather than storing images which improved the efficiency and computing time of the project exponentially. From 30-40 seconds as per the image processing to 0.725 seconds with mean calculation. This even reduces the cost of hardware and processing near the traffic signal and increase the scope of research to cloud computing and big data.

II I Calculation of Signal Duration

After calculating the correlation mean, the answer obtained is used to drive the signal. A Mat Lab code was generated through which we were able to process this correlation mean. We already had pre-defined threshold values. Whenever the correlation mean fell into a particular pre-defined threshold value we had the signal duration set to a constant value. The results of this will be clearly explained in the results section.

III. SIMULATION RESULTS

1. Image Acquisition

Image Acquisition is the very first step which has to be implemented. Without an image, it would be impossible for us to proceed to the later stages. The image can be acquired using the basic Mat Lab commands and built-in functions. The result of Image Acquisition can be seen in **Figure 2**. This figure represents the captured image of traffic at a signal junction. This image is then converted to a grayscale image and further Canny edge detection technique is used. This image is then compared with an image which is a traffic-free image or an empty road.

Figure 2: Image Acquisition (Traffic Monitoring in Seoul)

2. RGB to Gray scale Conversions

Figure 3 represents the basic RGB to binary and Gray Scale conversions. These conversions of a color image can be made using the built-in commands provided in Mat Lab.

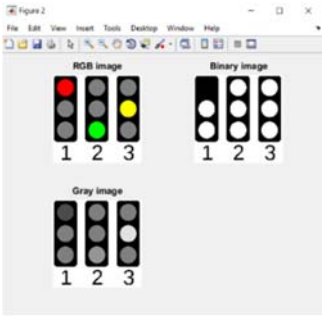


Figure 3: RGB to Binary and Gray conversions

3. Edge Detection

Figure 4 represents the execution results of using various in-built edge detection methods such as Sobel Edge Detection, Prewitt Edge Detection, and Roberts Edge Detection.

For greater accuracy and precision we used the Canny Edge Detection method for the purpose of edge detection so as to calculate the correlation between the captured image and the reference image.

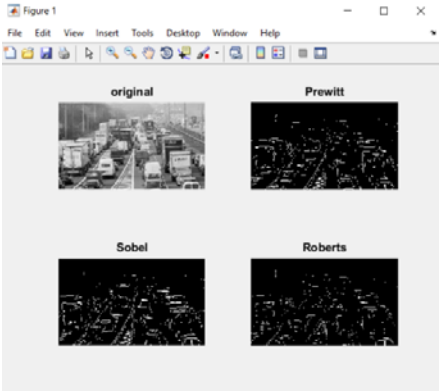


Figure 4: Edge detection (Prewitt, Sobel, Roberts)

4. Canny Edge Detection

Figure 5 represents the use of two different algorithms for Canny Edge detection. The output obtained using the built-in function was not desirable. The effect of noise on the output is greater and this affects the correlation values to a greater extent.

To account for the effect of noise and to suppress it, we use a slightly different algorithm for Canny Edge Detection. The built-in command only has a single threshold value whereas the modified algorithm has a double threshold and this suppresses the noise to a greater extent. This output is much better when compared with the original output and can be effectively used for the purpose of correlation [8].

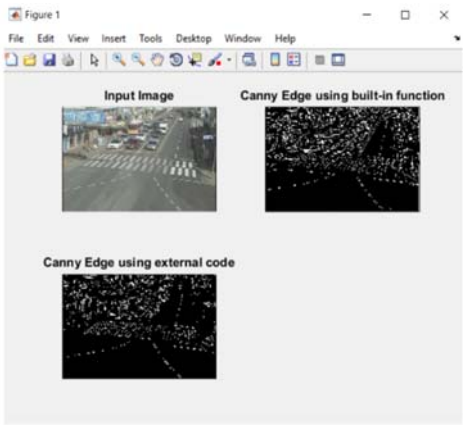


Figure 5: Canny Edge detection

6. Graphical User Interface (GUI's)

Figure 6 represents the Graphical User Interface. The main purpose of creating a GUI is to provide a user-friendly experience in an effective way. No doubt the results will be obtained without the GUI, but with them, it becomes much easier for everyone to understand and interpret what exactly is going on in the code. We load the reference image as image 1 and the captured image as image 2. Then hit the compute button to measure the correlation between the two images. The entire code for the process of correlation is already dumped in the GUI. Finally, the correlation value and the traffic signal duration is displayed basing on the traffic density at that moment.

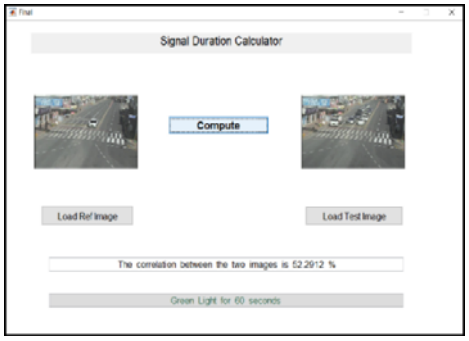
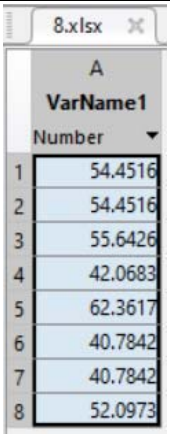


Figure 6: Final GUI output

7. Excel Database Creation

Figure 7 represents the automatic entry of the correlation values of the captured image with the reference image. Using a Mat Lab code all the correlation values that are computed are dumped into an Excel sheet row-wise.

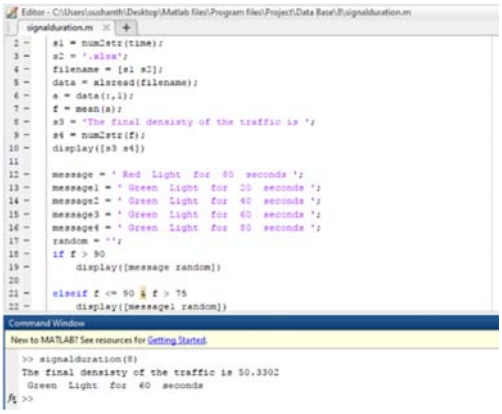


	A
	VarName1
	Number
1	54.4516
2	54.4516
3	55.6426
4	42.0683
5	62.3617
6	40.7842
7	40.7842
8	52.0973

Figure 7: Excel database

8. Signal Duration Calculation

Figure 8 represents the image of the Mat Lab code and the output of the signal duration **[7].**



```
s1 = num2str(time);
s2 = '.xlsx';
filename = [s1 s2];
data = xlsread(filename);
a = data(:,1);
f = mean(a);
s3 = 'The final denesity of the traffic is ';
s4 = num2str(f);
display([s3 s4])

message = ' Red  Light  for  80  seconds ';
message1 = ' Green  Light  for  20  seconds ';
message2 = ' Green  Light  for  40  seconds ';
message3 = ' Green  Light  for  60  seconds ';
message4 = ' Green  Light  for  80  seconds ';
random = '';
if f > 90
    display([message random])

elseif f <= 90 & f > 75
    display([message1 random])
```

```
>> signalduration(8)
The final denesity of the traffic is 50.3302
 Green  Light  for  40  seconds
```

Figure 8: Signal duration calculator

The results are classified as follows

If correlation values are **above 90**, then the result displayed is **Red Light for 80 seconds.**

If correlation values are in **between 75-90**, then the result is displayed is **Green Light for 20 seconds.**

If correlation values are in **between 55-75**, then the result displayed is **Green Light for 40 seconds.**

If the correlation values are in **between 35-65**, then the result displayed is **Green Light for 60 seconds.**

If the correlation values are **less than 35**, then the result displayed is **Green Light for 80 seconds.**

IV Conclusion and Future Scope

This paper emphasizes on the technique of calculating the traffic density at a junction using Image Processing techniques and interpreting the results using GUI. Due to the use of excel sheets, there is a significant improvement in the computation time of signal duration.

The proposed paper can still be improved and upgraded by connecting to concepts like the cloud for faster computations, automation by use of macros or teaching the system how to compute automatically with neural intelligence. It can also be made significantly efficient and powerful by using concepts of data sciences as it was observed that some correlation values did not add much meaning to the collected data and could be eliminated. So with the use of the above concepts the proposed concept of making traffic lights efficient would be industry compatible and very smart.

V References

1. David Beymer, Philip McLauchlan, Benn Coifman, and Jiten-dra Malik, "A real-time computer vision system for measuring traffic parameters," IEEE Conf. on Computer Vision and Pat-tern Recognition, pp495 -501, 1997.

2. Vikramaditya Dangi, Amol Parab, Kshitij Pawar & S.S Rathod, "Image Processing Based Intelligent Traffic Control-ler", Undergraduate Academic Research Journal (UARJ), Vol.1, Issue 1, 2012

3. Digital Image Processing by Rafael C Gonzalez (Text Book)

4. Arvind B.K And Dinesh S, Arun Karthik S And Ganga Am-brish "Traffic Gridlock Control Using Canny Algorithm Aided By Fuzzy Logic"3rd International Conference On Advanced In Computing And Emerging Elearning Technologies (ICAC2ET 2013) – Singapore On November 6 – 7, 2013.

5. Sabya sanchi kanojia, "Real –time Traffic light control and Congestion avoidance system", International Journal of Engineering Research and Applications (IJERA), pp.925-929, Vol. 2, Issue 2, Mar-Apr 2012.

6. Anthony J. Venables, "Evaluating Urban Transport Improvements", Journal of Transport Economics and Policy, Vol. 41, No.2, pp. 173-188, May 2007.

7. Indian Journal of Computer Science and Engineering (IJCSE) (International Journal of Engineering, Basic sciences, Management & Social studies Volume 1, Issue 1, May 2017)

8. Live Video Tracking for Traffic Monitoring in Seoul available at 1.250.137.142/mjpg/video.mjpg.