

Traffic Image Processing System

Pranav Maheshwari
Dept. of Physics, Delhi
Technical University
Kirti Nagar, New Delhi, India

Deepanshu Suneja
Dept. of Computer Science,
MSIT, IP University
Kirti Nagar, New Delhi, India

Praneet Singh
Dept. of Computer Science,
MSIT, IP University
Dwarka, New Delhi, India

Yogeshwar Mutneja
Dept. of Computer Science, ITM University
Rajouri Garden, New Delhi, India

ABSTRACT

Traffic flow never remains the same on any given time of the day, varying substantially from morning to evening, generally peaking in the evening office hours. A common man has to face different traffic conditions in his daily routine from nerve racking traffic jams to almost empty roads. The fixed nature of the traffic lights fail to take this in to account and all this leads to increase in waiting time for every vehicle and thus wasting precious time. This increase in waiting time has a cascading effect on fuel consumption too and thereby having severe consequences on the environment. In this article, we shed light on these issues and present a dynamic system that overcomes all these drawbacks. We make use of web cameras mounted on street lights to capture still images of the traffic which then undergo a series of steps related to Image Processing and Image Analysis. Finally, we calculate the traffic volume and operate the traffic light's timer accordingly. All this is automated with negligible scope of human error and a quick response time. The system helps in minimization of several factors like waiting time, fuel consumption and congestion..

General Terms

Gray Scale, Edge Detection, Binary Image Conversion, Image Analysis.

Keywords

Traffic Analysis, Edge Detection, Real time, Day and Night.

1. INTRODUCTION

In the current scenario of fixed time traffic lights, many a times situation arises wherein there is heavy incoming traffic only from one side of the intersection and the rest are relatively empty. In this case the people on the heavily occupied side have to wait for unreasonably long time as the green light timer is fixed for each side which fails to take in to account that there is no traffic to pass from the other sides.[4] This prolonged waiting time increases the average waiting time of every person in the traffic. Though care is taken while setting these timers by government officials according to the proportionate amount of traffic present on different sides of the intersection but this can never be so flexible as to adapt to the dynamic traffic throughout the day. Moreover, some areas of high traffic volume may receive scanty traffic at some point of the day and some low traffic volume areas might get congested likewise, which leads to an additional increase in waiting time because these timers are set according to average volume of traffic corresponding to the different areas of the

city respectively but not according to the different hours of the day. This fixed nature of the present traffic light timers turns out to be ineffective not only during the day but also during late night hours. At night, when there is negligible traffic present, the timers lead to an unreasonably long waiting time. Additionally most people, seeing empty roads ahead of them, tend to jump lights at this point, which quite often lead to accidents.

Our proposed system takes into account all these issues by dynamically changing traffic light timers. It intelligently recognizes the volume of the traffic at each side of the intersection thereby providing an adequate amount of time for the traffic to pass. The system monitors traffic throughout the day and takes care of the inability of the fixed timers to adjust as per the traffic. Since our system regulates the flow of traffic at night also, it helps in minimizing the chances of an accident.

2. LITERATURE SURVEY

AVikramaditya Dangi, Amol Parab, Kshitij Pawar & S.S Rathod[1] implement a system in which they perform real time digital image processing of the traffic. They make use of a camera installed on high structures that captures the current traffic scenario. The image captured is then analyzed using canny edge detection algorithm, which they have found to be the most efficient algorithm. Afterwards, they count the number of vehicles present at the intersection using Moore-neighborhood algorithm and the traffic is accordingly managed.

There is no special need for counting the objects, rather we can work with the total volume of the traffic present and save additional computing time.

Omkar Ramdas Gaikwad, Anil Vishwasrao, Prof. Kanchan Pujari, Tejas Talathi[2] proposes the implementation of traffic control using MATLAB and morphological operations. In their system they acquire an image, apply RGB to greyscale conversion and morphological operations. They propose to color the road with successive green strips near the traffic lights which are used for the amount of traffic present. After conversion of the image into greyscale the visible green strips turn white and thus giving an idea of the traffic present.

This method has some cost restraints as we have to devote significant resources to color the roads green for the system to work. Also, successive image matching applied by their

system can be avoided by just analyzing a single image and saving processing time.

Chandrasekhar. M, Saikrishna.C, Phaneendra Kumar[3] present a system to control traffic which uses a web camera installed around the intersection which captures images of real time traffic condition. Then the number of images captured in a small interval of time is matched with a reference image using image processing algorithms. They primarily make use of SIFT algorithm. This image matching method will increase the computation time of the system as at a single point of time it makes use of two images.

Our system uses a single image for performing edge detection and computing the volume of the traffic at a particular point of time. Instead of counting the number of objects we compute the length of the traffic present from the traffic light which significantly reduces the computation time.

3. WORKING AND DEVELOPMENT

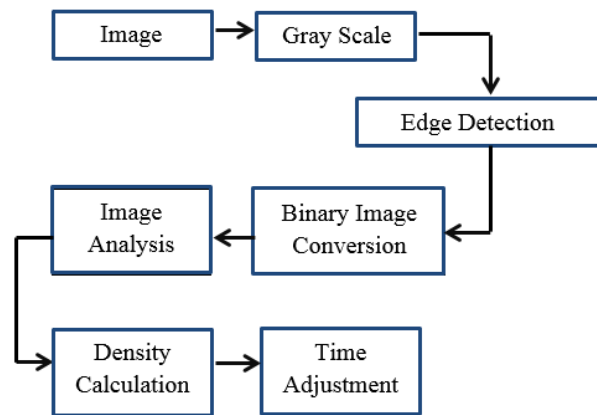


Fig. 2: Work Flow of the system

3.1 Gray Scale

The colored images acquired by the cameras are transmitted to the servers where they are resized and converted in to grayscale. Resizing is necessary as it reduces the processing time of the operation by reducing the no. of pixels to be processed and analyzed. Converting the color image in to grayscale is done because the RGB values of the individual pixels have no role to play in the process being carried out and would just create complexity for further processing. Grayscale image has pixels with value of intensity varying from 0 to 255, with 255 being the highest intensity of white.

3.2 Edge Detection

This grayscale image then undergoes edge detection. There are multiple methods to implement edge detection. Different methods have different pros and cons. We required a method that had good balance in the amount of detail and the processing times. A very high amount of detail, which can be provided by Canny Method, would lead to each edge getting detected regardless of its significance or prominence. After comparing the Prewitt and Sobel Methods, we zeroed down on Sobel as it was slightly better at detecting edges with almost similar processing time as Prewitt.

3.3 Binary Image Conversion

After edge detection, the image is turned in to a bilinear image, i.e. it either has pixels with value of 255 or 0. There are no grey levels left in the image. This is done by comparing each individual pixel value to a threshold value, which is different for different setups. If the value of the pixel is less

than the threshold, it is put as 0 and if it is greater than or equal to the threshold, it is put as 255. The bilinear image only has the most prominent edges left as they tend to have a value greater than the threshold value. The bilinear image further undergoes de-noising where solitary pixels, i.e. pixels which have value different than all of its neighboring pixels, are removed by median filtering. The end result of this marks the end of image processing and the final image is ready to be analyzed.

3.4 Image Analysis

A relation is formed between the pixel location and its distance from the base of the street light where the camera is mounted. This is achieved by forming a trigonometric relation which utilizes various values like the Pitch Angle, Vertical Angle of View of the Camera and Height of the street light. When the distance of corresponding to each pixel has been calculated, we can calculate the distance among the pixels also which helps us in determining the distance from the beginning of the road to the point of interest.

3.4.1 Day Time

For analysis during daytime, the value of pixels is scanned across straight lines originating from the beginning of the road and going right till the end of the road, as seen in the image. These lines are evenly spread and placed in a way that they have maximum chances of passing through the cars on the road. While scanning the pixels lying on these lines, we check their values. If the pixel has a value of 255, it means that it belongs to some part of a vehicle. If the value is 0, then it either belongs to the surface of the road or it might still belong to some part of a vehicle. When simultaneous black pixels are determined which, through the relation of distance and pixel developed earlier, encompass a distance of 3 meters or greater, it is decided by the program that these pixels belong to the road and they signify that the high density traffic has been accounted for. A similar process is carried out across various lines and finally the second highest distance covered by all of the lines is considered as being close to the mean and the correct distance. The line with the greatest distance might have been influenced by the shadows being cast on the roads and is therefore considered to be erroneous, even though this might not be correct. Since we need a median distance and people generally line up their cars in a uniform manner, the assumption that has been made can't have adverse effects on the output. The distance that is calculated is then multiplied with the width of the road which gives the area being covered by the traffic and through this we can give an appropriate time for the green light.

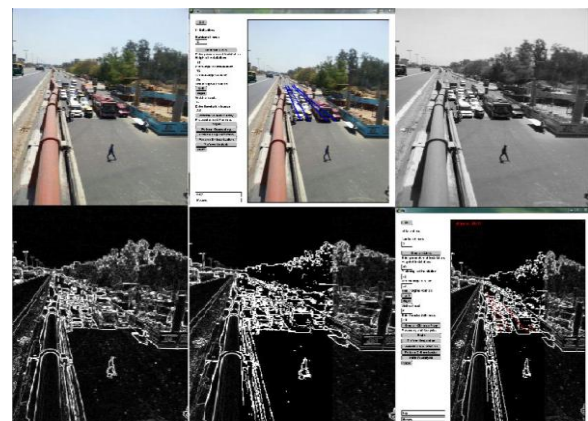


Fig. 2: Day Time Analysis

3.4.2 Night Time

Analysis during night time is carried out in a similar way as day time except that we do not perform edge detection on our given image. This is because we are using the sharp intensities of the headlights to identify the vehicles instead of their edges which are blurry at night. Setting a very high threshold value, Binary Image Conversion would leave us with only white spots which represent the headlights of the vehicles present on the road. Each white pixel of these spots is then stretched across its x co-ordinate to form a horizontal line. Then we make use of straight lines going across the middle of each lane, originating from the beginning of the road and going right till the end of the road, which cut across the set of white lines generated earlier. Similar to daytime, the line which covers is considered as the line with the longest distance might have been influenced by some noise and the line with the shortest distance might have been due to the presence of a long vehicle like a bus or a truck whose body length would have crossed the limit of 7 meters thereby terminating the line.



Fig. 3: Night Time Analysis

3.5 Traffic Volume Calculation

The relation between distance and pixels has been determined by applying trigonometry. Every camera has a specific angle of view which is usually stated for the diagonal of the image. For our purposes, we need the vertical angle of view to carry out the calculations. Taking the height of the street light as the perpendicular of a right angled triangle, the base will equal the distance from the pole. The no. of pixels lying vertically in the image in totality cover the entire vertical angle of view and by unitary method we can find that what angle is covered by a specific no. of pixels. For example if there are 100 pixels and the vertical angle of view is 60 degrees, each pixel covers 36'. Once the angle subtended by the camera between the first pixel and each subsequent pixel in the image has been calculated, we can find the distance between the starting pixel and each pixel thereafter by $height/\tan(\theta)$ where θ is the angle which was calculated earlier.

$$dstb = iht * \tan(0.01746 * (pth - vc/2 + (ht - fy)/ht * vc))$$

$$dsti = iht * \tan(0.01746 * (pth - vc/2 + (ht - y)/ht * vc)) - dstb$$

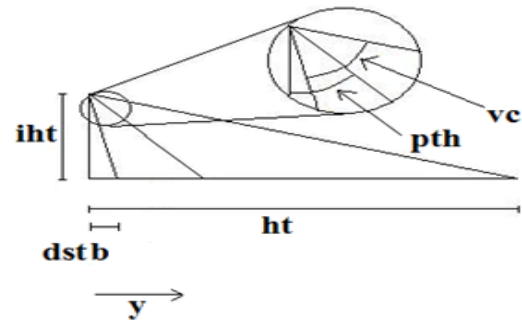


Fig. 4: Camera Angle Calculation

iht = height at which camera has been installed

ht = width of the image

pth = pitch angle

vc = vertical angle of view

$dstb$ = distance from which road begins wrt camera

$dsti$ = distance to any point wrt $dstb$

4. FUTUREWORK

For further expansion of our project, we decided on two modules for ambulance detection [5]. In our first method, we use a sound recognizing module which consists of a microphone and audio fingerprinting software. Whenever the microphone hears an ambulance's siren it matches it with a pre-stored audio of the siren. This audio matching triggers the camera to capture two successive images, one containing the siren's high intensity side of the red/blue beacon and the other one containing the low intensity of the red/blue beacon. This helps our system to detect the ambulance's position in the traffic and set our timer of the respective side of the intersection to a low value so that the ambulance is able to maneuver its way through the traffic quickly.

In our second method, we detect the presence of an ambulance in the traffic using photo diodes. The diodes receive a pattern that is produced by the ambulance's beacon and carries a similar pattern matching technique.

5. EXPERIMENTAL RESULT AND CONCLUSION

Our system optimizes the average waiting time of every person present in the traffic. It is set to be automatically activated just 5 seconds before the light turns green. This way it recognizes the traffic present at all sides of the intersection and sets all the timers according to real time. It successfully completes the processing of the image in less than 2 seconds. The system also effectively works at night and takes almost 3 seconds in this case. Some of the advantages of this system are that we use just one image for the entire processing. Simple calculation is required for estimating total distance, involving trigonometry. No detailed image processing has been involved, simple edge detection is carried out to judge the volume of traffic using coordinate geometry. The cost involved is quite low as even a 0.4MP camera can be used. Most importantly, it has a high accuracy, with low computation time ranging from 0.5-2 seconds.

4 secs yellow light

5 secs threshold time.

$$\frac{(\text{areacovered})}{(\text{maximumarea})} * (\text{timeallotted} - \text{thresholdtime}) + \text{thresholdtime} + \text{yellowlight}$$

6. REFERENCES

- [1] Vikramaditya Dangi, Amol Parab, Kshitij Pawar & S.S Rathod "Image Processing Based Intelligent Traffic Controller", Undergraduate Academic Research Journal (UARJ), ISSN : 2278 – 1129, Volume-1, Issue-1, 2012.
- [2] Omkar Ramdas Gaikwad, Anil Vishwasrao, Prof. Kanchan Pujari, Tejas Talathi, "Image Processing Based Traffic Light Control", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 3, Issue 4, April 2014.
- [3] Chandrasekhar. M, Saikrishna. C, Chakradhar. B, Phaneendra Kumar. P & Sasanka. C, "Traffic Control Using Digital Image Processing", International Journal of Advanced Electrical and Electronics Engineering, (IJAEED).
- [4] B. Fazenda, H. Atmoko, F. Gu, L. Guan1 and A. Ball, "Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems," ICCAS-SICE, Fukuoka, Aug 2009, pp.4250- 4255
- [5] Pezhman Niksaz, Science & Research Branch, Azad University of Yazd, Iran, "Automatic Traffic Estimation Using Image Processing", 2012 International Conference on Image, Vision and Computing