International Journal of Inventions in Electronics & Electrical Engineering http://www.ijieee.in

(IJIEEE) 2016, Vol. No. 2, Jan-Dec **e-ISSN: 2454-9592; p-ISSN: 2454-8081**

TRAFFIC LIGHT CONTROL SYSTEM - A CASE-SPECIFIC MODEL FOR REAL-TIME IMAGE PROCESSING AS LINKED TO DETERMINATION OF TRAFFIC DENSITY

NAMAN SEHGAL

ABSTRACT

Today, live traffic updates are available to the consumers on their tablets and car navigation devices with the help of traffic portals like traffic.mapmyindia.com, www.btis.in and applications like Google maps. This paper proposes the technique of contour counting followed by filtering the results on the basis of their sizes to estimate the traffic density at a junction. Next, a pixel counting method is proposed which estimates the traffic density in the real-time image on the basis of the change in the image from the reference traffic less image. First, the sequence of images is acquired from the traffic light camera and the edges are detected using the most efficient edge detection technique. Then the resultant images are used to compute the traffic density at the junction by the above-mentioned methods. By processing the resultant image we determined the green traffic light time. Finally, all these applications were reinforced into a solitary graphical UI.

Keywords—Traffic live update, Traffic density, Edge detection, contour counting, pixels counting, Image processing.

INTRODUCTION

Traffic congestion is an ever-increasing problem in the urban cities all over the world. Local government authorities in big cities have been making an effort to monitor their city‟s traffic at important junctions and roads. Different traffic management systems are being adopted, in an attempt to solve this problem. Some systems make use of inductive-loop traffic detectors installed in the pavements, which can detect moving vehicles crossing a particular point on the roadway. The SKOOT system implemented in cities like London, Toronto and Beijing etc. is an example of an elaborate traffic management system, which involves the use of similar sensors to enhance efficient traffic flow. However, the installation and maintenance of such systems is an expensive affair.

Nowadays, a relatively cheaper alternative to present the local traffic information is in the form of a video or a sequence of images captured by a camera mounted on a pole or traffic light surveilling the junction. In contrast to the traditional techniques, video-based systems have several advantages. They provide a greater amount of traffic information, combine surveillance with traffic control, are easy to install and can be easily scaled to meet the improving benchmarks in the field of digital image processing. Around the world, a number of computer vision based traffic management systems (CVTMS‟s) have been proposed in the last decade, for example, the projectROADWATCH at U.C. Berkley aimed at measuring various traffic parameters using real-time computer vision [1].

In this study, we propose a system, which utilizes digital image processing techniques to determine the traffic density at a junction and to compute the duration of the green light in order to ensure a better traffic flow. The traffic density results determined through the proposed analysis can be used to inform the public about the status of the traffic on different routes throughout the city. A similar system has been implemented in the city of Bangalore and the live traffic status is available to the public via an online portal at www.btis.in. The snapshot from the website in figure 1 shows how the system indicates traffic on different routes in Bangalore. The red dots mark heavy traffic, orange dots mark intermediate levels of traffic and green dots represent routes with light traffic.

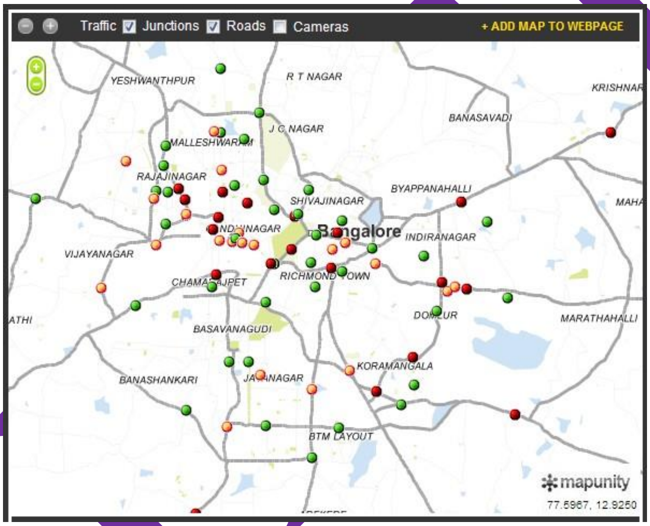
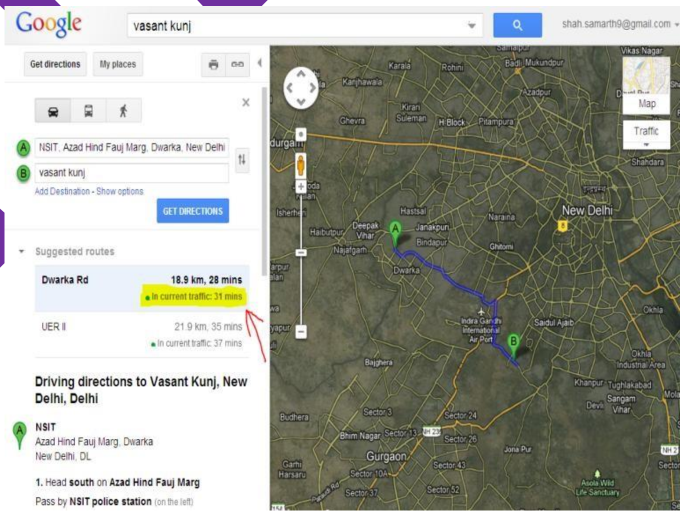


Figure 1: Traffic density on different routes on BTIS website



NSIT, Azad Hind Fauj Marg, Dwarka, New Delhi
vasant kunj
Add Destination - Show options
GET DIRECTIONS
Suggested routes
Dwarka Rd 18.9 km, 28 mins
In current traffic: 31 mins
UER II 21.9 km, 35 mins
In current traffic: 37 mins
Driving directions to Vasant Kunj, New Delhi, Delhi
NSIT
Azad Hind Fauj Marg, Dwarka
New Delhi, DL
1. Head south on Azad Hind Fauj Marg
Pass by NSIT police station (on the left)

Figure 2: Use of Traffic density estimation in Google maps

Apart from this, a major use of such a system can be in an application like Google maps where the traffic density on a route is estimated to determine the time needed to reach the destination. Currently, Google uses data from GPS of android mobile phone users to estimate the traffic in countries like India where their StreetView technology isn"t fully functional[2].This certainly isn"t a very reliable source for estimating traffic.

PROPOSED SYSTEM

The different stages of the proposed idea are as shown in the flowchart of fig.3.

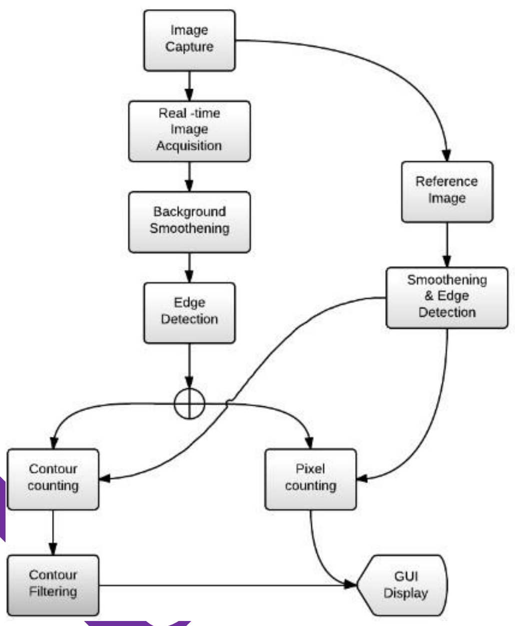


Figure 3: Flowchart for the proposed system.

A. *Background Smoothening*

Smoothing filters are utilized for blurring and for noise reduction. Blurring is utilized as a part of preprocessing steps, such as removal of small details from a picture earlier to(large) object extraction, and spanning of small gaps in lines or bends. Noise reduction can be accomplished by blurring with a linear filter and also by non-linear filtering.Usually, a Gaussian filter is used since it can be computed using a simple mask. Some edge detection techniques like canny edge and laplacian of Gaussian may not require such preprocessing, but this step is generally includedin order to maintain generality.

B. *Edge Detection*

Edge identification alludes to the way toward recognizing and finding sharp discontinuities in a picture. These discontinuities are abrupt changes in pixel force, which stamp the limits of articles in a picture. The edge location calculation has been utilized as a part of this procedure before coordinating basically in light of the fact that it has preferable execution over the backgrounddifferencing operations. The shades of autos and the incorporating lighting changes in traffic scenes are less delicate to edge discovery. There are four steps involved in an effective edge detection algorithm. First is *smoothing*which is used to suppress as much noise as possible without damaging the true edges. The second step is an *enhancement, which* is achieved by applying a filter to sharpen the existing edges. Next is *detection* or *thresholding,* used to determine whether an edge pixel should be retained or discarded. Points which lie on the edge can be detected either by distinguishing the nearby maxima or minima of the main subordinate (Gradient strategy) or by recognizing the zero intersection of the second subsidiary (Laplacian technique). At last, we require *localization* to decide the accurate area of the edge. This involves edge thinning and linking which are vital for obtaining continuous contours.

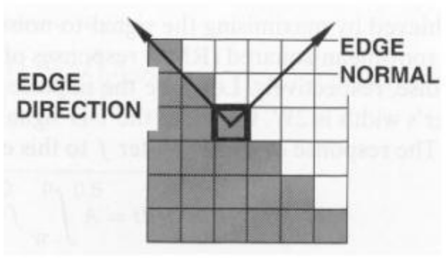


Figure 4: Direction of gradient in the direction of edge normal

A detailed explanation of edge detection is given in [4]. The following six edge detecting algorithms are available in MATLAB:

(1) Canny edge detection
(2) Sobel operator
(3) Prewitt operator
(4) Roberts operator
(5) Zero Crossing edge detection
(6) Laplacian of Gaussian

The first four techniques mentioned above are examples of a gradient method while the last two techniques utilize the laplacian method.

As edge detection is a fundamental step in PC vision, it is important to call attention to the genuine edges to get the best outputs from the coordinating procedure. That is the reason it is vital to choose edge identifiers that fit best to the application. In this respect, we first present a correlation result between the six different edge discovery methods accessible in MATLAB. The comparison results showed that the Laplacian of Gaussian method was more accurate in detecting the actual number of cars on the road.

We must make sure that once we have selected the edge detection algorithm, it must be used throughout the analysis that is for all the real time images as well as the initial reference image. A theoretical comparison between these different edge detection techniques can be found in [5].

C. Contour counting:

This approach can be used for making a rough estimation of the number of cars at the junction, by counting the total number of closed contours formed after the edge detection has been performed. The assumed principle is that each vehicle contributes to a single contour, hence by counting the number of loops/contours and filtering them by their sizes we can compute the traffic density. The contours can be detected using the Moore Neighborhood algorithm. The algorithm can be implemented using the „bwboundaries‟ command in MATLAB, which stores the values of the boundary pixels in an array cell. The cell has N arrays each array is M by 2in size, storing the values of the boundary pixels (i.e. 1 in this case as the boundaries are white). Here N is the total number of closed contours detected by the „bwboundaries‟ command and M is the number of boundary pixels ina particular contour. The value of M varies for each contour.

Since we want to detect only the contours, which are big enough to be that of the car‟s body, we can filter the total detected contours on the basis of their sizes (i.e. on the value of their M). For the sample set taken in our experiment, we eliminated all the contours having less than V pixels in its periphery. These smaller loops/contours are due to windows, tires etc. The final count of the contours can be adjusted by subtracting the number of large contours detected in the traffic free reference image. This way we remove the additional loops due to potholes, trees, any form of construction etc.

The value V was determined by statistical analysis of the data obtained from different images of the traffic. Let the actual number of vehicles in the real-time image be N. Then the first N values of the contour sizes (arranged in decreasing order of size), were extracted and the value of the N_{th} and the $(N+1)_{th}$ contour. The average of all these values for different cases was set as the threshold value V.

D. Pixels counting:

An alternative approach to contour counting can be by monitoring the change in the white pixels of the real-time image with respect to the reference image. The basic principle used is that greater the change in the captured image, greater is the traffic. In a case of a complete match or no change, there is no traffic. Ideally, the reference image should be devoid of any vehicle and should comprise ofonly the fixed objects like buildings, trees, potholes etc.

We can break our objective into two tasks: a) Determining traffic density and b) Computing traffic light time for traffic management.

D.1. Determining traffic density

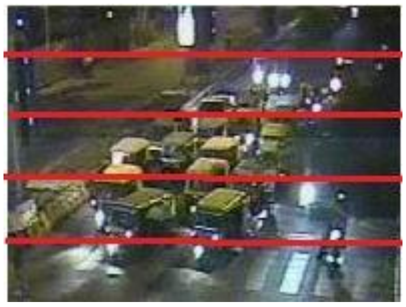
The technique used for estimating the flux of cars involves calculating the ratio of the number of white pixels to the total number of pixels in the real time image itself. So we compute the number of white pixels from the edge-detected image consisting of only black and white pixels. Let the number of white pixels be „*w*‟. The standard size of the images used in the analysis is 288×352 pixels hence the constant denominator (total pixels) will be 101376 (it can also be any other constant value). Thus we find:

$$Density = \frac{w}{101376}$$

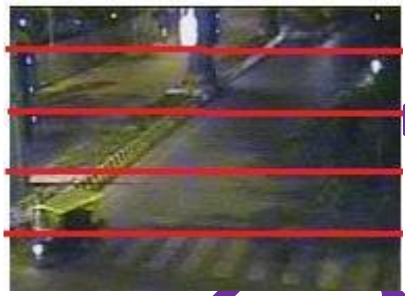
For example, when the road is completely traffic free the ideal count of white pixels is zero and hence Density→0. As the traffic increases, so does the count of white pixels and the density. In order to keep a check on the traffic, we can set threshold values to classify the density ratio as *low, medium* and *high*. This information can be shared with the public through the Internet for their convenience and better traffic management.

D.2. Determining the traffic light duration for traffic management

In order to have a smooth traffic flow, it is essential that the duration of the traffic signal‟s red light is regulated on the basis of the actual traffic present at the junction. In our experiment, we divide the image into five horizontal zones as shown in Fig.5. The corresponding zones of the real-time image and the reference image are matched sequentially starting from the bottom to the top (i.e. from the zone nearest to the signal to the farthest one). As soon as there is at least 60%(or any other desired percent) matching found in a particular zone, no further matching is performed and the duration of the red light is set to a value corresponding to the matched zone. For example, if there is no traffic,the very first zone will match the reference image‟s first zone and hence the matching will stop there and a maximum time will be allotted to the signal‟s red light. Similarly, if there is no traffic in the third zone then the matching will stop after three iterations and appropriate time will be allotted to thered light. Minimum time will be allotted if there is no sufficient matching in any of the zones implying a traffic jam. For our experiment, we multiplied the number of zones by 10 to get the maximum time limit for the green light.



a)



b)

Figure 5: An image divided into five zones a) crowded b) empty

RESULTS AND DISCUSSIONS

A. Contour counting method:

In order to compare between the different edge detection techniques, we performed all the techniques on a single traffic camera image. The resultant images of all the edge detection techniques are shown in fig. 6. By the naked eye, it was observed that the canny edge detection and laplacian of Gaussian method stood out as the most effective techniques for detecting edges in a traffic camera image as compared to the other four techniques. This perception was further substantiated with the outcomes acquired by the contour counting method.Table 1 shows the contour counting results for a set of ten different images taken.

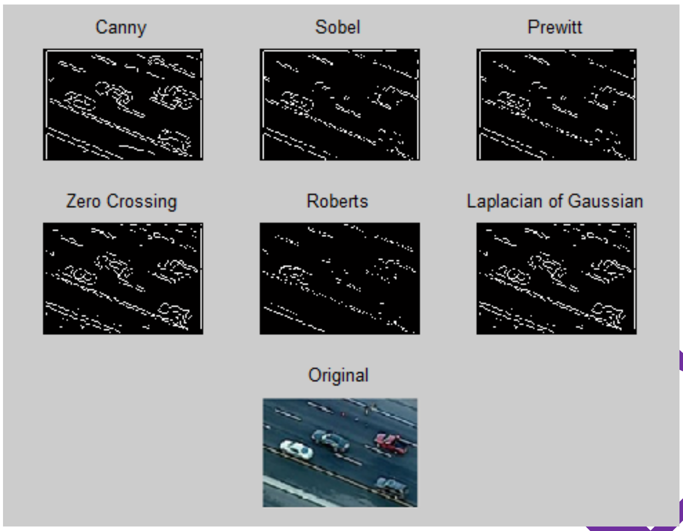


Figure 6: Visual comparison results of different edge detection algorithms in MATLAB.

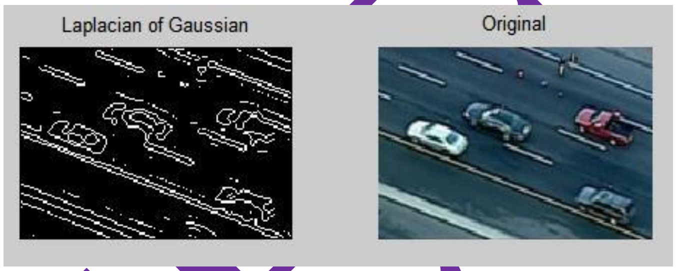


Figure 7: A magnified view of the sample image for contour counting method and its Laplacian of Gaussian result.

Table 1: The result of contour counting method with different edge detection algorithms.

S.no	Actual no. of vehicles	Canny Edge	Sobel	Prewitt	Zero Crossing	Roberts	Laplacian of Gaussian
1.	2	3	1	1	1	1	2
2.	5	6	3	3	4	3	4
3.	4	5	2	1	2	0	4
4.	5	6	2	3	4	0	5
5.	7	7	3	2	5	3	7
6.	4	4	1	1	2	1	4
7.	3	3	0	0	2	1	3
8.	4	5	2	2	2	2	2
9.	1	1	0	1	1	0	1
10.	0	0	0	0	0	0	0
Avg. accuracy in detection (in %)		85.72%	40%	37.14%	62.8%	31.4%	91.43%

The analysis of results in Table 1 shows that Laplacian of Gaussian method is marginally better than the canny edge technique and gives 91.43% accuracy, as compared to 85.72%. Canny edge technique proved to be more sensitive even to the very faint edges like those due to shadows; hence insome cases, it detected anumber of cars than the actual number of cars. On the other hand, the Laplacian of Gaussian operator has the highest reaction at the focal point of blob-like structures in thepictures[6] and turns out to be more efficient for detecting the actual number of cars.

Thus by determining the number of cars at a junction or roadway or highway we can estimate the traffic density of that location. Such estimation requires calibration as different roads have different traffic capacities and are prone to different forms of errors.

The accuracy in Table 1 was calculated by the following formulae:

$$Accuracy\% = \frac{|Actual\ no.of\ cars - Detected\ no.of\ cars|}{Actual\ no.of\ cars} * 100$$

B. Pixel counting method:

The traffic density computed using pixel counting is tabulated in Table 2. The table shows the amount of traffic as a percentage of the total traffic that can be accommodated on the road. To achieve this we have taken the ratio of the observed white pixels and the total pixels of the image. The results are shown for five timings. Each time sample is taken at an interval of 15 minutes each. It must be noted that the maximum possible traffic will never result in a 100% value but instead it will be a value much lower than that since only the edges of the vehicles are shown by the white pixels. Hence we need to calibrate the obtained results for better understanding and easier use.

Table 2: Estimated traffic density using pixel counting method

% Traffic	79.68 %	55.04 %	65.10 %	72.56 %	91.75 %
Time (In hrs.)	1300	1315	1330	1345	1400

C. Graphical User Interface (GUI)

Finally, a MATLAB-based graphical user interface was developed to present the possible applications of the data obtained by the pixel counting method. The developed GUI can be used to plot a graph of *traffic density versus time* using the data of table 2. Another function of the GUI is to control the duration of red and green light at the junction for better traffic management as discussed in section D.2. Figure 9 shows the snapshots of the GUI.

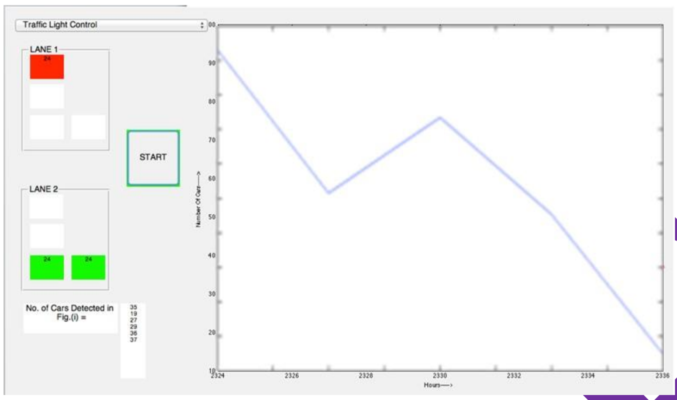


Figure 8: Snapshot of the GUI.

CONCLUSION

In this project, we have successfully implemented image-processing algorithms to determine the traffic density at a junction and to regulate the timings of the traffic light at the junction for efficient traffic management. Initially, an analysis of different edge detection techniques was done and it was observed that Laplacian of Gaussian technique gave the most accurate results. Two different methodswere proposed to determine the traffic density at the junction. First was by contour counting and second, by pixel counting. The pixel counting technique was also used to regulate the time of the traffic light. Reference images were introduced to help in eliminating the noise produced by the unwanted fixed objects detected in the image like trees, buildings, potholes etc. A size based filter wasalso introduced to remove the additional contours produced due to wheels, windows etc. Finally, a GUI was developed to plot the graph of *traffic density vs. time* and to demonstrate the regulation of thetraffic light based on the real-time traffic.

REFERENCES

[1] Y. Wu, F. Lian, and T. Chang, "Traffic monitoring and vehicle tracking using a roadside camera," IEEE Int. Conf. on Robotics and Automation, Taipei, Oct 2006, pp. 4631– 4636.

[2]http://googleblog.blogspot.in/2009/08/bright-side-of sitting-in-traffic.html(29/04/13)

[3] Z. Jinglei, L. Zhengguang, and T. Univ, "A vision-based road surveillance system using improved background subtraction and region growing approach," Eighth ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, August 2007, pp. 819-822.

[4]Trucco, Chapt 4 AND Jain et al., Chapt 5), University o Nevada, Reno “Edge Detection”

[5] http://stackoverflow.com/questions/13429134/canny-edge-detection-and-log- difference(29/04/13)

[6]Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", 2nd edition, Prentice Hall, 2002.

[7] K. Wang, Z. Li, Q. Yao, W. Huang, and F. Wang, "An automated vehicle counting system for traffic surveillance," IEEE Int.Conf. on Vehicular Electronics and Safety, Japan, Dec 2007, pp. 1-6.

[9] Ahmed S. Salama, Bahaa K. Saleh, Mohamad M. Eassa, "Intelligent Cross Road Traffic Management System (ICRTMS)," 2nd Int. Conf. on Computer Technology and Development, Cairo, Nov 2010, pp. 27-31.