ISSN 2278-3083
Volume 3, No.2, March - April 2014
International Journal of Science and Applied Information Technology
Available Online at http://warse.org/pdfs/2014/ijsait05322014.pdf

Total Load at Junction using CCTV Cameras

Saroj Kumari [1], **Manisha Rathore** [2]
[1] Banasthali University, India, Sarojpaperkumari@gmail.com
[2] Banasthali University, India, Manisharathore1416@gmail.com

ABSTRACT

Worldwide, traffic is the major problem to be faced in urban areas. Traffic load, defined as the measure of total number of vehicles passed at the crossings in a day, month or year along a particular road. The implementation will be performed with the help of traffic images captured through CCTV cameras installed at crossings.

This document presents a new effective method for computing load of vehicles at the junction using MATLAB; this technique can be used in management of traffic control. Firstly foreground objects is captured through subtraction of two images. Load will be computed by counting objects, presence of object will be detected by computing location of individual vehicle in the traffic image captured at the junction.

Keywords: – MATLAB; traffic control management; CCTV; subtraction

1. INTRODUCTION

Load computation at junctions, a new application and plays an important role in Intelligent Transportation System. The computed traffic then used to do some important calculations such as green light computing and the current state of traffic is monitored. This document concentrated at counting the vehicles in the traffic images captured through CCTV cameras installed at junction for real time traffic signal control. In recent years, this is one of the applications of image and video processing. This can be performed with the help of MATLAB Environment.

For detecting vehicle load, two source images have to be given as input to the system, one is blank road image and other is corresponding road image with vehicles. The two images loaded are subtracted, so that area where vehicles are present can be determined accurately. Exact location of vehicle is detected by computing intensity change, this can be done when traffic image is scanned from top to bottom and RGB intensity of each pixel is computed. If change in intensity is observed means there is a probability of occurrence of vehicle. Count the places where change in intensity is observed. Half of the counted intensities are total number of vehicles present in the traffic image. This experiment is implemented in MATLAB Environment.

The paper is organized as follows: Section II contains literature survey of Total Load at Junctions. Section III describes about the MATLAB. Section IV focus on proposed methodology for load detection. Section V shows the practical experiment. Section VI presents the experimental result. Section VII shows conclusion.

2. LITERATURE SURVEY

Gupta A. [1] developed a MATLAB Simulink model to compute traffic load used for real time traffic signal control. Detection of edges was an effective approach used to achieve greater degree of accuracy to set green signal duration.

Gupta P. et al. [2] has worked on traffic load computation using MATLAB Simulink model Blockset. Morphological operation was used as an effective approach. This is useful for counting objects, object detection, and object feature identification. [3] Same work was also performed using an efficient technique in MATLAB Simulink model i.e. Corner Detection. Approach used while dealing with Signal, video and image processing Blockset.

Mahalanobis A. et al. [4] and Christeet et al. [5] mainly focused on image processing. Transformation can be applied either on partial region or on the whole image. Algorithm was developed for detection of logo and predominant vehicle color. Morphological operators used for logo segmentation and fuzzy sets and pertinence rules of pixels used to obtain color. A system was developed to handle both stationary as well as moving objects. Developed system must be able to detect, track and handle moving objects between cameras in real-time. To monitor activities over large areas use multiple cameras in order. Yilmaz A. et al. [6] discussed about the important issues related to tracking including the use of appropriate image attributes, selection of motion models, color and detection of objects. Also, discussed about tracking methods based on the object and motion representations.

Pandey S. et al. [7] worked on traffic load computation for traffic signal control system to compute total traffic load using histogram equalization, edge detection, labeling and removing the noise using median filter. Whereas, Tsin Y. et al. [8] identify and quantify the more significant and heavier vehicle loads. Goyal et al. [9] discussed Visual sensor networks (VSNs) for identifying moving objects from a video sequence and is fundamental and critical task in computer-vision applications.

Chikkali P. S. et al. [10] used Histogram for automatically determine the threshold for determining different region in image. Histogram is also used for segmentation and features are extracted with the help of intelligent computer.

3. MATLAB

MATLAB (Matrix Laboratory) is a fourth generation language used for technical computing. MATLAB allows an easy access to matrix software developed by Linear System Package and Eigen System Package projects. MATLAB is matrix software which is consisting of matrix that does not require pre-dimensioning.

MATLAB supports plotting of data and functions, matrix manipulations, creation of user interfaces, implementation of algorithms, and interfacing with programs written in other languages such as FORTRAN, JAVA, C++, and C etc.

Uses of MATLAB are:

1. Math and computation
2. Scientific and engineering graphics
3. Data acquisition
4. Data analysis, exploration and visualization
5. Algorithm development

Five main parts of MATLAB are:

a. Development Environment

MATLAB environment consist of tools that helps while using MATLAB functions and files. These tools are graphical user interface, Command Window, Command history, an editor, debugger and browsers.

MATLAB is collection of computational algorithms, both elementary function (sum, sine, and cosine) and complex arithmetic, and sophisticated functions (matrix inverse, Matrix Eigen values, Bessel functions, and fast Fourier transforms).

b. MATLAB Language

MATLAB is a fourth generation high-level matrix language consists of control flow statements, input/output, data structures and functions.

c. MATLAB Application Program Interface

MATLAB allows interfacing with programs written in other languages such as FORTRAN, JAVA, C++, and C etc. MATLAB provides various facilities for calling, dynamic linking and calling MATLAB as a computational engine.

d. MATLAB Desktop

MATLAB desktop appears, when you start. And contain various tools for managing variables, files and applications associated with MATLAB.

e. SIMULINK Blockset

Simulink is a data flow graphical programming language tool developed by Math Works. It helps in Building the Model, Simulating the Model, Analyzing Simulation Results, Managing Projects and Connecting to Hardware**.**

4. PROPOSED METHODOLOGY

The flowchart shows the complete procedure for load computation:

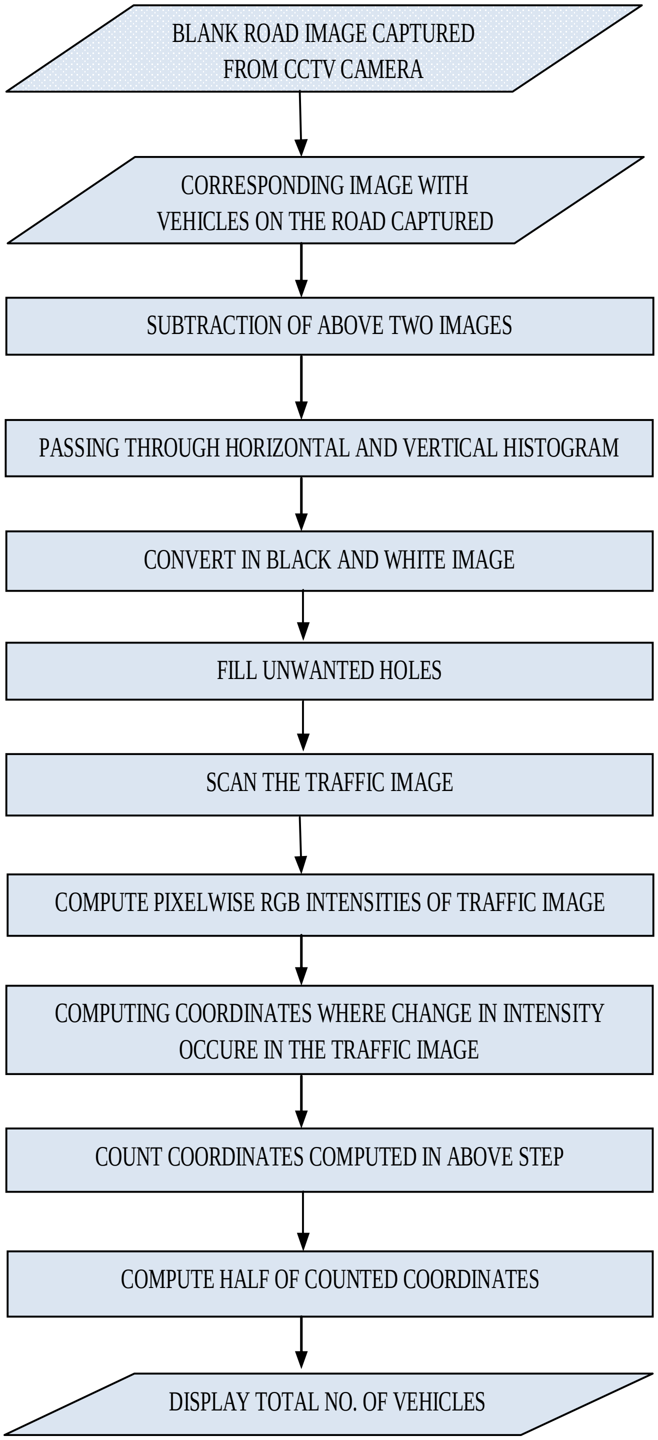

BLANK ROAD IMAGE CAPTURED FROM CCTV CAMERA
CORRESPONDING IMAGE WITH VEHICLES ON THE ROAD CAPTURED
SUBTRACTION OF ABOVE TWO IMAGES
PASSING THROUGH HORIZONTAL AND VERTICAL HISTOGRAM
CONVERT IN BLACK AND WHITE IMAGE
FILL UNWANTED HOLES
SCAN THE TRAFFIC IMAGE
COMPUTE PIXELWISE RGB INTENSITIES OF TRAFFIC IMAGE
COMPUTING COORDINATES WHERE CHANGE IN INTENSITY OCCURE IN THE TRAFFIC IMAGE
COUNT COORDINATES COMPUTED IN ABOVE STEP
COMPUTE HALF OF COUNTED COORDINATES
DISPLAY TOTAL NO. OF VEHICLES

Figure 1: Load Detection

5. EXPERIMENT

MATLAB is an effective tool used while dealing with problems related to video and image processing it provides a unique and good solution to the selected problem. The experiment can be performed using all versions of MATLAB. Output would be obtained both on the command window and on the figure window.

Steps used to compute Total Load at Junction are as follows:

Step 1: Image Capturing

a) Blank Road Image

Initially, image is captured at the junction of the road when there is no vehicle present on the road (Figure 2).

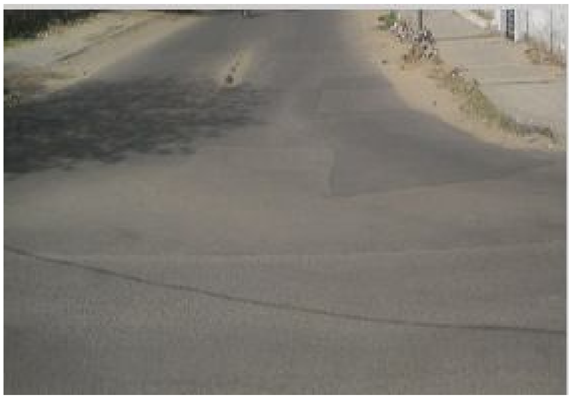


Figure 2: Blank Road Image

b) Image with Vehicles

Capture corresponding image having vehicles on same (when road is empty) road from CCTV Camera at crossings (Figure 3).

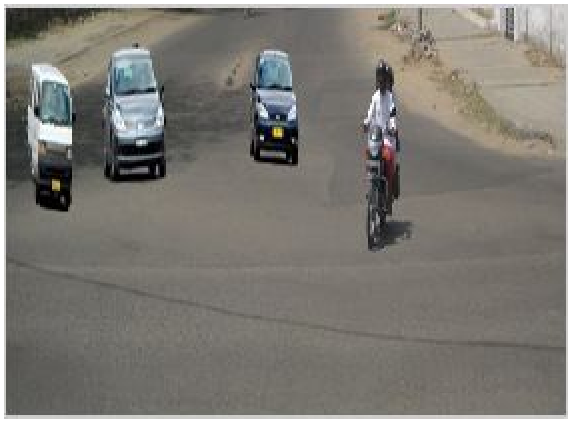


Figure 3: Image with Vehicles on the Road

Step 2: Subtraction of Images

After capturing the background image (containing static objects) is subtracted from the image with vehicles to obtain the foreground dynamic objects.

Remaining steps is applied on the image that is obtained after performing subtraction (Figure 4).

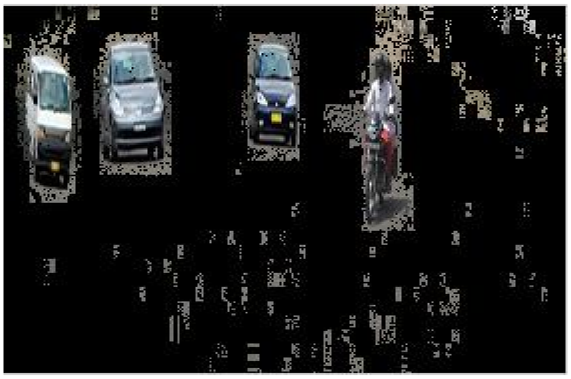


Figure 4: Subtracted Image

Step 3: Vertical and Horizontal Histogram

Vertical and Horizontal edge processing is applied on image (subtracted) by passing image (subtracted) through vertical histogram (Figure 6) and horizontal histogram (Figure 5).

Find all the regions from an image that has highest probability of containing vehicles. Histograms represent the differences of gray values between neighboring pixels of an image column-wise or row-wise. From all detected regions, the region with the maximum histogram value will be taken as the most probable candidate.

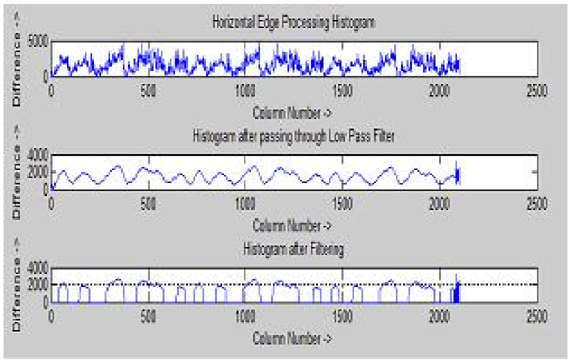


Figure 5: Horizontal Edge Processing

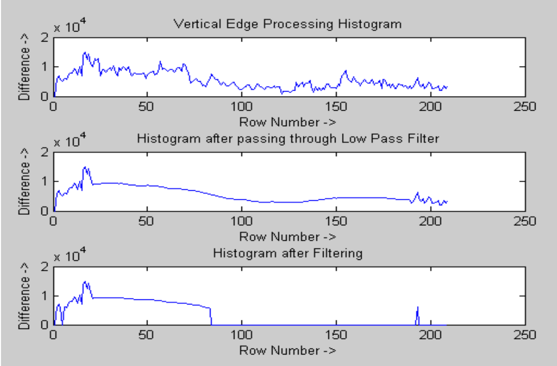


Figure 6: Vertical Edge Processing

After performing vertical and horizontal edge processing a new image is recived, as shown in Figure 7.

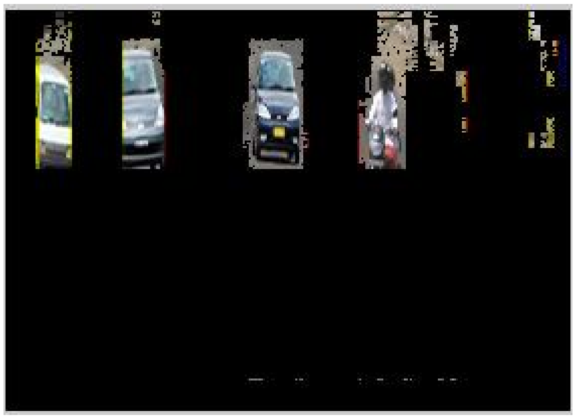


Figure 7: After passing through Horizontal and Vertical Histogram

Step 4: Obtain Black and White Image through Comparison

Black and white image is obtained (Figure 8) by comparing the image obtained through vertical or horizontal histogram (Figure 7) with background image (Figure 1). When, two images are compared pixel by pixel, at particular pixel if pixel intensity value is same for both the images then that pixel intensity value will be set to 1 otherwise set to 0.

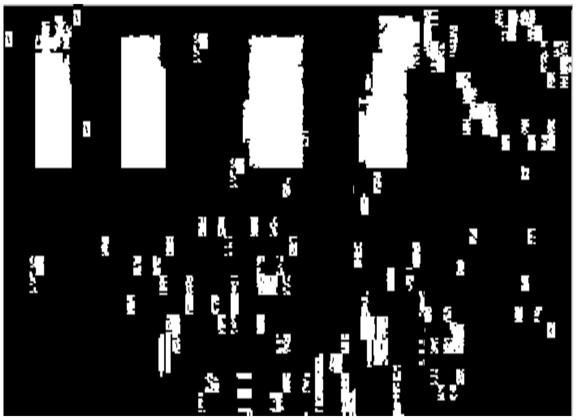


Figure 8: Black and White Image through Comparison

Step 5: Fill Unwanted Holes

Compute the area covered by individual vehicle, if area computed is below the specified area limit then intensity value of all the pixels in that area is set as 0. Through this process unwanted holes are filled; result of this step is shown in Figure 9.

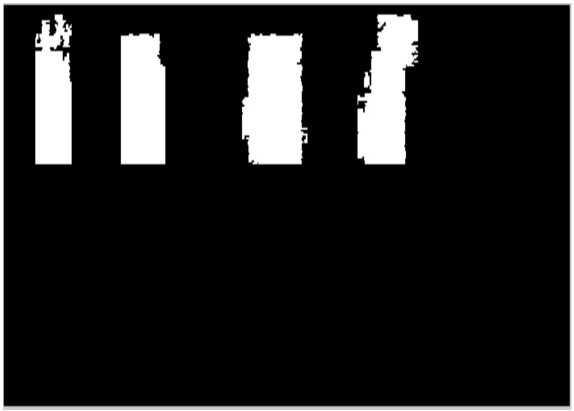


Figure 9: Image after Removing unwanted Holes

Step 6: Coordinate Detection

For coordinate detection above image (Figure 9) is scanned from top to bottom and compute intensity pixel wise.

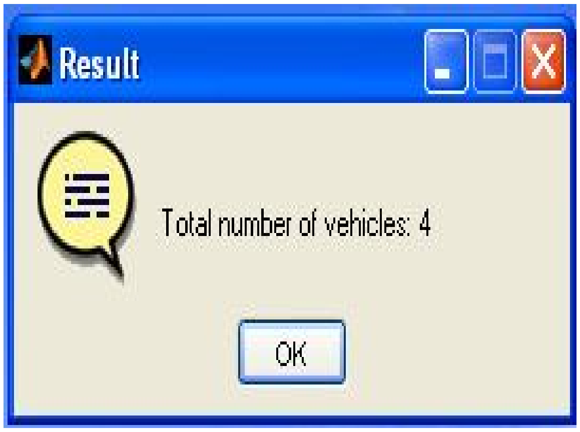
Image free from unwanted holes is used to compute starting and ending coordinates of all vehicles (individually) with respect to both row and column. Coordinate value is produced when sudden change in RGB intensity is noticed (means some positive value of RGB is obtained).

Step 7: Load Computation

While coordinates are detected, a variable count (initialized with 0) is maintained which is updated (incremented by 1) when change in intensity occurred.

Now, divide the count variable by value 2.

Value obtained after dividing will be reflected as the total number of vehicles present in the traffic image captured at the junction using CCTV camera. And Total vehicles counted are referred as load along particular junction (Figure 10).



Result

Total number of vehicles: 4

Figure 10: Total no. of Vehicles

6. RESULT

Result of experiment exhibits that Load of vehicles were estimated properly. Result of experimental is shown in Table 1.

TABLE I. TABLE FOR RESULT

Environment	Real Time Images	Number of vehicles	Result accuracy
MATLAB	1	4	100%

7. CONCLUSION

An efficient image processing technique has been suggested and analyzed for computing total load from CCTV traffic images captured using CCTV cameras across particular junction at an instance. This is an effective approach in Transportation system management. Above Experiment shows that total load at junction were properly estimated with 100% accuracy. Experiment was implemented using MATLAB environment.

REFERENCES

1. A. Gupta, **Hardware Software Co-Simulation for Traffic Load Computation using MATLAB Simulink model Blockset**, International Journal of computational science and information technology, Vol. 2, No. 2, may 2013, pp. 1-12
2. P. Gupta, G. N. Purohit and A. Gupta, **Traffic Load Computation using Corner Detection Technique in Matlab Simulink Model**, International Journal of Computer Applications, vol. 72, No. 16, pp. 6-13, June 2013
3. P. Gupta, G. N. Purohit, A. Gupta, **Traffic Load Computation using Matlab Simulink Model Blockset**, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue- 6, pp. 2427- 2432, June 2013
4. S. A. Christe, M. Vignesh, A. Kandaswamy, **An efficient FPGA implementation of MRI image filtering and tumour characterization using Xilinx system generator**, International Journal of VLSI design & Communication Systems (VLSICS), Vol. 2, No. 4, pp. 95-109, December, 2011
5. M. Y. Siyal, M. Fathi, M. Atiquzzama, **A Parallel Pipeline Based Multiprocessor System for Real- Time Measurement of Road Traffic Parameters**, Elsevier, Vol. 6, No. 3, pp. 241-249, June 2000
6. A. Yilmaz, A. Javed, M. Shah, **Object Tracking: A Survey**, CSUR Journal (ACM Computing Surveys), vol. 38, issue. 4, Article no.13, pp. 1-45, 2006
7. P. Gupta, G. N. Purohit, S. Pandey, **Traffic Load Computation for Real Time Traffic Signal Control**, International Journal of Engineering and Advanced Technology, ISSN: 2249 – 8958,Vol. 2, Issue-4, pp. 891-895, April 2013
8. H. Goyal, **Frame Differencing with Simulink model for Moving Object Detection**, International Journal of Advanced Research in Computer Engineering & Technology, Vol. 2, Issue 1, pp. 263-266, January 2013
9. Y. Tsin, R.T. Collins, T. Kanade, and V. Ramesh, **Bayesian color constancy for outdoor object recognition**, IEEE Computer Society Conferance on Computer Vision and Pattern Recognition (CVPR-01)-Vol. 1, pp. 1132, 2001
10. P. S. Chikkali, **FPGA based image edge detection and segmentation**, Vol. no 9, Issue No. 2, pp. 87-192