2004 IEEE Intelligent Transportation Systems Conference
Washington, D.C., USA, October 3-6, 2004

Automated Extraction of Queue Lengths from Airborne Imagery

Ashish Agrawal and Mark Hickman

Abstract— **In this paper, we propose a methodology to obtain intersection queue length estimates from an airborne image by using connected component analysis of the thresholded region of interest. The queue estimates are obtained by two techniques: by counting individual vehicles in the queue and by the area of a polygon containing the vehicles in the queue. The techniques are tested rigorously on a set of images taken from the cities of Tucson and Seattle and the results are presented.**

Index Terms—**intersection queue length, connected component analysis, region of interest**

I. INTRODUCTION

With the increasing urban congestion, an efficient transportation system has become a necessity of today's world. In order to make the transportation system efficient, it is important to monitor the performance of the intersections constantly since a major portion of the vehicular delay is experienced in queue at the intersections. In addition, frequent feedback of queue lengths can facilitate the automated updating of traffic status of the transportation system. The queue lengths can also be used for determining the storage length and capacity requirements at the intersection. Therefore, there has been significant research [1]-[3] in modeling the traffic flow to estimate the queue length at intersections. However, the need for historical data on queue lengths and the inability to observe the actual queue has limited the applicability of these models. As a result, researchers have investigated other methods such as video monitoring for estimating the length of these queues.

A. Agrawal is with University of Arizona, Department of Civil Engineering and Engineering Mechanics, P.O. Box 210072 Tucson, AZ, 85721-0072, USA (E-mail: aagrawal@email.arizona.edu)

M. Hickman is an Assistant Professor, Department of Civil Engineering and Engineering Mechanics, University of Arizona, P.O. Box 210072 Tucson, AZ, 85721-0072, USA (E-mail: mhickman@engr.arizona.edu)

Video monitoring of the queues is usually done by using fixed or moving cameras. Fixed cameras are generally installed on a stationary platform such as high-rise building or a pole. Hence, the video has a similar background in all the frames. This property is exploited in analyzing the video [4] [5]. However, these cameras have a limited field of view so a more flexible technique of monitoring queues (i.e. a moving camera) has been examined by researchers. The cameras are installed on a moving platform and have the capability to capture the traffic from different locations [6]. However, processing of these videos may be computationally expensive and hence the use of this technique is also limited. Therefore, for practical applications, a technique is sought which can process the visual information obtained from moving cameras in sufficiently short time.

This paper describes a method to analyze a single frame obtained from a moving camera to get information about the queue lengths. The main objective of the paper is to present a technique which can process an image and obtain queue lengths in real time. In this paper, we propose a methodology to obtain the queue estimates from an image by using connected component analysis of the thresholded region of interest. Firstly, vehicles are detected based on the threshold value for the region of interest and the shape of the connected components. The queue length is then estimated by two methods: by counting connected components in the queue and by the area of the polygon in the image containing all the vehicles in the queue. The complete methodology is summarized in the flow chart (see **Fig 1**). The following section illustrates the methodology with an example.

II. METHODOLOGY

a. Cropping the image

A typical image of an intersection taken from an airborne platform encompasses different objects on and around the roadway depending on the field of view (FOV) of the camera. **Fig 2** shows a typical high resolution (2272x1704) image of an intersection. This image captures the intersection of Speedway and Country Club in Tucson taken from an altitude of about 1800 ft above level ground. As a first step, one should select a portion of the image (either by machine or manually) such that it contains entire region of interest. To manually execute this step, a rectangular portion is cut from the image which has its sides parallel to the image edges.

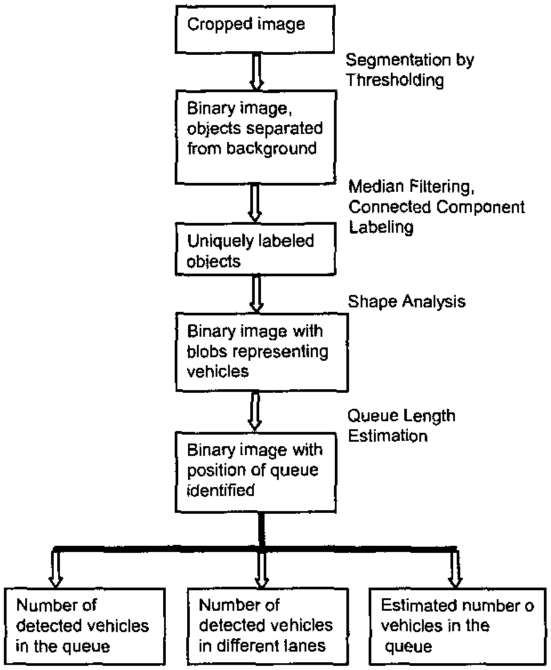


Fig. 1. Flowchart of the methodology

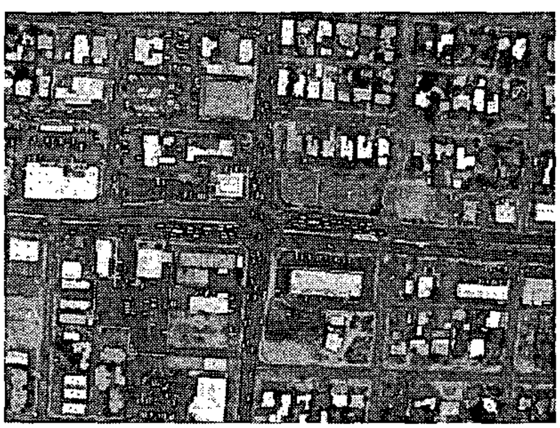


Fig. 2. Original intersection image

Fig. 3. Cropped image

The resulting image contains the entire region of interest, where region of interest signifies the region in the image where the queues are expected. **Fig 3** shows the resulting image after cropping the initial image. It may be noted that the roadway is significantly askew and the roadway boundaries have an irregular shape. Consequently, the cropped image contains much more information than just the intersection.

b. Image segmentation

Let us consider the image histogram for the cropped image. Since the image is cropped to retain the roadway, the highest peak in the histogram represents the road pixels. Moreover, due to the presence of a median, dirt and other off-road objects, small peaks or a shoulder are expected to the right of the highest peak. See **Fig 4**, showing the histogram for **Fig 3**.

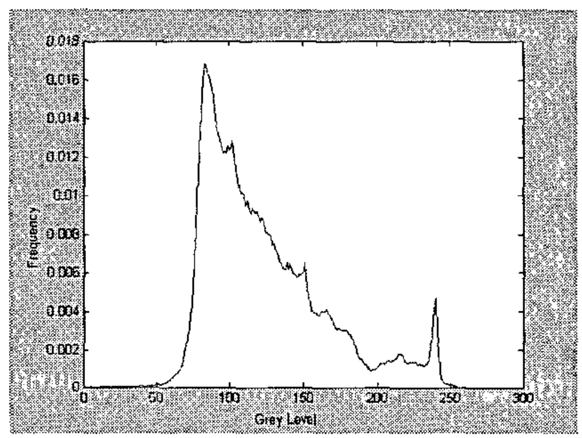


Fig 4 Histogram of the cropped image

For the purpose of this study, it is sufficient to separate the roadway from other objects. Assuming the histogram to be a mixture of normal distributions, a threshold [7] is obtained such that it separates the roadway from other objects such as the roadway median, vehicles etc. **Fig 5** shows the binary image resulting from the threshold segmentation.

c. Median filtering

Due to the windshields and other noise in the image, vehicles may appear as many pieces in the binary thresholded image. Therefore, consolidation of the vehicle into one blob is important for detecting the vehicles. For this purpose, a horizontal median filter is used where the length of the horizontal filter is selected to be of the size of a standard car (15 ft). **Fig 6** shows the same image after median filtering.

d. Connected component analysis

Median filtering produces a binary image which shows the consolidated objects. In order to analyze these objects separately, connected component analysis of the image is performed. Using the efficient run length implementation of the local table method [8], all the connected objects in the binary image are labeled uniquely. The area of the object, and the major and minor axes of the object, are also calculated [9]. The major and minor axes are determined such that the second moment of the equivalent ellipse is the same as that of the object itself.

e. Shape analysis

In order to distinguish between vehicles and other objects in the binary image, it is useful to analyze the shapes of the objects. The objects that have clearly dissimilar features to a typical vehicle are eliminated. The scale of the image is used for determining the features of a standard vehicle such as area and length on the ground. The scale of the image may be obtained automatically by using camera properties and the information about the elevation of the camera.

Limiting values for the features such as length, width, area and aspect ratio are selected in such a way to include all the vehicles that are served by that intersection. For example, maximum length of the vehicle on a major arterial can be 75 ft (a large tractor-trailer). An object which has a major axis larger than 75 ft thus is eliminated. Similarly, the maximum width of the vehicle is limited by the lane width. Hence, an object having minor axis greater than 14 ft, a maximum lane width, may also be eliminated.

All the objects that have any feature significantly different from a typical vehicle are eliminated. In this way, the roadway median, parts of the buildings and other non-vehicle objects are removed from the image. See **Fig 7**.

f. Queue length estimation

According to the Institute of Transportation Engineers [10], the queue at an intersection may be defined as the number of vehicles that are separated by less than a car length. Furthermore, it is reasonable to assume that the queue lengths in different lanes at an approach are approximately the same.

Once the non-vehicle objects are eliminated from the image, the remaining objects in the image are assumed to represent vehicles. To find the position of the queue in this image, the sum of pixel values in a column is plotted against the number of the column. Since only the vehicle pixels are 'on', a plateau is expected at the queue position. Furthermore, since the scale of the image is known, it is possible to set a threshold value for detecting the position of this plateau.

$$S_j = \sum_i g(i,j)$$

where g(i,j) is the pixel value at position (i,j) .

Due to the non-uniform gaps between the stopped vehicles, S_j has sharp fluctuations. Therefore, to locate the queue more accurately, S_j is smoothed in the following fashion.

$$S_j = \text{average}(S_{j-a/2} : S_{j+a/2})$$

$$\text{for all } j \in (a/2, n-a/2)$$

where

S_j: sum of pixel values in column j

n: total number of columns

The value of 'a' is selected to be two times the length of standard car (15ft), assuming the average spacing between vehicles is about one car length. See **Fig 8**.

The position of the queue on the horizontal axis of the image is determined by thresholding the value of S_j in **Fig 8**. The plateau region between the highest peaks signifies the horizontal position of the queue. To locate the position of the queue on the vertical axis, a similar approach is taken. The sum of pixel values across a row contained by the horizontal extent of the queue is plotted against the row number. Again, smoothing of the sum (S_i) results in a clear indication of the position of the queue on the vertical axis.

$$S_i = \sum_{j=Q_{x1}}^{Q_{x2}} g(i,j)$$

where Q_{x1}, Q_{x2} are the horizontal limits of the queue.

g. Estimating the number of vehicles in the queue

I. Counting individual vehicles

As indicated before, the connected component labeling algorithm labels all the objects uniquely. Therefore all the vehicles that lie in the queue box defined above are counted to provide an estimate of the number of vehicles in the queue.

II. Queue polygon

An alternate way to estimate the number of vehicles in the queue is to define a polygon containing all the vehicles in the queue [11]. This polygon is drawn in such a way that it has its vertices on the outmost vehicles in the queue box defined above. See **Fig 9**. The area of the polygon is calculated using the scale of the image. Assuming an average vehicle spacing, the total number of vehicles in the queue can be calculated using the following equation.

$$V_s = \frac{A_{POL}}{S \times l_w \times H_s}$$

where

V_s: Total queue length for the approach [veh]
A_{pol}: Area of the polygon [pixel]
S: Image scale [pixel/ft]
l_w: Average lane width [ft]
H_s: Average spacing for stopped vehicles [assumed to be 30 ft]

III. Identifying vehicles in different lanes

Sufficiently accurate information can be gathered about the turning movements at an intersection if the position of the lanes can be detected. It may be recalled that the position of the queue, the major and minor axes of the object, and the orientation of the objects are known from the connected component analysis. The following algorithm finds the number of vehicles in queue in each lane. Therefore, to identify vehicles in a lane, a vehicle is identified at one end of the queue and based on the orientation of the vehicle, the closest vehicle in the same lane is detected.

1. Identify an object at the start of the queue, given by step (f).
2. Find out the major axis and orientation of the object , given by step (d).
3. Extend the major axis of the vehicle towards the back of the queue in the direction given by orientation until it finds another object or it reaches the end of the queue.
   If the end of the queue is detected, then remove all the objects in the lane from the image and go to step 1.
   Else, include the object, and go to step 2.
   If no object remains, terminate.

This technique thus calculates the number of vehicles in each lanes that are in the queue.

Fig. 5. Thresholded image

Fig. 6. Image after median filtering

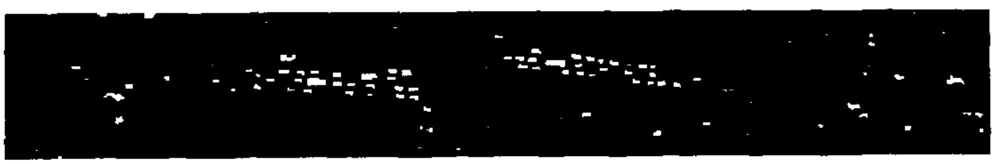


Fig. 7. Image after shape analysis

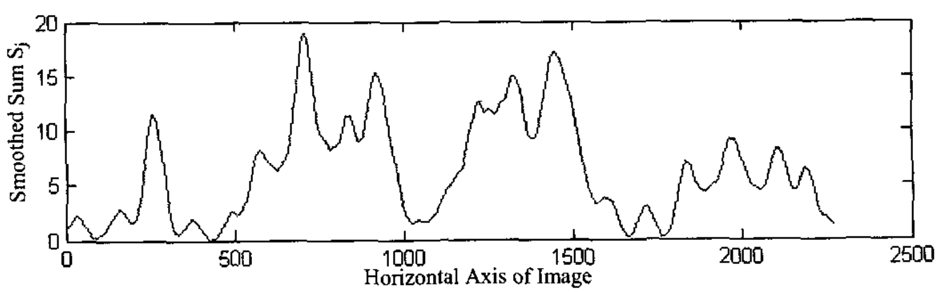


Fig. 8. Smoothed plot of S_j

Fig. 9. Queue polygon

III. EXPERIMENTS

A set of test images were collected as a part of a project conducted at the University of Arizona [12]. Aerial imagery from Tucson and Seattle were collected. As can be seen in **Table 1**, data collection was performed in reasonably different conditions and hence may be considered to represent a sufficiently comprehensive set of conditions.

TABLE I
AIRBORNE IMAGERY TO TEST THE METHODOLOGY

Date	City	Time of day	Number of images
08/29/02	Seattle	7:00 am	6
08/29/02	Seattle	2:30 pm	8
08/30/02	Seattle	7:00 am	3
08/30/02	Seattle	2:00 pm	3
05/06/02	Tucson	2:00 pm	12
05/06/02	Tucson	5:00 pm	4
06/21/02	Tucson	2:00 pm	3
06/24/02	Tucson	8:00 am	4
02/05/03	Tucson	2:00 pm	6
02/05/03	Tucson	10:00 am	2

A sample of more than 50 images was selected from this wide set of video imagery. These images were first cropped manually to minimize the off-road objects and then the scale of the image was determined. These images were then fed into a MATLAB program [13] which processed the images and estimated the number of vehicles by both techniques: counting individual vehicles, and using the queue polygon method.

The algorithm was implemented in MATLAB [13] and run on a Pentium IV processor with processing speed of 2.8 GHz. On average, a high resolution image (2272 x 1704) took around 8-10 seconds to provide the estimates of the queue from both techniques.

A low resolution image (720 x 480) took around 4-5 seconds for processing. It should be noted that the traffic state at an intersection does not change significantly in this short duration and hence it is reasonable to grade this analysis as a real-time analysis.

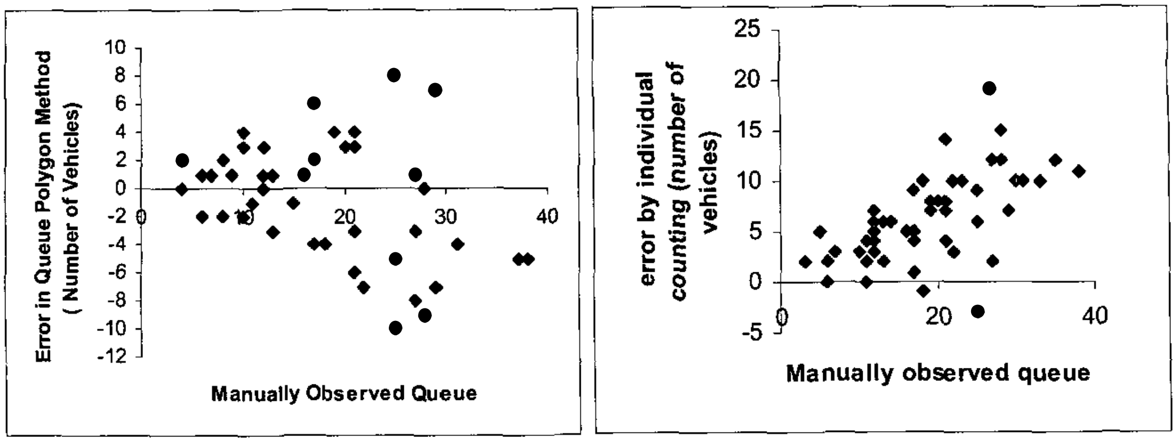
IV. RESULTS

Fig 10 shows how the magnitude of error changes with respect to the total number of vehicles in the queue that are observed manually. For the set of test images, the root mean square errors for individual counting of vehicles and for the queue polygon method were found to be 7.6 vehicles (32%) and 5.8 vehicles (22%) respectively. Similarly, average absolute errors for individual counting of vehicles and for the queue polygon method were found to be 5.1 vehicles (28%) and 3.3 vehicles (19%) respectively. It can be observed that the error in the queue polygon method is fairly symmetric with respect to the X-axis and the magnitude of error is reasonably less than the error in counting individual vehicles. [14] may be referred for further information about the test images and the accuracy of queue estimates.

V. CONCLUSIONS

A methodology has been presented in this paper to estimate the queue lengths at an intersection from an aerial image obtained from a camera mounted on a moving platform. The two techniques developed in this paper compliment each other and provide a good approximation of the queue length at the intersection. The algorithm is tested rigorously against the low and high resolution images taken at major intersections in the Tucson and Seattle metropolitan areas. It may be noted that the accuracy of counting individual vehicles is less than that of the queue polygon method. However, the estimates by the former technique are usually lower than the actual queue lengths. Therefore, a correction factor obtained on the basis of average error may produce superior results, particularly if the proportion of dark vehicles is relatively constant across images.

One of the main drawbacks of this algorithm, which needs more research, is the process of finding the region of interest on the image. A technique is sought which can find the region of interest in a given image without human intervention. Apart from this, the proportion of dark vehicles in the queue also affects the performance of the algorithm because intensity of those pixels is similar to that of surrounding road. However, the queue polygon method does not depend on the number of vehicles detected inside the queue polygon and hence provides better results in the case of a high proportion of dark vehicles.



(a) (b)

Fig 10 Error in queue estimation: (a) for queue polygon; (b) for counting individual vehicles.

REFERENCES

1. F.V.Webster, "Traffic signal settings." *Road Research Laboratory Paper No. 39*, London, 1958.
2. I. Catling, "A time dependent approach to junction delays." *Traffic Engineering and Control*, Vol. 18(11), pp. 520-526,1977.
3. M. Cremer and T. Henninger, "Estimation of queue length in urban road network." *Proceedings, Pacific Rim TransTech Conference*, pp. 29-35, Seattle, 1993.
4. F.A. Wagner and A.D. May Jr, "Use of aerial photography in freeway traffic operations studies." *Highway Research Record 19, HRB, National Research Council*, pp. 24- 34, Washington, D.C., 1963.
5. M. Fathy and M.Y. Siyal, "An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis." *Pattern Recognition Letters*, Vol. 16, No. 12, pp. 1321-1330, December 1995.
6. D.J. Dailey and L. Li, "Video image processing to create a speed sensor." *Final Report, Prepared for Transportation Northwest and FHWA, US Department of Transportation*, University of Washington, March 1999.
7. I.Sekita, T.Kurita, N.Otsu, and N. Abelmalek, "A thresholding method using mixture of normal density functions." *International Symposium on Speech, Image Processing and Neural Networks*, Hong Kong, April 1994.
8. M.R. Haralick, and L. G. Shapiro, *Computer and Robot Vision*, Vol. I. Addison-Wesley, pp. 28-48, 1992.
9. W.K. Pratt, *Digital Image Processing*. New York: John Wiley & Sons, Inc., p. 634, 1991.
10. Institute of Transportation Engineers, *Traffic Engineering Handbook*, 1992.
11. A. Angel, "Measuring highway level of service from airborne platforms." *MS Thesis*, University of Arizona, 2002.
12. A. Angel, M. Hickman, D. Chandnani, and P. Mirchandani, "Application of aerial video for traffic flow monitoring and management." *ASCE 7th International Conference on Applications of Advanced Technology in Transportation*, Cambridge, MA, August 2002.
13. MATLAB, *User's Guide and Reference Manual*, Version 6.5.
14. A. Agrawal, "Automated queue estimation from airborne imagery." *MS Thesis*, University of Arizona, 2004.